インテル® アーキテクチャ対応 インテル® インテグレーテッド・パフォーマンス・プリミティブ（IPP）

リファレンス・マニュアル

第 3 巻：小行列

資料番号 : A68761-4001J

Web : http://www.intel.co.jp/jp/developer/（日本語）
: http://developer.intel.com（英語）

バージョン	バージョン情報	日付
-2001	インテル®インテグレーテッド・パフォーマンス・プリミティブ（インテル®IPP）2.0 ベータ・リリースについて説明。	2001 年 8 月
-3001	インテル IPP 3.0 ベータ・リリースについて説明。	2002 年 6 月
-3002	インテル IPP 3.0 ゴールド・リリースについて説明。	2002 年 11 月
-4001	インテル IPP 4.0 ベータ・リリースについて説明。	2003 年 5 月

【輸出規制に関する告知と注意事項】

本資料に掲載されている製品のうち、外国為替および外国為替管理法に定める戦略物資等または役務に該当するものについては、輸出または再輸出する場合、同法に基づく日本政府の輸出許可が必要です。また、米国産品である当社製品は日本からの輸出または再輸出に際し、原則として米国政府の事前許可が必要です。

【資料内容に関する注意事項】

- 本ドキュメントの内容を予告なしに変更することがあります。
- インテルでは、この資料に掲載された内容について、市販製品に使用した場合の保証あるいは特別な目的に合うことの保証等は、いかなる場合についてもいたしかねます。また、このドキュメント内の誤りについても責任を負いかねる場合があります。
- インテルでは、インテル製品の内部回路以外の使用にて責任を負いません。また、外部回路の特許についても関知いたしません。
- 本書の情報はインテル製品を使用できるようにする目的でのみ記載されています。

  インテルは、製品について「取引条件」で提示されている場合を除き、インテル製品の販売や使用に関して、いかなる特許または著作権の侵害をも含み、あらゆる責任を負わないものとします。
- いかなる形および方法によっても、インテルの文書による許可なく、この資料の一部またはすべてを複写することは禁じられています。

インテル、Intel ロゴ、Intel XScale、Itanium、MMX、Pentium は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

* その他の社名、製品名などは、一般に各社の商標または登録商標です。

© 2000-2004, Intel Corporation.

目 次

第 1 章　概要

インテル® IPP ソフトウェアについて 1-1
　必要なハードウェアとソフトウェアの構成 1-1
本書について 1-2
　本書の構成 1-2
　関数の説明 1-2
　本書の対象読者 1-3
　表記規則 1-3
　　字体の規則 1-3
　　命名規則 1-3

第 2 章　はじめに

行列演算用インテル® IPP の目的 2-1
ROI（Regions of Interest） 2-1
　ROI シフトの概念 2-2
　ストライド 2-3
　ポインタ 2-3
関数の命名 2-7
　名前 2-7
　オブジェクト 2-7
　データ・タイプ 2-8
　オブジェクトのサイズ 2-8
　ディスクリプタ 2-8
　引数 2-9

第 3 章　ユーティリティ関数

　Copy 3-1
　Extract 3-3
　LoadIdentity 3-4

第 4 章　ベクトル代数関数

　Saxpy 4-1
　Add 4-3
　Sub 4-5
　Mul 4-7
　CrossProduct 4-8
　DotProduct 4-10
　L2Norm 4-11
　LComb 4-13

第 5 章　行列代数関数

Transpose 5-1
Invert 5-3
FrobNorm 5-4
Det 5-5
Trace 5-6
Mul 5-8
Add 5-12
Sub 5-14
Gaxpy 5-16

第 6 章　連立 1 次方程式の解の関数

LUDecomp 6-1
LUBackSubst 6-3

第 7 章　最小 2 乗問題関数

QRDecomp 7-1
QRBackSubst 7-3

付録 A ユーティリティ関数のプロトタイプ

Copy A-1
Extract A-4
LoadIdentity A-6

付録 B ベクトル代数関数のプロトタイプ

Saxpy B-1
Add B-18
Sub B-30
CrossProduct B-58
DotProduct B-62
L2Norm B-72
LComb B-76
Mul B-95

付録 C 行列代数関数のプロトタイプ

Transpose C-1
Invert C-7
FrobNorm C-14
Det C-19
Trace C-24
Mul C-29
Gaxpy C-193
Add C-201
Sub C-256

付録 D 連立 1 次方程式の解の関数のプロトタイプ

LUDecomp D-1

LUBackSubst D-8

付録 E 最小 2 乗問題の解の関数のプロトタイプ

QRDecomp E-1
QRBackSubst E-8

索 引

概要

1

本書では、小行列の演算用のインテル® インテグレーテッド・パフォーマンス・プリミティブ（インテル® IPP）の構造、動作、関数について説明する。本書は、インテル IPP リファレンス・マニュアルの第 3 巻である。第 1 巻は信号処理用のインテル IPP、第 2 巻は画像および動画処理用のインテル IPP、第 4 巻は暗号化用のインテル IPP を対象としている。インテル IPP ソフトウェア・パッケージがサポートしている多くの機能は、インテル® アーキテクチャ（IA）、特にインテル® MMX® テクノロジとストリーミング SIMD 拡張命令（SSE）向けに最適化されている。

行列演算用のインテル IPP は、IA-32 およびインテル® Itanium® アーキテクチャ向けに最適化され、Microsoft* Windows* および Linux* オペレーティング・システム上で実行される、クロスプラットフォーム・ソフトウェア・レイヤである。

本書では、小行列の演算用に開発されたインテル IPP 関数について詳しく説明する。

本章では、インテル IPP 行列演算ソフトウェアについて紹介し、本書の構成について説明する。

インテル® IPP ソフトウェアについて

インテル® IPP ソフトウェアは、インテル® MMX® テクノロジとストリーミング SIMD 拡張命令の中核をなす SIMD（single-instruction, multiple-data）命令を活用できる。

必要なハードウェアとソフトウェアの構成

インテル® アーキテクチャ・ソフトウェア用のインテル® IPP は、Microsoft* Windows* 2000、Windows ME、Windows XP、または Linux* オペレーティング・システムをサポートしている、IA-32 プロセッサ・ベースまたはインテル® Itanium® アーキテクチャ・プロセッサ・ベースのパーソナル・コンピュータ上で動作する。本ソフトウェアは、C または C++ で記述されたユーザのアプリケーションやライブラリに組み込むことができる。

本書について

本書は、インテル® IPP ソフトウェアの行列演算の概念の背景と、個々のインテル IPP 関数の詳細について説明する。インテル IPP 関数は、機能別にグループ分けされている。各関数グループについて、本書の章ごとに説明する。

本書の構成

本書は、次の章で構成されている。

第 1 章　「概要」。行列演算用インテル® IPP の紹介、本書の構成、表記規則について説明する。

第 2 章　「はじめに」。行列演算用のインテル IPP 関数の基本概念と、使用できるデータ・レイアウトおよび動作モードについて説明する。

第 3 章　「ユーティリティ関数」。ベクトルと行列のコピー、ROI（Regions of Interest）の抽出、行列のタイプの変換、行列の初期化に使用される関数について説明する。

第 4 章　「ベクトル代数関数」。ベクトルの加算、減算、スケーリングと、ベクトル代数グループのその他の演算に使用されるインテル IPP 関数について説明する。

第 5 章　「行列代数関数」。行列の加算、減算、およびスケーリング、行列 - ベクトルの積と行列 - 行列の積の計算、行列代数のその他の演算に使用されるインテル IPP 関数について説明する。

第 6 章　「連立 1 次方程式の解の関数」。LU 分解の計算と、後退代入による連立 1 次方程式の解の算出に使用される関数について説明する。

第 7 章　「最小 2 乗問題関数」。行列の QR 分解の計算と、最小 2 乗問題の解の算出に使用されるインテル IPP 関数について説明する。

すべてのインテル IPP 行列演算関数のすべてのプロトタイプは、第 3 章 ～ 第 7 章の順序に従って、付録 A、B、C、D、および E に記載されている。

本書の巻末には、本巻で使用する主な用語と定義の「索 引」がある。

関数の説明

第 3 章 ～ 第 7 章では、各関数を短い名前で（ippm プリフィックスとディスクリプタを省略して）示し、関数の目的について簡単に説明する。それに続いて、関数の呼び出しシーケンスの例、すべての引数の定義、および関数の目的の詳しい説明を示す。関数の説明は、以下の項目で構成される。

プロトタイプ例	現在のグループの関数のうち 1 つの呼び出しシーケンスを示す。それに続いて、同じグループの他の関数型のプロトタイプへのリンクをまとめた表が示される。
引数	すべての関数型の引数を説明する。
説明	関数の定義を示し、関数が実行する演算について詳しく説明する。必要に応じて、コード例と関数が計算する式を示す。
戻り値	関数の実行の結果として設定されるステータス・コードの値について説明する。

本書の対象読者

本書は、小行列の演算を多用する（または、小行列の演算の結果を利用する）アプリケーションの開発者を対象としている。本書の読者には、C 言語の使用経験と、行列の演算に関する用語と原理の知識が必要である。

表記規則

本書では、以下の表記規則を使用する。

- 字体の規則（テキストとコードの区別）
- 各種のアイテムの命名規則

字体の規則

本書全体を通して、以下の字体の規則を使用する。

`This type style`	関数名、コード例、呼び出し文を示す。大文字が含まれる。 例：`ippmAdd_mTmT_32f`
`This type style`	引数とパラメータの説明で変数を示す。 例：*`src1Stride1`*, *`width`*

命名規則

インテル® IPP ソフトウェアは、各種のアイテムに対して以下の命名規則を使用する。

- 行列の演算に使用されるすべての関数の名前には、ippm プリフィックスが付く。コード例では、このプリフィックスによって、インテル IPP インターフェイス関数とアプリケーションの関数を区別できる。

注 : 本書のコード例と関数プロトタイプの中では、関数名に必ず ippm プリフィックスが付いている。本文中では、このプリフィックスを省略する場合がある。

- 関数名の新しい部分は、下線を挿入せずに大文字で始められる（例えば、ippmFrobNorm）。下線は、関数名に続くデータ・タイプとデータ・ディスクリプタの区切りとして使用される。例 : ippmFrobNorm_ma_32f_5x5)

インテル IPP の関数名の構造の詳細は、第 2 章の「関数の命名」を参照のこと。

はじめに

2

本章では、行列演算用のインテル® インテグレーテッド・パフォーマンス・プリミティブの目的と構造について説明する。

また、行列演算用インテル® IPP の基本的な概念と、本書の関数の命名規則についても説明する。

行列演算用インテル® IPP の目的

小行列の演算用のインテル® IPP は、各種のグラフィックス、コンピュータ・ゲーム・アプリケーション、CAD アプリケーションの開発など、多くのソフトウェア開発タスクに対応するソリューションである。例えば、インテル IPP ソリューションは、点座標の座標系の変換、物理的な動きのモデリング用のダイナミクスの計算、連立 1 次方程式の解の算出などの処理を必要とするアプリケーションに便利である。

注： ソース・アドレスとデスティネーション・アドレスが同一または重なる場合は、異なるプロセッサ向けに最適化されたインテル IPP の各バージョンの間で、演算結果が一致しないときがある。

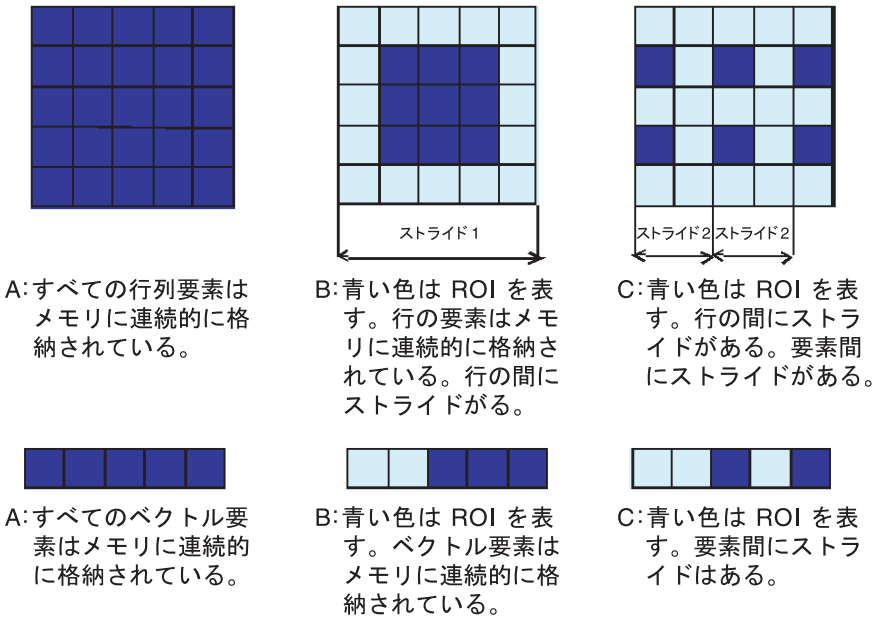
ROI（Regions of Interest）

特定領域のアプリケーションでは、オブジェクト全体を操作するより、オブジェクトの一部分を操作する方が効率的である。例えば、処理する行列のサイズを小さくして計算量を減らすには、結節点の定義などの複雑な制限と要件を伴う行列全体を使用する代わりに、部分行列を使用すればよい。

行列演算用インテル® IPP は、ROI（Region of Interest）の概念を使用して、オブジェクトの一部分の操作を可能にする。ROI とは、ベクトル / 行列の特定の部分またはベクトル / 行列の全体の領域である。

ROI は、推定、演算、または処理を通常より高速で実行する必要がある場合や、特定のフィールドを除外してデータ・レイアウトまたはデータ構造を処理したい場合に、柔軟に利用できる。

図 2-1　行列内の ROI とベクトル内の ROI



ROI シフトの概念

ROI シフトの概念により、大きなデータ配列を処理する場合に、データ処理の柔軟性が向上する。ROI シフトとは、オブジェクトへのポインタと ROI へのポインタの間のバイト差である。

ROI シフトは、*L*- ポインタと *P*- ポインタでは非常に便利であるが、*S*- ポインタでは使用されない。

L- ポインタと *P*- ポインタをマスクとして扱い、メモリに格納されているオブジェクトまたはオブジェクトの要素の定義と抽出に使用できる。ROI シフトを使ってこれらのマスクを動かすことで、オブジェクトとオブジェクトの要素の定義と抽出が簡単に行える。

ストライド

標準的なオブジェクトのレイアウトでは、データはメモリに連続的に格納されるとは限らないため、データの処理が難しくなる。行列演算用インテル® IPP は、その解決策として、次の 3 種類のストライドを使用する

ストライド s_0 は、オブジェクト間のストライドである（図 2-2 を参照）。このストライドは、*s*- ポインタでのみ使用できる。

ストライド s_1 は、行列の行の間のストライドである（図 2-1 を参照）。図 2-1 の条件 *C* の行列では、ストライド s_1 は幅の 2 倍に等しい。このストライドは、どのポインタでも使用できる。

ストライド s_2 は、ベクトル要素間または行列の行要素間のストライドである（図 2-2 を参照）。

すべてのストライドは、バイト単位で測定される。

ポインタ

オブジェクト配列のアドレス指定に、次の 3 種類のポインタを使用できる。

s- ポインタ　最初のオブジェクトまたは別のオブジェクトへのポインタ

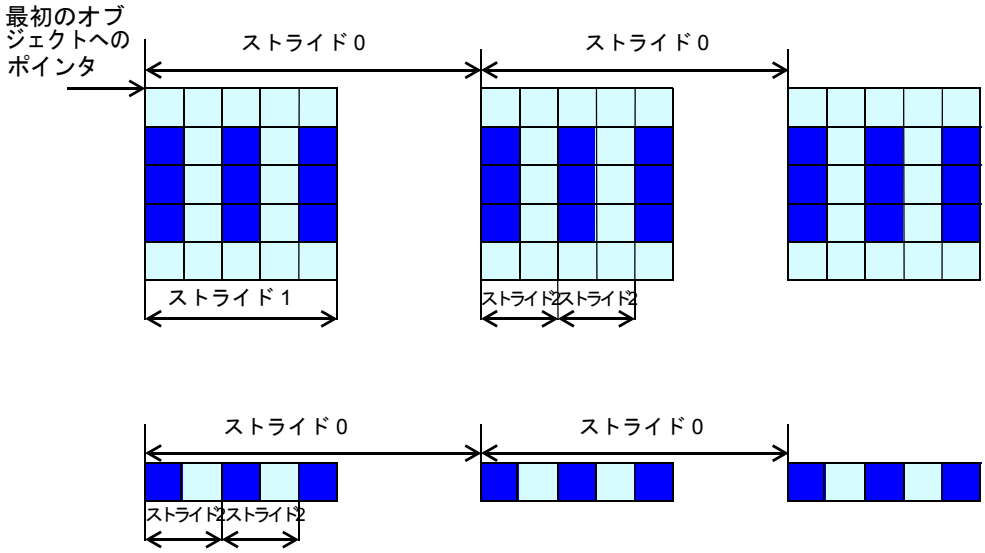
L- ポインタ　配列内の各オブジェクトへのポインタ

P- ポインタ　すべてのオブジェクトにわたる値へのポインタ

***s*- ポインタ**は、オブジェクトがメモリにシーケンシャルに（つまり、オブジェクト間の一定のストライドで）格納されている場合に使用される標準的なポインタである。この意味では、シーケンシャルなレイアウトは、オブジェクトのサイズに等しいストライドを持つ、連続的レイアウトの特殊な条件である。

図 2-2 のシーケンシャルなオブジェクト配列では、オブジェクトはメモリにシーケンシャルに格納され、オブジェクト間のストライド、行列の行の間のストライド、値の間のストライドが存在する。

図 2-2　オブジェクトのシーケンシャルな配列。最初のオブジェクトへのポインタ

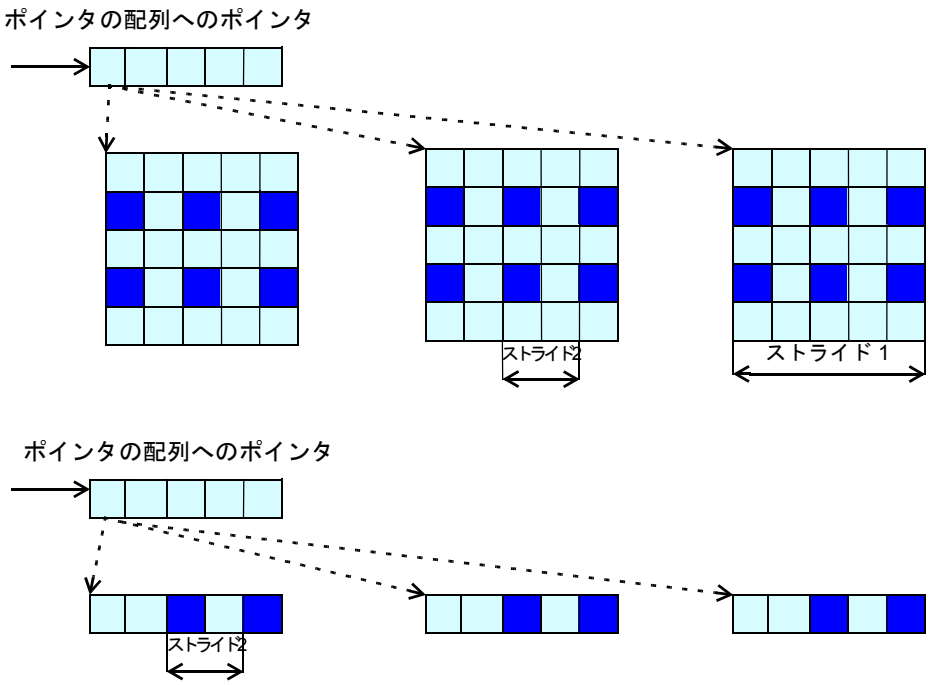


L-ポインタは、それぞれのオブジェクトへのポインタを格納する配列へのポインタである。このポインタは、オブジェクトがシーケンシャルに格納されていない場合に使用される。*L*- ポインタは、画像処理の *L*- ポインタ（*L* は「レイヤ」を表す）と同様である。行列演算用インテル® IPP にはレイヤは存在しないが、メモリに格納されたオブジェクトを「レイヤ」として考えられる。

行列演算用インテル IPP では、*L*- ポインタは、同じサイズ、ROI シフト、ストライドを持つ複数のオブジェクトに使用される。

任意のオブジェクト配列で（図 2-3 を参照）、オブジェクトは任意の方法でメモリに格納されている。各オブジェクトに 1 つのポインタが対応し、青い色で示される ROI がある。行列の行の間と値の間にストライドがある。

図 2-3　任意のオブジェクト配列。各オブジェクトへのポインタ



P- **ポインタ**は、各オブジェクト要素へのポインタを格納する配列へのポインタである。このポインタは、配列構造体（Structure of Array:SOA）レイアウトのデータを処理する場合に使用する。図 2-4 では、*P-* ポインタを使用してベクトル配列を定義する方法を示す。上の図では、ベクトルは配列構造体（Structure of Arrays: SOA）形式で表現される。つまり、1 つの構造体が複数の配列から構成される。下の図では、データは構造体配列（Array of Structures: AOS）形式で表現される。つまり、その要素がデータ構造体である配列である。*P-* ポインタを使用することで、どちらの形式でもデータを表すことができる。

図 2-4　オブジェクトのシーケンシャルな配列。値へのポインタ

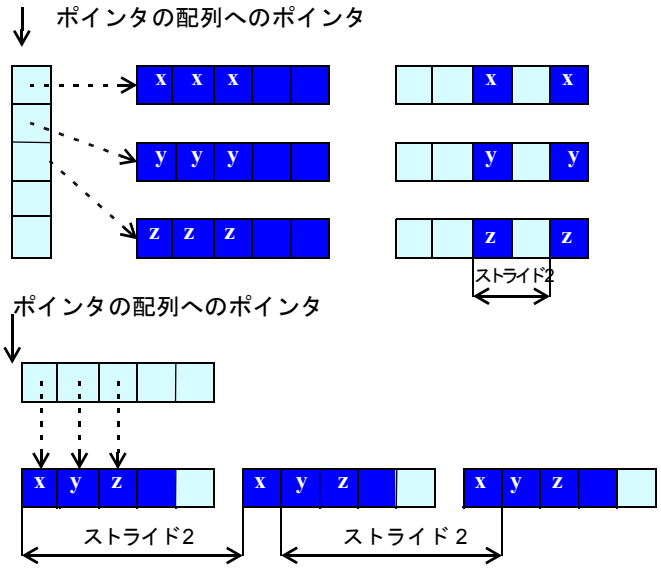
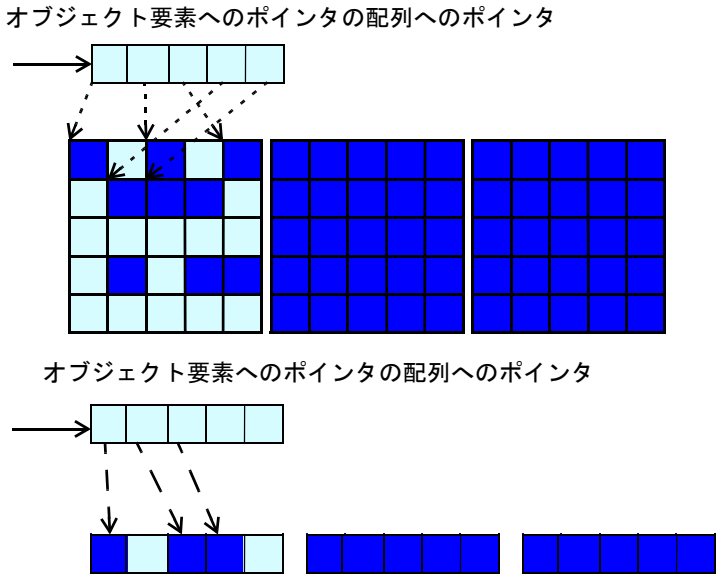


図 2-5 に示すオブジェクトのシーケンシャルな配列では、すべての値はメモリにシーケンシャルに格納されている。値の間にストライドが存在し、ROI が選択されている可能性がある。

ROI シフトを使用して、メモリ上の任意の位置からオブジェクトの要素を選択できる。

図 2-5　P- ポインタの指定



P- ポインタを使用して、各種のマスクを作成できる。ROI シフトを使用して、メモリ上の任意の位置にマスクを動かせる。

関数の命名

インテル® IPP の行列演算領域の関数名は、先頭に ippm プリフィックスが付けられ、以下の一般的な形式で表される。

```
ippm<name>_<objects>_<datatype>[_objectsize][_descriptor](<arguments>);
```

名前

<name> は、関数の基本的な動作を示す省略形である（例えば、Add、Copy など）。<name> は、複数の機能で構成される場合がある。関数名の新しい部分は、下線を挿入せずに大文字で始められる（例えば、ippmFrobNorm）。

オブジェクト

```
objects = <objecttype1><objecttype2>
```

オブジェクトのタイプは、処理のために関数に渡されるオブジェクトのタイプを記述する。オブジェクトのタイプには、次のものがある。

c	定数
v	ベクトル
va	ベクトルの配列
m	行列
mT	転置行列
ma	行列の配列
maT	転置行列の配列

データ・タイプ

インテル® IPP の現在のバージョンでは、行列演算を実行する関数のソースとデスティネーションに、次のデータ・タイプをサポートしている。

32f	32 ビット浮動小数点
64f	64 ビット倍精度

オブジェクトのサイズ

関数名のこの部分は、関数が処理するベクトルまたは行列のサイズを記述する。

- ベクトルのサイズとして、3x1、4x1、5x1、6x1 を選択できる。
- 列のサイズとして、3x3、4x4、5x5、6x6 を選択できる。

この構成要素が関数名に含まれていない場合、その関数は Nx1（ベクトル）または NxM（行列）の条件で記述されている。

ディスクリプタ

<descriptor> フィールドは、演算に関連するデータをさらに記述する。このフィールドには、暗黙のパラメータや追加の必須パラメータを指定できる。関数内のコード分岐数を最小限に抑え、不要な実行オーバーヘッドを減らすために、ほとんどの汎用関数は、個々のプリミティブ関数に分割されている。関数のパラメータの一部は、プリミティブ関数名の中でディスクリプタとして指定される。

行列演算関数には、次のディスクリプタが使用される。

I	インプレース演算。
L	オブジェクトは *L*- ポインタ（各オブジェクトへのポインタの配列へのポインタ）によって定義される。

P	オブジェクトは *P*- ポインタ（すべてのオブジェクトにわたるオブジェクト要素へのポインタの配列へのポインタ）によって定義される。
S2	オブジェクト要素間のストライド。

関数名の中では、ディスクリプタの省略形は常にアルファベット順で記述される。

一部のディスクリプタは、関数名から省略されるときがある。例えば、ベクトルの演算の場合、行優先のレイアウトか列優先のレイアウトかの指定は意味がない。

また、一部の演算では、特定のデータ・ディスクリプタが暗黙的に前提とされる。

インテル® IPP 行列演算関数のデフォルト値は次のとおりである。

- *S*- ポインタ（最初のオブジェクトへのポインタ）を使用する。
- インプレース計算を行わない。
- 行優先の順序で格納された行列を処理する。
- ROI なしのオブジェクトを処理する。
- オブジェクト要素間のストライドがないオブジェクトを処理する。

これらの場合、暗黙的な省略形は関数名に含まれない。

引数

<arguments> は、関数の引数を指定する。

パラメータの順序は次のとおりである。

- すべてのソース・オペランド。ベクトルの後に定数が続く。
- すべてのデスティネーション・オペランド。ベクトルの後に定数が続く。
- その他、演算固有のパラメータ。

パラメータの名前には以下の規則がある。

- ポインタとして定義されたすべてのパラメータは *p* で始まる（例えば、*pSrc*、*pDst*）。値として定義されたパラメータは小文字で始まる（例えば、*val*、*len*、*count*）。
- パラメータ名の新しい部分は、下線を挿入せずに大文字で始まる（例えば、*pSrc*、*lenSrc*）。

- 各パラメータ名は、そのパラメータの機能を指定する。ソース・パラメータは *pSrc* または *src* と呼ばれ、その後に名前または番号が続くことがある（例えば、*pSrc2*、*srcLen*）。
- 出力パラメータは *pDst* または *dst* と呼ばれ、その後に名前または番号が続く（例えば、*pDst*、*dstLen*）。インプレース演算の場合は、入力 / 出力パラメータ名に *pSrcDst* が含まれる。

ユーティリティ関数

3

本章では、ベクトルと行列のコピー、ROI（Regions of Interest）の抽出、行列のタイプの変換、行列の初期化に使用されるインテル® IPP 関数について説明する。

表 3-1　ユーティリティ関数

関数の基本名	操作
Copy	コピー操作を実行する。
Extract	ROI の抽出を実行する。
LoadIdentity	単位行列を初期化する。

Copy

コピー操作を実行する。

プロトタイプ例

```
IppStatus ippmCopy_ma_32f_SS (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s
        dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
        Ipp32u height, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector array	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc　ソース・オブジェクトまたはオブジェクトの配列へのポインタ。

srcStride0　ソース配列内のオブジェクト間のストライド。

srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース・オブジェクトの要素間のストライド。
srcROIShift	ソース・オブジェクト内の ROI シフト。
pDst	デスティネーション・オブジェクトまたはオブジェクトの配列へのポインタ。
dstStride0	デスティネーション配列内のオブジェクト間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション・オブジェクトの要素間のストライド。
dstROIShift	デスティネーション・オブジェクト内の ROI シフト。
width	行列の幅。
height	行列の高さ。
len	ベクトルの長さ。
count	配列内のオブジェクト数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、任意のタイプのオブジェクトを任意のタイプの他のオブジェクトにコピーし、デスティネーションへのポインタを *pDst* に格納する。

行列に対して実行する場合、演算に使用されるすべての行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

ベクトルに対して実行する場合、演算に使用されるすべてのベクトルの要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Extract

ROI の抽出を実行する。

プロトタイプ例

```
IppStatus ippmExtract_maT_64f_LS2 (const Ipp64f** pSrc, Ipp32u
srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u
matrWidth, Ipp32u matrHeight, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector	32f	64f
vector array	32f	64f
matrix	32f	64f
transposed matrix	32f	64f
matrix array	32f	64f
transposed matrix array	32f	64f

全引数のリスト

pSrc	ソース・オブジェクトまたはオブジェクトの配列へのポインタ。
srcStride0	ソース配列内のオブジェクト間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース・オブジェクトの要素間のストライド。
srcROIShift	ソース・オブジェクト内の ROI シフト。
pDst	デスティネーション・オブジェクトまたはオブジェクトの配列へのポインタ。
len	ベクトルの長さ。
matrwidth	行列の幅。
matrheight	行列の高さ。
count	配列内のオブジェクト数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、任意のタイプのオブジェクトから特定のプロパティを持つ他のオブジェクトに ROI を抽出する。

行列に対して実行する場合、演算に使用されるすべての行列の列の数は *matrwidth* と同じかそれ以上、行の数は *matrheight* と同じかそれ以上でなければならない。

ベクトルに対して実行する場合、演算に使用されるすべてのベクトルの要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

LoadIdentity

単位行列を初期化する。

プロトタイプ例

```
IppStatus ippmLoadIdentity_ma_32f_S (const Ipp32f* pDst, Ipp32s
        dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32s
        dstWidth, Ipp32s dstHeight, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix array	32f	64f

全引数のリスト

pDst	デスティネーション行列または行列の配列へのポインタ。
srcStride0	ソース配列内のオブジェクト間のストライド。
srcStride1	ソース行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。

dstROIShift	デスティネーション行列内の ROI シフト。
dstWidth	行列の幅。
dstHeight	行列の高さ。
count	配列内のオブジェクトの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、単位行列をロードし、その行列へのポインタを *pDst* に格納する。

デスティネーション行列の列の数は *dstWidth*、行の数は *dstHeight* と同じになる。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

ベクトル代数関数

4

本章では、ベクトル代数演算を実行するインテル® IPP 行列関数について説明する。

表 4-1　ベクトル代数関数

関数の基本名	操作
Saxpy	ベクトルの「SAXPY」演算を実行する。
Add	定数をベクトルに加算するか、ベクトルを他のベクトルに加算する。
Sub	ベクトルから定数を引くか、他のベクトルからベクトルを引く。
Mul	ベクトルに定数を掛ける。
CrossProduct	2 つの 3D ベクトルの外積を計算する。
DotProduct	2 つのベクトルの内積を計算する。
L2Norm	ベクトルの L2 ノルムを計算する。
LComb	2 つのベクトルの線形結合を構成する。

Saxpy

ベクトルの「SAXPY」演算を実行する。

プロトタイプ例

```
IppStatus ippmSaxpy_vva_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
        src1ROIShift,  Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
        src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
        dstROIShift,  Ipp32s dstStride2, Ipp32u len, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - vector	32f	64f
vector - vector array	32f	64f
vector array - vector	32f	64f
vector array - vector array	32f	64f

全引数のリスト

pSrc1	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src1Stride0	1 番目のソース・ベクトル配列内のベクトル間のストライド。
src1Stride2	1 番目のソース・ベクトルの要素間のストライド。
src1ROIShift	1 番目のソース・ベクトル内の ROI シフト。
pSrc2	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のベクトル配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。
dstStride0	デスティネーション配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
scale	乗数。
len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、2 つのベクトルの線形結合を構成する。そのために、1 番目のソース・ベクトルに定数を掛けて、その積を 2 番目のソース・ベクトルに加算し、結果をデスティネーション・ベクトルに格納する。

$$dst = c \times src1 + src2$$

ソース・ベクトル（配列に対して実行する場合は複数のベクトル）の要素の数は、*len* と同じかそれ以上でなければならない。

配列に対して実行する場合、この関数は、処理されるすべてのベクトルに対して反復される。

$$dst[i] = c \times src1[i] + src2[i]$$

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Add

定数をベクトルに加算するか、
ベクトルを他のベクトルに加算する。

プロトタイプ例

```
IppStatus ippmAdd_vac_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
        Ipp32s  dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vav_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
        src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
        src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
        dstStride2, Ipp32u len, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - vector	32f	64f
vector array - vector	32f	64f
vector array - vector array	32f	64f

全引数のリスト

pSrc	ソース・ベクトルまたはベクトルの配列へのポインタ。
srcStride0	ソース配列内のベクトル間のストライド。
srcStride2	ソース・ベクトルの要素間のストライド。
srcROIShift	1 番目のソース・ベクトル内の ROI シフト。
pSrc1	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src1Stride0	1 番目のソース・ベクトル配列内のベクトル間のストライド。

src1Stride2	1 番目のソース・ベクトルの要素間のストライド。
src1ROIShift	1 番目のソース・ベクトル内の ROI シフト。
pSrc2	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のベクトル配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。
dstStride0	デスティネーション配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
val	加算される値。
len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数の動作は、他のすべてのインテル® IPP 行列演算関数と同じように、パラメータに依存する。関数名の後に続く、下線の直後のすべての入力パラメータは、関数の実行方法と使用される引数（定数または他のベクトル）を指定する。したがって、完全な関数名とともに、リストされた引数の一部だけが入力リスト内で指定され、他の引数は省略される。

定数とベクトルに対して実行する場合、この関数は、ソース・ベクトルの各要素に *val* を加算し、結果を *pDst* に格納する。

$$dst[i] = val + src[i]$$

ソース・ベクトル（配列に対して実行する場合は複数のベクトル）の要素の数は、*len* と同じかそれ以上でなければならない。

2 つのベクトルに対して実行する場合、この関数は、1 番目のソース・ベクトルの各要素と 2 番目のソース・ベクトルの各要素を加算し、結果のベクトルへのポインタを *pDst* に格納する。

$$dst[i] = src1[i] + src2[i]$$

ソース・ベクトル（配列に対して実行する場合は複数のベクトル）の要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Sub

ベクトルから定数を引くか、
他のベクトルからベクトルを引く。

プロトタイプ例

```
IppStatus ippmSub_vac_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
        Ipp32s  dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vva_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
        src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
        src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
        dstStride2, Ipp32u len, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - constant	32f	64f
vector array - constant	32f	64f
constant - vector	32f	64f
constant - vector array	32f	64f
vector - vector	32f	64f
vector array - vector	32f	64f
vector array - vector array	32f	64f

全引数のリスト

pSrc	ソース・ベクトルまたはベクトルの配列へのポインタ。
srcStride0	ソース配列内のベクトル間のストライド。
srcStride2	ソース・ベクトルの要素間のストライド。
srcROIShift	ソース・ベクトル内の ROI シフト。

pSrc1	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src1Stride0	1 番目のソース・ベクトル配列内のベクトル間のストライド。
src1Stride2	1 番目のソース・ベクトルの要素間のストライド。
src1ROIShift	1 番目のソース・ベクトル内の ROI シフト。
pSrc2	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のベクトル配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。
dstStride0	デスティネーション配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
val	差し引く値。
len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数の動作は、他のすべてのインテル® IPP 行列演算関数と同じように、パラメータに依存する。関数名の後に続く、下線の直後のすべての入力パラメータは、関数の実行方法と使用される引数（定数または他のベクトル）を指定する。したがって、完全な関数名とともに、リストされた引数の一部だけが入力リスト内で指定され、他の引数は省略される。

この関数は、1 番目の引数から 2 番目の引数を引く（例えば、*dst*[*i*] = *src*[*i*] – *val*）。ソース・ベクトル（配列に対して実行する場合は複数のベクトル）の要素の数は、*len* と同じかそれ以上でなければならない。

また、この関数を 2 つのベクトルに対して実行する場合は、この関数は 1 番目のソース・ベクトルの各要素から 2 番目のソース・ベクトルの各要素を引き、結果のベクトルへのポインタを *pDst* に格納する。

dst[*i*] = *src*1[*i*] – *src*2[*i*]

ソース・ベクトル（配列に対して実行する場合は複数のベクトル）の要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Mul

ベクトルに定数を掛ける。

プロトタイプ例

```
IppStatus ippmMul_vac_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
        Ipp32s  dstStride2, Ipp32u len, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - constant	32f	64f
vector array - constant	32f	64f

全引数のリスト

pSrc	ソース・ベクトルまたはベクトルの配列へのポインタ。
srcStride0	ソース配列内のベクトル間のストライド。
srcStride2	ソース・ベクトルの要素間のストライド。
srcROIShift	ソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。
dstStride0	デスティネーション配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
val	乗数。

len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース・ベクトルの要素に定数を掛けて、結果を *pDst* に格納する。

$$dst[i] = c \times src[i]$$

演算に使用されるベクトルの要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

CrossProduct

2 つの 3D ベクトルの外積を計算する。

プロトタイプ例

```
IppStatus ippmCrossProduct_vva_32f_3x1_PS2 (const Ipp32f** pSrc1,
        Ipp32s  src1ROIShift, const Ipp32f** pSrc2, Ipp32s
        src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
        dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - vector	32f	64f
vector - vector array	32f	64f
vector array - vector	32f	64f
vector array - vector array	32f	64f

全引数のリスト

pSrc1	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src1Stride0	1 番目のソース・ベクトル配列内のベクトル間のストライド。
src1Stride2	1 番目のソース・ベクトルの要素間のストライド。
src1ROIShift	1 番目のソース・ベクトル内の ROI シフト。
pSrc2	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のソース・ベクトル配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。
dstStride0	デスティネーション配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、2 つのソース・ベクトルのベクトル積を構成する。デスティネーション・ベクトルの最初の要素は、1 番目のソース・ベクトルの 2 番目の要素と 2 番目のソース・ベクトルの 3 番目の要素の積から、1 番目のソース・ベクトルの 3 番目の要素と 2 番目のソース・ベクトルの 2 番目の要素の積を引くことによって得られる。

次の関係を使用して、デスティネーション・ベクトルの最初の要素と他の 2 つの要素を計算する。

$$dst[0] = src1[1] \times src2[2] - src1[2] \times src2[1]$$

$$dst[1] = src1[2] \times src2[0] - src1[0] \times src2[2]$$

$$dst[2] = src1[0] \times src2[1] - src1[1] \times src2[0]$$

デスティネーション・ベクトルへのポインタは、*pDst* に格納される。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

注 : 関数 ippmCrossProduct は、3 次元ベクトルについてのみ定義されているため、他のすべての引数タイプでは実行に失敗する。

DotProduct

2 つのベクトルの内積を計算する。

プロトタイプ例

```
IppStatus ippmDotProduct_vav_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
        src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
        src2ROIShift, Ipp32f* pDst, Ipp32u len, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - vector	32f	64f
vector array - vector	32f	64f
vector array - vector array	32f	64f

全引数のリスト

pSrc1	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src1Stride0	1 番目のソース・ベクトル配列内のベクトル間のストライド。
src1Stride2	1 番目のソース・ベクトルの要素間のストライド。
src1ROIShift	1 番目のソース・ベクトル内の ROI シフト。

pSrc2	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のソース・ベクトル配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。
len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、2 つのソース・ベクトルの内積（内積）を計算する。そのために、2 つのベクトルの各要素の積をすべて加算し、結果の値を *pDst* に格納する。

$$dst = \sum_{i} src1[i] \times src2[i]$$

演算に使用されるすべてのベクトルの要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

L2Norm

ベクトルの L2 ノルムを計算する。

プロトタイプ例

```
IppStatus ippmL2Norm_va_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride2, Ipp32f* pDst, Ipp32u len, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector	32f	64f
vector array	32f	64f

全引数のリスト

pSrc	ソース・ベクトルまたはベクトルの配列へのポインタ。
srcStride0	ソース配列内のベクトル間のストライド。
srcStride2	ソース・ベクトルの要素間のストライド。
srcROIShift	ソース・ベクトル内の ROI シフト。
pDst	デスティネーション値へのポインタ。
len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース・ベクトルの L2 ノルムを計算し、結果を *pDst* に格納する。デスティネーションの値は、ソース・ベクトルのすべての要素を 2 乗した値の和の平方根(すなわち、以下の式)になる。

$$dst = \sqrt{\sum_i src[i]^2}$$

演算に使用されるすべてのベクトルの要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

LComb

2 つのベクトルの線形結合を構成する。

プロトタイプ例

```
IppStatus ippmLComb_vva_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
        src1ROIShift,  Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
        src2ROIShift, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f** pDst,
        Ipp32s  dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u
        count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector - vector	32f	64f
vector - vector array	32f	64f
vector array - vector	32f	64f
vector array - vector array	32f	64f

全引数のリスト

pSrc1	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src1Stride0	1 番目のソース・ベクトル配列内のベクトル間のストライド。
src1Stride2	1 番目のソース・ベクトルの要素間のストライド。
src1ROIShift	1 番目のソース・ベクトル内の ROI シフト。
pSrc2	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のソース・ベクトル配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトルの配列へのポインタ。
dstStride0	デスティネーション配列内のベクトル間のストライド。

dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
scale1	1 番目の乗数。
scale2	2 番目の乗数。
len	ベクトルの長さ。
count	配列内のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、2 つのベクトルの線形結合を構成する。そのために、1 番目のソース・ベクトルに *scale1*、2 番目のソース・ベクトルに *scale2* を掛けて、それぞれの積を加算し、得られたベクトルへのポインタを *pDst* に格納する。

演算に使用されるすべてのベクトルの要素の数は、*len* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

行列代数関数

5

本章では、行列代数演算を実行するインテル® IPP 関数について説明する。

表 5-1　行列代数関数

関数の基本名	操作
Transpose	行列の転置を実行する。
Invert	行列の逆行列を計算する。
FrobNorm	行列の Frobenius ノルムを計算する。
Det	行列の行列式を計算する。
Trace	行列のトレースを計算する。
Mul	行列に定数、ベクトル、または他の行列を掛ける。
Add	行列を他の行列に加算する。
Sub	行列を他の行列から引く。
Gaxpy	行列の「GAXPY」演算を実行する。

Transpose

行列の転置を実行する。

プロトタイプ例

```
IppStatus ippmTranspose_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s
        srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
        Ipp32u height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
        dstStride1, Ipp32s dstStride2, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	ソース配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド
srcROIShift	ソース行列内の ROI シフト。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	デスティネーション配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列内の ROI シフト。
width	行列の幅。
height	行列の高さ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース行列を転置し、結果を *pDst* に格納する。デスティネーションは、ソース行列の列をデスティネーション行列の行に変換することによって、ソース行列から得られる（すなわち、各行列要素について *dst*[*i*][*j*] = *src*[*j*][*i*]）。

すべての行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Invert

行列の逆行列を計算する。

プロトタイプ例

```
IppStatus ippmInvert_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight,
        Ipp32f*  pBuffer, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
        dstStride1, Ipp32s dstStride2, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	ソース配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド
srcROIShift	ソース行列内の ROI シフト。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	デスティネーション配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列内の ROI シフト。
widthHeight	行列の次元。
pBuffer	内部計算用に指定されたサイズで割り当てられたバッファへのポインタ。少なくとも *widthHeight*^{2} と同じ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース行列の逆行列を計算し、結果を *pDst* に格納する。完了時には、次の式が成り立つ。

$$dst = src^{-1}$$

ソース行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。
ippStsDivByZeroErr	エラー。ソース行列が特異行列。

FrobNorm

行列の Frobenius ノルムを計算する。

プロトタイプ例

```
IppStatus ippmFrobNorm_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s
        srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
        Ipp32u height, Ipp32f* pDst, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	ソース配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド

srcROIShift	ソース行列内の ROI シフト。
pDst	デスティネーション値へのポインタ。
width	行列の幅。
height	行列の高さ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース行列の Frobenius ノルムを計算し、結果を *pDst* に格納する。デスティネーションの値は、ソース行列のすべての要素を 2 乗した値の和の平方根 (すなわち、以下の式) になる。

$$dst = \sqrt{\sum_{i,j} src[i][j]^2}$$

ソース行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Det

行列の行列式を計算する。

プロトタイプ例

```
IppStatus ippmDet_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight,
        Ipp32f* pBuffer, Ipp32f* pDst, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	ソース配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド
srcROIShift	ソース行列内の ROI シフト。
pDst	デスティネーション値へのポインタ。
widthHeight	行列の次元。
pBuffer	内部計算用に指定されたサイズで割り当てられたバッファへのポインタ。少なくとも *widthHeight*^{2} と同じ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース行列の行列式を計算し、結果を *pDst* に格納する。完了時には、次の式が成り立つ。

dst = det(*src*)

注 : この関数は正方行列以外には使用できないため、行列の次元はいずれも同じ値で、*widthHeight* と同じかそれ以上でなければならない。

Trace

行列のトレースを計算する。

プロトタイプ例

```
IppStatus ippmTrace_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight,
        Ipp32f* pDst, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	ソース配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド。
srcROIShift	ソース行列内の ROI シフト。
pDst	デスティネーション値へのポインタ。
widthHeight	行列の次元。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース行列の主対角線上の要素を合計し、結果を *pDst* に格納する。次の式は、トレースの計算方法を示している。

$$dst = \sum_{i} src[i][i]$$

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

注： この関数は正方行列以外の行列には使用できないため、行列の次元はいずれも同じ値で、*widthHeight* と同じかそれ以上でなければならない。

Mul

行列に定数、ベクトル、または他の
行列を掛ける。

プロトタイプ例

```
IppStatus ippmMul_mac_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
        Ipp32s  srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f*
        pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2,
        Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mava_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
        Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u matrWidth,
        Ipp32u matrHeight, const Ipp32f* pSrc2, Ipp32s src2Stride0,
        Ipp32s src2Stride2, Ipp32u vecLen, Ipp32f* pDst, Ipp32s
        dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
        Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width,
        Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s src2Stride0,
        Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width,
        Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
        dstStride1, Ipp32s
        dstStride2, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix - constant	32f	64f
transposed matrix - constant	32f	64f
matrix array - constant	32f	64f
transposed matrix array - constant	32f	64f
matrix - vector	32f	64f
transposed matrix -vector	32f	64f
matrix - vector array	32f	64f
transposed matrix - vector array	32f	64f
matrix array - vector	32f	64f
transposed matrix array - vector	32f	64f
matrix array - vector array	32f	64f
transposed matrix array - vector array	32f	64f
vector - matrix	32f	64f
vector - transposed matrix	32f	64f

オペランド	データ・タイプ	
vector array - matrix	32f	64f
vector Array - transposed matrix	32f	64f
Vector - matrix Array	32f	64f
vector - transposed matrix array	32f	64f
vector array - matrix array	32f	64f
vector array - transposed matrix array	32f	64f
matrix - matrix	32f	64f
transposed matrix - matrix	32f	64f
matrix - transposed matrix	32f	64f
transposed matrix - transposed matrix	32f	64f
matrix - matrix array	32f	64f
transposed matrix - matrix array	32f	64f
matrix - transposed matrix array	32f	64f
transposed matrix - transposed matrix array	32f	64f
matrix array - matrix	32f	64f
matrix array - transposed matrix	32f	64f
transposed matrix array - transposed matrix	32f	64f
matrix array - matrix array	32f	64f
transposed matrix array - matrix array	32f	64f
matrix array - transposed matrix array	32f	64f
transposed matrix array - transposed matrix array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	ソース配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド
srcROIShift	ソース行列内の ROI シフト。
pSrc1	1 番目のソース行列または行列の配列へのポインタ。
src1Stride0	1 番目のソース配列内の行列間のストライド。
src1Stride1	1 番目のソース行列の行の間のストライド。
src1Stride2	1 番目のソース行列の要素間のストライド。
src1ROIShift	1 番目のソース行列内の ROI シフト。
src1Width	1 番目のソース行列の幅。*src2Height* と同じ値でなければならない。

src1Height	1 番目のソース行列の高さ。
pSrc2	2 番目のソース行列または行列の配列へのポインタ。
src2Stride0	2 番目のソース配列内の行列間のストライド。
src2Stride1	2 番目のソース行列の行の間のストライド。
src2Stride2	2 番目のソース行列の要素間のストライド。
src2ROIShift	2 番目のソース行列内の ROI シフト。
src2Width	2 番目のソース行列の幅。
src2Height	2番目のソース行列の高さ。*src1Width* と同じ値でなければならない。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	デスティネーション配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列内の ROI シフト。
val	行列のスケーリングに使用される乗数。
vecLen	ベクトルの長さ。
width	行列の幅。
height	行列の高さ。
matrWidth	行列の幅。
matrHeight	行列の高さ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数の動作は、他のすべてのインテル® IPP 行列演算関数と同じように、パラメータに依存する。関数名の後に続く、下線の直後のすべての入力パラメータは、関数の実行方法と使用される引数（定数、ベクトル、または他の行列）を指定する。したがって、完全な関数名とともに、リストされた引数の一部だけが入力リスト内で指定され、他の引数は省略される。

定数と行列に対して実行する場合、この関数は、ソース行列の各要素に *val* を掛けてソース行列をスケーリングし、完了時に結果を *pDst* に格納する。

$$mdst[i][j] = c \times msrc[i][j]$$

ソース行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

ベクトルと行列に対して実行する場合、この関数は、ソース行列の行のすべての要素にソース・ベクトルの各要素を掛ける。この演算をソース行列のすべての行に対してループ形式で実行して、デスティネーション・ベクトルを生成し、デスティネーション・ベクトルへのポインタを *pDst* に格納する。行列のすべての行 *i* とベクトルに対して、次の公式が適用される。

$$vdst[i] = \sum_{j} msrc[i][j] \times vsrc[j]$$

反復ごとに *i* は変化せず、*j* はベクトル次元の範囲内のすべての値をとる。

ソース行列の列の数は *matrWidth* と同じかそれ以上、行の数は *matrHeight* と同じかそれ以上でなければならない。ソース・ベクトルの要素の数は、*vecLen* と同じかそれ以上でなければならない。

2 つの行列に対して実行する場合、この関数は、1 番目のソース行列の行の各要素に 2 番目のソース行列の列の各要素を掛けて、得られた積を合計してデスティネーション行列の新しい要素を得る。この演算を 1 番目のソース行列のすべての行と 2 番目のソース行列のすべての列に対してループ形式で実行して、デスティネーション行列全体を生成し、デスティネーション行列へのポインタを *pDst* に格納する。デスティネーション行列の要素について、次の式が成り立つ。

$$mdst[i][j] = \sum_{k} src1[i][k] \times src2[k][j]$$

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Add

行列を他の行列に加算する。

プロトタイプ例

```
IppStatus ippmAdd_mama_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
        Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2,
        Ipp32s  src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
        Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
        dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix - matrix	32f	64f
transposed matrix - matrix	32f	64f
matrix - transposed matrix	32f	64f
transposed matrix - transposed matrix	32f	64f
matrix array - matrix	32f	64f
matrix array - transposed matrix	32f	64f
transposed matrix array - matrix	32f	64f
transposed matrix array - transposed matrix	32f	64f
matrix array - matrix array	32f	64f
transposed matrix array - matrix array	32f	64f
matrix array - transposed matrix array	32f	64f
transposed matrix array - transposed matrix array	32f	64f

全引数のリスト

pSrc1	1 番目のソース行列または行列の配列へのポインタ。
src1Stride0	1 番目のソース配列内の行列間のストライド。
src1Stride1	1 番目のソース行列の行の間のストライド。
src1Stride2	1 番目のソース行列の要素間のストライド。
src1ROIShift	1 番目のソース行列内の ROI シフト。
pSrc2	2 番目のソース行列または行列の配列へのポインタ。
src2Stride0	2 番目のソース配列内の行列間のストライド。
src2Stride1	2 番目のソース行列の行の間のストライド。

src2Stride2	2 番目のソース行列の要素間のストライド。
src2ROIShift	2 番目のソース行列内の ROI シフト。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	デスティネーション配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列内の ROI シフト。
width	行列の幅。
height	行列の高さ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、1 番目のソース行列の各要素と 2 番目のソース行列の各要素を加算し、結果を *pDst* に格納する。

$$dst[i][j] = src1[i][j] + src2[i][j]$$

演算に使用されるすべての行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Sub

行列を他の行列から引く。

プロトタイプ例

```
IppStatus ippmSub_mama_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
        Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2,
        Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
        Ipp32f* pDst, Ipp32s  dstStride0, Ipp32s dstStride1, Ipp32s
        dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix - matrix	32f	64f
transposed matrix - matrix	32f	64f
matrix - transposed matrix	32f	64f
transposed matrix - transposed matrix	32f	64f
matrix - matrix array	32f	64f
transposed matrix -matrix array	32f	64f
matrix - transposed matrix array	32f	64f
transposed matrix - transposed matrix array	32f	64f
matrix array - matrix	32f	64f
matrix array - transposed matrix	32f	64f
transposed matrix array - matrix	32f	64f
transposed matrix array - transposed matrix	32f	64f
matrix array - matrix array	32f	64f
transposed matrix array - matrix array	32f	64f
matrix array - transposed matrix array	32f	64f
transposed matrix array - transposed matrix array	32f	64f

全引数のリスト

pSrc1	1 番目のソース行列または行列の配列へのポインタ。
src1Stride0	1 番目のソース配列内の行列間のストライド。
src1Stride1	1 番目のソース行列の行の間のストライド。
src1Stride2	1 番目のソース行列の要素間のストライド。
src1ROIShift	1 番目のソース行列内の ROI シフト。

pSrc2	2 番目のソース行列または行列の配列へのポインタ。
src2Stride0	2 番目のソース配列内の行列間のストライド。
src2Stride1	2 番目のソース行列の行の間のストライド。
src2Stride2	2 番目のソース行列の要素間のストライド。
src2ROIShift	2 番目のソース行列内の ROI シフト。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	デスティネーション配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列内の ROI シフト。
width	行列の幅。
height	行列の高さ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、1 番目のソース行列の各要素から 2 番目のソース行列の要素を引き、結果を *pDst* に格納する。この演算をすべての行列要素に対してループ形式で実行する。

$$dst[i][j] = src1[i][j] - src2[i][j]$$

演算に使用されるすべての行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

Gaxpy

行列の「GAXPY」演算を実行する。

プロトタイプ例

```
IppStatus ippmGaxpy_va_32f_LS2 (const Ipp32f* pSrc1, Ipp32s
        src1Stride1,
        Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s src2ROIShift,
        Ipp32s src2Stride2, const Ipp32f** pSrc3, Ipp32s src3ROIShift,
        Ipp32s src3Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
        dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
vector	32f	64f
vector array	32f	64f

全引数のリスト

pSrc1	ソース行列または行列の配列へのポインタ。
src1Stride1	1 番目のソース行列の行の間のストライド。
src1Stride2	1 番目のソース行列の要素間のストライド。
src1ROIShift	1 番目のソース行列内の ROI シフト。
pSrc2	1 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	2 番目のソース配列内のベクトル間のストライド。
src2Stride2	2 番目のソース・ベクトルの要素間のストライド。
src2ROIShift	2 番目のソース・ベクトル内の ROI シフト。
pSrc3	2 番目のソース・ベクトルまたはベクトルの配列へのポインタ。
src3Stride0	3 番目のソース配列内のベクトル間のストライド。
src3Stride2	3 番目のソース・ベクトルの要素間のストライド。
src3ROIShift	3 番目のソース・ベクトル内の ROI シフト。
pDst	デスティネーション・ベクトルまたはベクトル配列へのポインタ。

dstStride0	デスティネーション配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトル内の ROI シフト。
width	行列の幅。
height	行列の高さ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、行列の GAXPY 演算を次の手順で実行する。

1. 1 番目のソース行列の行と 2 番目のソース・ベクトルのドット積を計算する。
2. 前の手順の結果を 3 番目のソース・ベクトルの各要素に加算する。
3. 前の手順の結果をデスティネーションの各要素に格納する。

$$t[j] = \sum_{j}(msrc[i][j] \cdot vsrc1[j]) + vsrc2[i]$$

処理される行列のすべての行に対して、この手順を反復する。

演算に使用されるすべての行列の列の数は *width* と同じかそれ以上、行の数は *height* と同じかそれ以上でなければならない。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。入力サイズ・パラメータの値が 0。

連立 1 次方程式の解の関数 6

本章では、正方行列の LU 分解と、後退代入による連立 1 次方程式の解の算出に使用されるインテル® IPP 関数について説明する。

表 6-1 連立 1 次方程式の解の関数

関数の基本名	操作
LUDecomp	正方行列を上三角行列と下三角行列の積に分解する。
LUBackSubst	LU 因数分解された正方行列を使用して連立 1 次方程式を解く。

LUDecomp

正方行列を上三角行列と下三角行列の積に分解する。

プロトタイプ例

```
IppStatus ippmLUDecomp_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
        srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
        pDstIndex, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
        dstStride1, Ipp32s dstStride2, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc ソース行列または行列の配列へのポインタ。

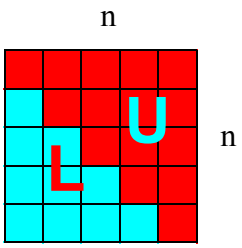
srcStride0 特定の行列配列内の行列間のストライド。

srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド。
srcROIShift	ソース行列の ROI。
index	ピボット・インデックスの配列。行 *i* は行 *index*(*i*) と交換される。配列のサイズは、*widthHeight* と同じかそれより大きいことがある。行列の配列の場合、*index* のサイズは、*count***widthHeight* と同じかそれより大きいことがある。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	特定のデスティネーション行列配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列の ROI。
widthHeight	行列のサイズ。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ソース行列 *pSrc* を 2 つの行列 *L* と *U* の積として表す。*L* は単位対角要素を含む下三角行列、*U* は上三角行列である（図 6-1 を参照）。*L* と *U* はいずれも *pDst* に格納される。行列の対角線より下にある行列要素は、下三角行列 *L* になる。行列 *L* の単位対角要素は格納されない。それ以外の対角要素は、上三角行列 *U* になる。

図 6-1 LU 分解行列の格納



この関数は、必要に応じて行を交換する部分的ピボット演算のアルゴリズムを実行する。

戻り値

ippStsOk	エラーなし。

ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。ソース行列のサイズが 0。
ippStsDivByZeroErr	エラー。ソース行列が特異行列。

LUBackSubst

LU 因数分解された正方行列を使用して連立 1 次方程式を解く。

プロトタイプ例

```
IppStatus ippmLUBackSubst_mva_32f_6x6_PS2 (const Ipp32f** pSrc1,
        Ipp32s src1ROIShift, Ipp32s src1Stride2, Ipp32u* pSrcIndex,
        const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2,
        Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u
        count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix - vector	32f	64f
matrix - vector array	32f	64f

全引数のリスト

pSrc1	ソース行列または行列の配列へのポインタ。LUDecomp の呼び出しの結果でなければならない。
src1Stride0	ソース行列配列内の行列間のストライド。
src1Stride1	ソース行列の行の間のストライド。
src1Stride2	ソース行列の要素間のストライド。
src1ROIShift	ソース行列の ROI。
pSrc2	ソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	ソース・ベクトル配列内のベクトル間のストライド。
src2Stride2	ベクトルの要素間のストライド。
src2ROIShift	ソース・ベクトルの ROI。

index	インデックスの配列。配列のサイズは、*widthHeight* と同じかそれより大きいことがある。行列の配列の場合、*index* のサイズは、*count***widthHeight* と同じかそれより大きいことがある。この配列は、関数 LUDecomp の結果である。
pDst	デスティネーション・ベクトルまたはベクトルの配列へのポインタ。
dstStride0	特定のデスティネーション・ベクトル配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトルの ROI。
widthHeight	行列とベクトルのサイズ。
count	配列内の行列と右辺部分のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、次の連立 1 次方程式を *x* について解く。

$$A \cdot x = L \cdot U \cdot x = b$$

この関数を呼び出す前に、関数 LUDecomp を呼び出して、A の LU 分解を計算する必要がある。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。少なくとも 1 つの入力ポインタが NULL。
ippStsSizeErr	エラー。ソース行列のサイズが 0。

最小 2 乗問題関数

7

本章では、行列の QR 分解の計算と、QR 分解された行列の最小 2 乗（LS）問題の解の算出に使用されるインテル® IPP 関数について説明する。

表 7-1　最小 2 乗問題関数

関数の基本名	操作
QRDecomp	特定の行列の QR 分解を計算する。
QRBackSubst	QR 分解された行列の最小 2 乗問題を解く。

QRDecomp

特定の行列の QR 分解を計算する。

プロトタイプ例

```
IppStatus ippmQRDecomp_ma_32f_LS2 (const Ipp32f** pSrc, Ipp32s
        srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f*
        pBuffer, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
        dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
        Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix	32f	64f
matrix array	32f	64f

全引数のリスト

pSrc　ソース行列または行列の配列へのポインタ。

srcStride0　特定の行列配列内の行列間のストライド。

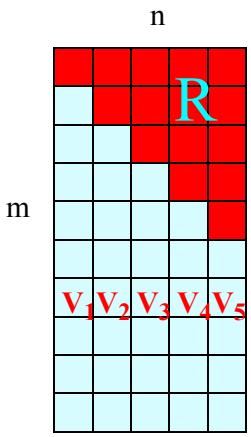
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド。
srcROIShift	ソース行列の ROI。
pBuffer	補助的な配列。この配列のサイズは、*height* と同じかそれより大きいことがある。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	特定のデスティネーション行列配列内の行列間のストライド。
dstStride1	デスティネーション行列の行の間のストライド。
dstStride2	デスティネーション行列の要素間のストライド。
dstROIShift	デスティネーション行列の ROI。
width	ソース行列の列の数。
height	ソース行列の行の数。
count	配列内の行列の数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、ピボット演算を使用せずに、サイズ $m \times n$ の一般行列 A の QR 分解を計算する（m は n と同じかそれより大きい）。

この関数は、行列 Q を直接には作成しない。Q は、n 個のハウスホルダ・ベクトルの累積的な積として表現される。行列の対角線より下にある行列要素は、ハウスホルダ・ベクトル V_n になる。これらのベクトルの最初の成分は、値が 1 であるために格納されない。それ以外の行列要素は、上三角行列 R になる（図 7-1 を参照）。

図 7-1　QR 分解行列の格納



戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。入力ポインタのうち 1 つが NULL。
ippStsSizeErr	エラー。ソース行列のサイズが 0。
ippStsDivByZeroErr	エラー。ソース行列の列の階数が不完全。

QRBackSubst

QR 分解された行列の最小 2 乗問題を解く。

プロトタイプ例

```
IppStatus ippmQRBackSubst_mva_64f_6x6_LS2 (const Ipp64f** pSrc1, Ipp32s
        src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp64f*
        pBuffer, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
        src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
        dstStride2, Ipp32u count);
```

特定のプロトタイプを見つけるには、表中の対応するリンクを参照のこと。

オペランド	データ・タイプ	
matrix - vector	32f	64f
matrix - vector array	32f	64f

全引数のリスト

pSrc	ソース行列または行列の配列へのポインタ。
srcStride0	特定の行列配列内の行列間のストライド。
srcStride1	ソース行列の行の間のストライド。
srcStride2	ソース行列の要素間のストライド。
srcROIShift	ソース行列の ROI。
pSrc2	ソース・ベクトルまたはベクトルの配列へのポインタ。
src2Stride0	特定のベクトル配列内のベクトル間のストライド。
src2Stride2	ベクトルの要素間のストライド。
src2ROIShift	ソース行列の ROI。
pBuffer	補助的な配列。この配列のサイズは、*height* と同じかそれより大きいことがある。
pDst	デスティネーション行列または行列の配列へのポインタ。
dstStride0	特定のデスティネーション・ベクトル配列内のベクトル間のストライド。
dstStride2	デスティネーション・ベクトルの要素間のストライド。
dstROIShift	デスティネーション・ベクトルの ROI。
width	ソース行列の列の数。
height	ソース行列の行の数。
count	配列内の行列と右辺部分のベクトルの数。

説明

この関数は、ippm.h ヘッダ・ファイルで宣言される。この関数は、次の最小 2 乗問題を解く。

$$\|A \cdot x - b\|_2^2 = \|R \cdot x - c\|_2^2 + \|d\|_2^2$$

A は連立 1 次方程式 $A \cdot x = b$ の行列、x は不明な変数ベクトル、b は連立方程式の右辺のベクトル、R は上三角行列（図 7-1 を参照）であり、次の式が成り立つ。

$$Q^T \cdot b = \begin{bmatrix} c \\ d \end{bmatrix},$$

Q は直交行列である。

連立方程式の式の数は、不明な変数の数と同じかそれより大きい。

この関数を呼び出す前に、関数 QRDecomp を呼び出して、*A* の QR 分解を計算する必要がある。

戻り値

ippStsOk	エラーなし。
ippStsNullPtrErr	エラー。入力ポインタのうち 1 つが NULL。
ippStsSizeErr	エラー。ソース行列のサイズが 0。

ユーティリティ関数のプロトタイプ

A

Copy

Copy 関数は、ある位置から別の位置にデータを移動する。この関数の説明とすべての引数のリストは、Copy を参照のこと。

Copy 関数 : 32f データのベクトル配列の演算

```
IppStatus ippmCopy_va_32f_SS (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_SP (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_SL (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_LS (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_PS (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_LP (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_LL (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_PP (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_32f_PL (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
```

Copy 関数 : 64f データのベクトル配列の演算

```
IppStatus ippmCopy_va_64f_SS (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_SP (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_SL (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_LS (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_PS (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_LP (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_LL (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_PP (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmCopy_va_64f_PL (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
```

Copy 関数 : 32f データの行列配列の演算

```
IppStatus ippmCopy_ma_32f_SS (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_32f_SP (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmCopy_ma_32f_SL (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
```

```
IppStatus ippmCopy_ma_32f_LS (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_32f_PS (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmCopy_ma_32f_LP (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmCopy_ma_32f_LL (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_32f_PP (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_32f_PL (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp32f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
```

Copy 関数 : 64f データの行列配列の演算

```
IppStatus ippmCopy_ma_64f_SS (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_64f_SP (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmCopy_ma_64f_SL (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_64f_LS (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_64f_PS (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
```

```
IppStatus ippmCopy_ma_64f_LP (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmCopy_ma_64f_LL (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_64f_PP (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmCopy_ma_64f_PL (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp32s srcStride2, Ipp64f** pDst, Ipp32u dstROIShift, Ipp32s
    dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
```

Extract

Extract 関数は、ROI (Region of Interest) を抽出する。この関数の説明とすべての引数のリストは、Extract を参照のこと。

Extract 関数 : 32f データのベクトルの演算

```
IppStatus ippmExtract_v_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f* pDst, Ipp32u len);
IppStatus ippmExtract_v_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst, Ipp32u len);
```

Extract 関数 : 64f データのベクトルの演算

```
IppStatus ippmExtract_v_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f* pDst, Ipp32u len);
IppStatus ippmExtract_v_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst, Ipp32u len);
```

Extract 関数 : 32f データのベクトル配列の演算

```
IppStatus ippmExtract_va_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmExtract_va_32f_PS2 (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmExtract_va_32f_LS2 (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
```

Extract 関数 : 64f データのベクトル配列の演算

```
IppStatus ippmExtract_va_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmExtract_va_64f_PS2 (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmExtract_va_64f_LS2 (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u len, Ipp32u
    count);
```

Extract 関数 : 32f データの行列の演算

```
IppStatus ippmExtract_m_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32u matrWidth, Ipp32u
    matrHeight);
IppStatus ippmExtract_m_32f_P (const Ipp32f** pSrc, Ipp32u srcROIShift,
    Ipp32f* pDst, Ipp32u matrWidth, Ipp32u matrHeight);
```

Extract 関数 : 64f データの行列の演算

```
IppStatus ippmExtract_m_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32u matrWidth, Ipp32u
    matrHeight);
IppStatus ippmExtract_m_64f_P (const Ipp64f** pSrc, Ipp32u srcROIShift,
    Ipp64f* pDst, Ipp32u matrWidth, Ipp32u matrHeight);
```

Extract 関数 : 32f データの転置行列の演算

```
IppStatus ippmExtract_mT_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32u matrWidth, Ipp32u
    matrHeight);
IppStatus ippmExtract_mT_32f_P (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32f* pDst, Ipp32u matrWidth, Ipp32u matrHeight);
```

Extract 関数 : 64f データの転置行列の演算

```
IppStatus ippmExtract_mT_64f_P (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp64f* pDst, Ipp32u matrWidth, Ipp32u matrHeight);
```

Extract 関数 : 32f データの行列配列の演算

```
IppStatus ippmExtract_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u
    matrWidth, Ipp32u matrHeight, Ipp32u count);
IppStatus ippmExtract_ma_32f_PS2 (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u matrWidth,
    Ipp32u matrHeight, Ipp32u count);
IppStatus ippmExtract_ma_32f_LS2 (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u matrWidth, Ipp32u matrHeight, Ipp32u count);
```

Extract 関数 : 64f データの行列配列の演算

```
IppStatus ippmExtract_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u
    matrWidth, Ipp32u matrHeight, Ipp32u count);
IppStatus ippmExtract_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u matrWidth,
    Ipp32u matrHeight, Ipp32u count);
IppStatus ippmExtract_ma_64f_LS2 (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u matrWidth, Ipp32u matrHeight, Ipp32u count);
```

Extract 関数 : 32f データの転置行列配列の演算

```
IppStatus ippmExtract_maT_32f_PS2 (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u matrWidth,
    Ipp32u matrHeight, Ipp32u count);
IppStatus ippmExtract_maT_32f_LS2 (const Ipp32f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u matrWidth, Ipp32u matrHeight, Ipp32u count);
```

Extract 関数 : 64f データの転置行列配列の演算

```
IppStatus ippmExtract_maT_64f_PS2 (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u matrWidth,
    Ipp32u matrHeight, Ipp32u count);
IppStatus ippmExtract_maT_64f_LS2 (const Ipp64f** pSrc, Ipp32u
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u matrWidth, Ipp32u matrHeight, Ipp32u count);
```

LoadIdentity

LoadIdentity 関数は、恒等行列を初期化する。この関数の説明とすべての引数のリストは、LoadIdentity を参照のこと。

LoadIdentity 関数 : 32f データの行列配列の演算

```
IppStatus ippmLoadIdentity_ma_32f_S (const Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32s dstWidth,
    Ipp32s dstHeight, Ipp32u count);
IppStatus ippmLoadIdentity_ma_64f_S (const Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32s dstWidth,
    Ipp32s dstHeight, Ipp32u count);
IppStatus ippmLoadIdentity_ma_32f_L (const Ipp32f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32s dstWidth,
    Ipp32s dstHeight, Ipp32u count);
```

LoadIdentity 関数 : 64f データの行列配列の演算

```
IppStatus ippmLoadIdentity_ma_64f_L (const Ipp64f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32s dstWidth,
    Ipp32s dstHeight, Ipp32u count);
IppStatus ippmLoadIdentity_ma_32f_P (const Ipp32f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride2, Ipp32s dstWidth, Ipp32s dstHeight,
    Ipp32u count);
IppStatus ippmLoadIdentity_ma_64f_P (const Ipp64f** pDst, Ipp32u
    dstROIShift, Ipp32s dstStride2, Ipp32s dstWidth, Ipp32s dstHeight,
    Ipp32u count);
```

ベクトル代数関数のプロトタイプ

B

Saxpy

Saxpy 関数は、ベクトルの SAXPY 演算を実行する。この関数の説明とすべての引数のリストは、Saxpy を参照のこと。

Saxpy 関数 : 32f データのベクトル - ベクトルの演算

```
IppStatus ippmSaxpy_vv_32f (const Ipp32f* pSrc1, Ipp32f scale, const
    Ipp32f* pSrc2, Ipp32f* pDst, Ipp32u len);
IppStatus ippmSaxpy_vv_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride2,
    Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f*
    pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmSaxpy_vv_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmSaxpy_vv_32f_3x1 (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmSaxpy_vv_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSaxpy_vv_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSaxpy_vv_32f_4x1 (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmSaxpy_vv_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSaxpy_vv_32f_4x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSaxpy_vv_32f_5x1 (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmSaxpy_vv_32f_5x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride2);
```

```
IppStatus ippmSaxpy_vv_32f_5x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSaxpy_vv_32f_6x1 (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmSaxpy_vv_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSaxpy_vv_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
```

Saxpy 関数 : 64f データのベクトル - ベクトルの演算

```
IppStatus ippmSaxpy_vv_64f (const Ipp64f* pSrc1, Ipp64f scale, const
    Ipp64f* pSrc2, Ipp64f* pDst, Ipp32u len);
IppStatus ippmSaxpy_vv_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride2,
    Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f*
    pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmSaxpy_vv_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmSaxpy_vv_64f_3x1 (const Ipp64f* pSrc1, Ipp64f scale,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmSaxpy_vv_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSaxpy_vv_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSaxpy_vv_64f_4x1 (const Ipp64f* pSrc1, Ipp64f scale,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmSaxpy_vv_64f_4x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSaxpy_vv_64f_4x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSaxpy_vv_64f_5x1 (const Ipp64f* pSrc1, Ipp64f scale,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmSaxpy_vv_64f_5x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride2);
```

```
IppStatus ippmSaxpy_vv_64f_5x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSaxpy_vv_64f_6x1 (const Ipp64f* pSrc1, Ipp64f scale,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmSaxpy_vv_64f_6x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSaxpy_vv_64f_6x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
```

Saxpy 関数 : 32f データのベクトル - ベクトル配列の演算

```
IppStatus ippmSaxpy_vva_32f (const Ipp32f* pSrc1, Ipp32f scale, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride0,
    Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmSaxpy_vva_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_L (const Ipp32f* pSrc1, Ipp32f scale, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_3x1 (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride0,
    Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmSaxpy_vva_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_3x1_L** (const Ipp32f* *pSrc1*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_3x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_4x1_L** (const Ipp32f* *pSrc1*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_4x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSaxpy_vva_32f_5x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_5x1_L (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_5x1_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_6x1 (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride0,
    Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_6x1_L (const Ipp32f* pSrc1, Ipp32f scale,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vva_32f_6x1_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Saxpy 関数 : 64f データのベクトル - ベクトル配列の演算

```
IppStatus ippmSaxpy_vva_64f (const Ipp64f* pSrc1, Ipp64f scale, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride0,
    Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vva_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmSaxpy_vva_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_L** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_3x1** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_3x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_3x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_4x1** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_4x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_4x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_5x1** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_5x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_5x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_6x1** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_6x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vva_64f_6x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Saxpy 関数 : 32f データのベクトル配列 - ベクトルの演算

IppStatus **ippmSaxpy_vav_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_3x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_3x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_3x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_3x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSaxpy_vav_32f_5x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_5x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f* pSrc2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_5x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_6x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32f scale, const Ipp32f* pSrc2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f* pSrc2,
    Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_6x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f* pSrc2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vav_32f_6x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
```

Saxpy 関数 : 64f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmSaxpy_vav_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp64f scale, const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vav_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp64f scale, const Ipp64f* pSrc2,
    Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vav_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmSaxpy_vav_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_6x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vav_64f_6x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Saxpy 関数 : 32f データのベクトル配列 - ベクトル配列の演算

IppStatus **ippmSaxpy_vava_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_3x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_3x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_3x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_3x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSaxpy_vava_32f_5x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_5x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_5x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_6x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32f scale, const Ipp32f* pSrc2, Ipp32s src2Stride0,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f* pSrc2,
    Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_6x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vava_32f_6x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Saxpy 関数 : 64f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmSaxpy_vava_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp64f scale, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f*
    pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vava_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp64f scale, const Ipp64f* pSrc2,
    Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSaxpy_vava_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmSaxpy_vava_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSaxpy_vava_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSaxpy_vava_64f_6x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f scale, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSaxpy_vava_64f_6x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSaxpy_vava_64f_6x1_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f scale, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Add

Add 関数は、定数をベクトルに加算するか、ベクトルを他のベクトルに加算する。この関数の説明とすべての引数のリストは、Add を参照のこと。

Add 関数 : 32f データのベクトル - ベクトルの演算

```
IppStatus ippmAdd_vv_32f (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst, Ipp32u len);
IppStatus ippmAdd_vv_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride2,
    const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride2, Ipp32u len);
IppStatus ippmAdd_vv_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u len);
IppStatus ippmAdd_vv_32f_3x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmAdd_vv_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmAdd_vv_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_vv_32f_4x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmAdd_vv_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmAdd_vv_32f_4x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_vv_32f_5x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
```

```
IppStatus ippmAdd_vv_32f_5x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmAdd_vv_32f_5x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_vv_32f_6x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmAdd_vv_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmAdd_vv_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Add 関数 : 64f データのベクトル - ベクトルの演算

```
IppStatus ippmAdd_vv_64f (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst, Ipp32u len);
IppStatus ippmAdd_vv_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride2,
    const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s
    dstStride2, Ipp32u len);
IppStatus ippmAdd_vv_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u len);
IppStatus ippmAdd_vv_64f_3x1 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst);
IppStatus ippmAdd_vv_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmAdd_vv_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_vv_64f_4x1 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst);
IppStatus ippmAdd_vv_64f_4x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmAdd_vv_64f_4x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_vv_64f_5x1 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst);
```

IppStatus **ippmAdd_vv_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_vv_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_vv_64f_6x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*);

IppStatus **ippmAdd_vv_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_vv_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Add 関数 : 32f データのベクトル配列 - ベクトルの演算

IppStatus **ippmAdd_vav_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_3x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_3x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_3x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_3x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmAdd_vav_32f_5x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmAdd_vav_32f_5x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmAdd_vav_32f_5x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmAdd_vav_32f_6x1 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmAdd_vav_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmAdd_vav_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_vav_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmAdd_vav_32f_6x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmAdd_vav_32f_6x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
```

Add 関数 : 64f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmAdd_vav_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u len,
    Ipp32u count);
IppStatus ippmAdd_vav_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmAdd_vav_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u len, Ipp32u count);
```

IppStatus **ippmAdd_vav_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_6x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_6x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データのベクトル配列 - ベクトル配列の演算

IppStatus **ippmAdd_vava_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

```
IppStatus ippmAdd_vava_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmAdd_vava_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_vava_32f_3x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_32f_3x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_vava_32f_3x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_32f_4x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmAdd_vava_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_32f_4x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmAdd_vava_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_5x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_5x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_6x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_6x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_32f_6x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmAdd_vava_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_32f_6x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_vava_32f_6x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Add 関数 : 64f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmAdd_vava_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmAdd_vava_64f_3x1 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmAdd_vava_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmAdd_vava_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vav_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_vava_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmAdd_vava_64f_6x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_vava_64f_6x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_vava_64f_6x1_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Sub

Sub 関数は、ベクトルから定数を引くか、他のベクトルからベクトルを引く。この関数の説明とすべての引数のリストは、Sub を参照のこと。

Sub 関数 : 32f データのベクトル - 定数の演算

```
IppStatus ippmSub_vc_32f (const Ipp32f* pSrc, Ipp32f val, Ipp32f* pDst,
    Ipp32u len);
IppStatus ippmSub_vc_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmSub_vc_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmSub_vc_32f_3x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_vc_32f_3x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_32f_3x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vc_32f_4x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_vc_32f_4x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_32f_4x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vc_32f_5x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_vc_32f_5x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_32f_5x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
```

```
IppStatus ippmSub_vc_32f_6x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_vc_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
```

Sub 関数 : 64f データのベクトル - 定数の演算

```
IppStatus ippmSub_vc_64f (const Ipp64f* pSrc, Ipp64f val, Ipp64f* pDst,
    Ipp32u len);
IppStatus ippmSub_vc_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmSub_vc_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmSub_vc_64f_3x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_vc_64f_3x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_64f_3x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vc_64f_4x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_vc_64f_4x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_64f_4x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vc_64f_5x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_vc_64f_5x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_64f_5x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vc_64f_6x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_vc_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_vc_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

Sub 関数 : 32f データのベクトル配列 - 定数の演算

```
IppStatus ippmSub_vac_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmSub_vac_32f_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_3x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_3x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_3x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_3x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_3x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_3x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_4x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_4x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_4x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vac_32f_4x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmSub_vac_32f_4x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_32f_4x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_32f_5x1 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_vac_32f_5x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_32f_5x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_32f_5x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_32f_5x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_32f_5x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_32f_6x1 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_vac_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_32f_6x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_32f_6x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_32f_6x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 64f データのベクトル配列 - 定数の演算

```
IppStatus ippmSub_vac_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmSub_vac_64f_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_3x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_3x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_3x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_3x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_3x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_3x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_4x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_4x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_4x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vac_64f_4x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmSub_vac_64f_4x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_64f_4x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_64f_5x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_vac_64f_5x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_64f_5x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_64f_5x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_64f_5x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_64f_5x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_64f_6x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_vac_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_64f_6x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vac_64f_6x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_vac_64f_6x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 32f データの定数 - ベクトルの演算

```
IppStatus ippmSub_cv_32f (const Ipp32f* pSrc, Ipp32f val, Ipp32f* pDst,
    Ipp32u len);
```

```
IppStatus ippmSub_cv_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmSub_cv_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmSub_cv_32f_3x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_cv_32f_3x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_32f_3x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_cv_32f_4x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_cv_32f_4x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_32f_4x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_cv_32f_5x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_cv_32f_5x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_32f_5x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_cv_32f_6x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmSub_cv_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
```

Sub 関数 : 64f データの定数 - ベクトルの演算

```
IppStatus ippmSub_cv_64f (const Ipp64f* pSrc, Ipp64f val, Ipp64f* pDst,
    Ipp32u len);
IppStatus ippmSub_cv_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmSub_cv_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmSub_cv_64f_4x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_cv_64f_4x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_64f_4x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

```
IppStatus ippmSub_cv_64f_5x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_cv_64f_5x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_64f_5x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmSub_cv_64f_6x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmSub_cv_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmSub_cv_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

Sub 関数 : 32f データの定数 - ベクトル配列の演算

```
IppStatus ippmSub_cva_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u
    count);
IppStatus ippmSub_cva_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_cva_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmSub_cva_32f_PS2 (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_cva_32f_L (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmSub_cva_32f_LS2 (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_cva_32f_3x1 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_cva_32f_3x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_cva_32f_3x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_cva_32f_3x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmSub_cva_32f_3x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_3x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_4x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_4x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_4x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_4x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_4x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_4x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_5x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_5x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_5x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_5x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_5x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_5x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_32f_6x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmSub_cva_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_cva_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_cva_32f_6x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_cva_32f_6x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_cva_32f_6x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 64f データの定数 - ベクトル配列の演算

```
IppStatus ippmSub_cva_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u
    count);
IppStatus ippmSub_cva_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_cva_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmSub_cva_64f_PS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_cva_64f_L (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmSub_cva_64f_LS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_cva_64f_3x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_cva_64f_3x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_cva_64f_3x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
```

IppStatus **ippmSub_cva_64f_3x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_3x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_3x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_4x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_4x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_4x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_4x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_4x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_4x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_5x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_5x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_5x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_5x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_5x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_cva_64f_5x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmSub_cva_64f_6x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_cva_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_cva_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_cva_64f_6x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_cva_64f_6x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmSub_cva_64f_6x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 32f データのベクトル - ベクトルの演算

```
IppStatus ippmSub_vv_32f (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst, Ipp32u len);
IppStatus ippmSub_vv_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride2,
    const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride2, Ipp32u len);
IppStatus ippmSub_vv_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u len);
IppStatus ippmSub_vv_32f_3x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmSub_vv_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmSub_vv_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vv_32f_4x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmSub_vv_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmSub_vv_32f_4x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

```
IppStatus ippmSub_vv_32f_5x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmSub_vv_32f_5x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmSub_vv_32f_5x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vv_32f_6x1 (const Ipp32f* pSrc1, const Ipp32f* pSrc2,
    Ipp32f* pDst);
IppStatus ippmSub_vv_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmSub_vv_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 64f データのベクトル - ベクトルの演算

```
IppStatus ippmSub_vv_64f (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst, Ipp32u len);
IppStatus ippmSub_vv_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride2,
    const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s
    dstStride2, Ipp32u len);
IppStatus ippmSub_vv_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u len);
IppStatus ippmSub_vv_64f_3x1 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst);
IppStatus ippmSub_vv_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmSub_vv_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vv_64f_4x1 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst);
IppStatus ippmSub_vv_64f_4x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmSub_vv_64f_4x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_vv_64f_5x1 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp64f* pDst);
```

IppStatus **ippmSub_vv_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmSub_vv_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmSub_vv_64f_6x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*);

IppStatus **ippmSub_vv_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmSub_vv_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Sub 関数 : 32f データのベクトル - ベクトル配列の演算

IppStatus **ippmSub_vva_32f** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_3x1** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_3x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_3x1_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_3x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_4x1** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_4x1_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_4x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_5x1** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_5x1_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_5x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_6x1** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_6x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_6x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_6x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_6x1_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_32f_6x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データのベクトル - ベクトル配列の演算

IppStatus **ippmSub_vva_64f** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_3x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_3x1_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_3x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_4x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_4x1_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_4x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_5x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_5x1_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_5x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_6x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_6x1_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vva_64f_6x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データのベクトル配列 - ベクトルの演算

IppStatus **ippmSub_vav_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_3x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_3x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_3x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_3x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_vav_32f_5x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_vav_32f_5x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmSub_vav_32f_5x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_vav_32f_6x1 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmSub_vav_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_vav_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_vav_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_vav_32f_6x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmSub_vav_32f_6x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
```

Sub 関数 : 64f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmSub_vav_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u len,
    Ipp32u count);
IppStatus ippmSub_vav_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmSub_vav_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u len, Ipp32u count);
```

IppStatus **ippmSub_vav_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vav_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_vav_64f_6x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_vav_64f_6x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f* pSrc2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmSub_vav_64f_6x1_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
```

Sub 関数 : 32f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmSub_vava_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_vava_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vava_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmSub_vava_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_3x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_3x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_vava_32f_5x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vava_32f_5x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_vava_32f_5x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vava_32f_6x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmSub_vava_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vava_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_vava_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vava_32f_6x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_vava_32f_6x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 64f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmSub_vava_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u len, Ipp32u count);
IppStatus ippmSub_vava_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
```

IppStatus **ippmSub_vava_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_vava_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_vava_64f_6x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_vava_64f_6x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_vava_64f_6x1_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

CrossProduct

CrossProduct 関数は、2 つのベクトルのクロス積を計算する。この関数の説明とすべての引数のリストは、CrossProduct を参照のこと。

CrossProduct 関数 : 32f データのベクトル - ベクトルの演算

```
IppStatus ippmCrossProduct_vv_32f_3x1 (const Ipp32f* pSrc1, const
    Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmCrossProduct_vv_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmCrossProduct_vv_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

CrossProduct 関数 : 64f データのベクトル - ベクトルの演算

```
IppStatus ippmCrossProduct_vv_64f_3x1 (const Ipp64f* pSrc1, const
    Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmCrossProduct_vv_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmCrossProduct_vv_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

CrossProduct 関数 : 32f データのベクトル - ベクトル配列の演算

```
IppStatus ippmCrossProduct_vva_32f_3x1 (const Ipp32f* pSrc1, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32u count);
IppStatus ippmCrossProduct_vva_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
```

IppStatus **ippmCrossProduct_vva_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_32f_3x1_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_32f_3x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

CrossProduct 関数 : 64f データのベクトル - ベクトル配列の演算

IppStatus **ippmCrossProduct_vva_64f_3x1** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_64f_3x1_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vva_64f_3x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmCrossProduct_vav_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

CrossProduct 関数 : 32f データのベクトル配列 - ベクトルの演算

IppStatus **ippmCrossProduct_vav_32f_3x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmCrossProduct_vav_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmCrossProduct_vav_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vav_32f_3x1_PS2 (const Ipp32f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2,
    Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmCrossProduct_vav_32f_3x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vav_32f_3x1_LS2 (const Ipp32f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2,
    Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
```

CrossProduct 関数 : 64f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmCrossProduct_vav_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmCrossProduct_vav_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vav_64f_3x1_PS2 (const Ipp64f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2,
    Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmCrossProduct_vav_64f_3x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f* pSrc2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vav_64f_3x1_LS2 (const Ipp64f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp64f* pSrc2,
    Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
```

CrossProduct 関数 : 32f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmCrossProduct_vava_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
```

```
IppStatus ippmCrossProduct_vava_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmCrossProduct_vava_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vava_32f_3x1_PS2 (const Ipp32f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2,
    Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmCrossProduct_vava_32f_3x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vava_32f_3x1_LS2 (const Ipp32f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2,
    Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

CrossProduct 関数 : 64f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmCrossProduct_vava_64f_3x1 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmCrossProduct_vava_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmCrossProduct_vava_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vava_64f_3x1_PS2 (const Ipp64f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2,
    Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmCrossProduct_vava_64f_3x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmCrossProduct_vava_64f_3x1_LS2 (const Ipp64f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2,
    Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

DotProduct

DotProduct 関数は、2 つのベクトルのドット積を計算する。この関数の説明とすべての引数のリストは、DotProduct を参照のこと。

DotProduct 関数 : 32f データのベクトル - ベクトルの演算

```
IppStatus ippmDotProduct_vv_32f (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32f* pDst, Ipp32u len);
IppStatus ippmDotProduct_vv_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32u len);
IppStatus ippmDotProduct_vv_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u len);
IppStatus ippmDotProduct_vv_32f_3x1 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32f* pDst);
IppStatus ippmDotProduct_vv_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f*
    pDst);
IppStatus ippmDotProduct_vv_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst);
IppStatus ippmDotProduct_vv_32f_4x1 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32f* pDst);
IppStatus ippmDotProduct_vv_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f*
    pDst);
IppStatus ippmDotProduct_vv_32f_4x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst);
IppStatus ippmDotProduct_vv_32f_5x1 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32f* pDst);
IppStatus ippmDotProduct_vv_32f_5x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f*
    pDst);
IppStatus ippmDotProduct_vv_32f_5x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst);
IppStatus ippmDotProduct_vv_32f_6x1 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32f* pDst);
IppStatus ippmDotProduct_vv_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f*
    pDst);
```

```
IppStatus ippmDotProduct_vv_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst);
```

DotProduct 関数 : 64f データのベクトル - ベクトルの演算

```
IppStatus ippmDotProduct_vv_64f (const Ipp64f* pSrc1, const Ipp64f*
    pSrc2, Ipp64f* pDst, Ipp32u len);
IppStatus ippmDotProduct_vv_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32u len);
IppStatus ippmDotProduct_vv_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst, Ipp32u len);
IppStatus ippmDotProduct_vv_64f_3x1 (const Ipp64f* pSrc1, const Ipp64f*
    pSrc2, Ipp64f* pDst);
IppStatus ippmDotProduct_vv_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f*
    pDst);
IppStatus ippmDotProduct_vv_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst);
IppStatus ippmDotProduct_vv_64f_4x1 (const Ipp64f* pSrc1, const Ipp64f*
    pSrc2, Ipp64f* pDst);
IppStatus ippmDotProduct_vv_64f_4x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f*
    pDst);
IppStatus ippmDotProduct_vv_64f_4x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst);
IppStatus ippmDotProduct_vv_64f_5x1 (const Ipp64f* pSrc1, const Ipp64f*
    pSrc2, Ipp64f* pDst);
IppStatus ippmDotProduct_vv_64f_5x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f*
    pDst);
IppStatus ippmDotProduct_vv_64f_5x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst);
IppStatus ippmDotProduct_vv_64f_6x1 (const Ipp64f* pSrc1, const Ipp64f*
    pSrc2, Ipp64f* pDst);
IppStatus ippmDotProduct_vv_64f_6x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp64f*
    pDst);
```

```
IppStatus ippmDotProduct_vv_64f_6x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst);
```

DotProduct 関数 : 32f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmDotProduct_vav_32f (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vav_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vav_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_3x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_3x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_3x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_4x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32u count);
```

IppStatus **ippmDotProduct_vav_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_5x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_5x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_6x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_6x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_6x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_6x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_6x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_32f_6x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

DotProduct 関数 : 64f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmDotProduct_vav_64f (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vav_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vav_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_3x1 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_3x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_3x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_3x1_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_4x1 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, const Ipp64f* pSrc2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_4x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vav_64f_4x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst, Ipp32u count);
```

IppStatus **ippmDotProduct_vav_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_6x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vav_64f_6x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

DotProduct 関数 : 32f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmDotProduct_vava_32f (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vava_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vava_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u len, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmDotProduct_vava_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmDotProduct_vava_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_3x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_3x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f*
    pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_3x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_32f_4x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst,
    Ipp32u count);
```

IppStatus **ippmDotProduct_vava_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_5x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_5x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_6x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_6x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_6x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_6x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_6x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_32f_6x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32u *count*);

DotProduct 関数 : 64f データのベクトル配列 - ベクトル配列の演算

IppStatus **ippmDotProduct_vava_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmDotProduct_vava_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32u *count*);

```
IppStatus ippmDotProduct_vava_64f_6x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_64f_6x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_64f_6x1_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f*
    pDst, Ipp32u count);
IppStatus ippmDotProduct_vava_64f_6x1_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32u count);
```

L2Norm

L2Norm 関数は、ベクトルの L2 ノルムを計算する。この関数の説明とすべての引数のリストは、L2Norm を参照のこと。

L2Norm 関数 : 32f データのベクトルの演算

```
IppStatus ippmL2Norm_v_32f (const Ipp32f* pSrc, Ipp32f* pDst, Ipp32u
    len);
IppStatus ippmL2Norm_v_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f* pDst, Ipp32u len);
IppStatus ippmL2Norm_v_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst, Ipp32u len);
IppStatus ippmL2Norm_v_32f_3x1 (const Ipp32f* pSrc, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_3x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride2, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_3x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_4x1 (const Ipp32f* pSrc, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_4x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride2, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_4x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_5x1 (const Ipp32f* pSrc, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_5x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride2, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_5x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_6x1 (const Ipp32f* pSrc, Ipp32f* pDst);
```

```
IppStatus ippmL2Norm_v_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride2, Ipp32f* pDst);
IppStatus ippmL2Norm_v_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
```

L2Norm 関数 : 64f データのベクトルの演算

```
IppStatus ippmL2Norm_v_64f (const Ipp64f* pSrc, Ipp64f* pDst, Ipp32u
    len);
IppStatus ippmL2Norm_v_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f* pDst, Ipp32u len);
IppStatus ippmL2Norm_v_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst, Ipp32u len);
IppStatus ippmL2Norm_v_64f_3x1 (const Ipp64f* pSrc, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_3x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride2, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_3x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_4x1 (const Ipp64f* pSrc, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_4x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride2, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_4x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_5x1 (const Ipp64f* pSrc, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_5x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride2, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_5x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_6x1 (const Ipp64f* pSrc, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride2, Ipp64f* pDst);
IppStatus ippmL2Norm_v_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
```

L2Norm 関数 : 32f データのベクトル配列の演算

```
IppStatus ippmL2Norm_va_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_32f_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmL2Norm_va_32f_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_3x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_3x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_3x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_3x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_3x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_3x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_4x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_4x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_4x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_4x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_4x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_4x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_5x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_5x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_5x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_5x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_5x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_5x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmL2Norm_va_32f_6x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f* *pDst*, Ipp32u *count*);

```
IppStatus ippmL2Norm_va_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_32f_6x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_32f_6x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_32f_6x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
```

L2Norm 関数 : 64f データのベクトル配列の演算

```
IppStatus ippmL2Norm_va_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_64f_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmL2Norm_va_64f_L (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst, Ipp32u len, Ipp32u count);
IppStatus ippmL2Norm_va_64f_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u len, Ipp32u
    count);
IppStatus ippmL2Norm_va_64f_3x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_3x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_3x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_3x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_3x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_3x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_4x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_4x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
```

```
IppStatus ippmL2Norm_va_64f_4x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_4x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_4x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_4x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_5x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_5x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_5x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_5x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_5x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_5x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_6x1 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_6x1_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_6x1_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmL2Norm_va_64f_6x1_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
```

LComb

LComb 関数は、2 つのベクトルの線形結合を構成する。この関数の説明とすべての引数のリストは、LComb を参照のこと。

LComb 関数 : 32f データのベクトル - ベクトルの演算

```
IppStatus ippmLComb_vv_32f (const Ipp32f* pSrc1, Ipp32f scale1, const
    Ipp32f* pSrc2, Ipp32f scale2, Ipp32f* pDst, Ipp32u len);
```

```
IppStatus ippmLComb_vv_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride2,
    Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32f
    scale2, Ipp32f* pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmLComb_vv_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u len);
IppStatus ippmLComb_vv_32f_3x1 (const Ipp32f* pSrc1, Ipp32f scale1,
    const Ipp32f* pSrc2, Ipp32f scale2, Ipp32f* pDst);
IppStatus ippmLComb_vv_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f scale2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLComb_vv_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmLComb_vv_32f_4x1 (const Ipp32f* pSrc1, Ipp32f scale1,
    const Ipp32f* pSrc2, Ipp32f scale2, Ipp32f* pDst);
IppStatus ippmLComb_vv_32f_4x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f scale2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLComb_vv_32f_4x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmLComb_vv_32f_5x1 (const Ipp32f* pSrc1, Ipp32f scale1,
    const Ipp32f* pSrc2, Ipp32f scale2, Ipp32f* pDst);
IppStatus ippmLComb_vv_32f_5x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f scale2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLComb_vv_32f_5x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmLComb_vv_32f_6x1 (const Ipp32f* pSrc1, Ipp32f scale1,
    const Ipp32f* pSrc2, Ipp32f scale2, Ipp32f* pDst);
IppStatus ippmLComb_vv_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f scale2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLComb_vv_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift);
```

LComb 関数 : 64f データのベクトル - ベクトルの演算

```
IppStatus ippmLComb_vv_64f (const Ipp64f* pSrc1, Ipp64f scale1, const
    Ipp64f* pSrc2, Ipp64f scale2, Ipp64f* pDst, Ipp32u len);
```

IppStatus **ippmLComb_vv_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride2*, Ipp32u *len*);

IppStatus **ippmLComb_vv_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*);

IppStatus **ippmLComb_vv_64f_3x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*);

IppStatus **ippmLComb_vv_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmLComb_vv_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmLComb_vv_64f_4x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*);

IppStatus **ippmLComb_vv_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmLComb_vv_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmLComb_vv_64f_5x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*);

IppStatus **ippmLComb_vv_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmLComb_vv_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmLComb_vv_64f_6x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*);

IppStatus **ippmLComb_vv_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmLComb_vv_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

LComb 関数 : 32f データのベクトル - ベクトル配列の演算

IppStatus **ippmLComb_vva_32f** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_L** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_3x1** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_3x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_3x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_3x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_3x1_L** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_3x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_4x1_L** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_4x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_5x1_L** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_5x1_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_32f_6x1** (const Ipp32f* *pSrc1*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmLComb_vva_32f_6x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLComb_vva_32f_6x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u count);
IppStatus ippmLComb_vva_32f_6x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f scale2,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmLComb_vva_32f_6x1_L (const Ipp32f* pSrc1, Ipp32f scale1,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f scale2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmLComb_vva_32f_6x1_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

LComb 関数 : 64f データのベクトル - ベクトル配列の演算

```
IppStatus ippmLComb_vva_64f (const Ipp64f* pSrc1, Ipp64f scale1, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f scale2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vva_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp64f scale2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vva_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f scale2, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vva_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f scale1, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f scale2,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vva_64f_L (const Ipp64f* pSrc1, Ipp64f scale1,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f scale2, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vva_64f_LS2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, Ipp64f scale1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f scale2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
```

IppStatus **ippmLComb_vva_64f_3x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_3x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_3x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_4x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_4x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_4x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_5x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_5x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_5x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_6x1** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_6x1_L** (const Ipp64f* *pSrc1*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vva_64f_6x1_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

LComb 関数 : 32f データのベクトル配列 - ベクトルの演算

```
IppStatus ippmLComb_vav_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32f scale1, const Ipp32f* pSrc2, Ipp32f scale2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vav_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vav_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vav_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vav_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32f scale2,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vav_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vav_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32f scale2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmLComb_vav_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLComb_vav_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u count);
IppStatus ippmLComb_vav_32f_3x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLComb_vav_32f_3x1_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32f scale2,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmLComb_vav_32f_3x1_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f scale2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmLComb_vav_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_5x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_5x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_6x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_6x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_6x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_6x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_6x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_32f_6x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

LComb 関数 : 64f データのベクトル配列 - ベクトルの演算

IppStatus **ippmLComb_vav_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_3x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_3x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_3x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_3x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_6x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vav_64f_6x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

LComb 関数 : 32f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmLComb_vava_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f
    scale2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vava_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp32f scale2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vava_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vava_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f scale2,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vava_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vava_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f scale2,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vava_32f_3x1 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32f scale1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32f scale2, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmLComb_vava_32f_3x1_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp32f scale2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLComb_vava_32f_3x1_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f scale1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f scale2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u count);
IppStatus ippmLComb_vava_32f_3x1_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32f scale1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f scale2,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u
    count);
```

IppStatus **ippmLComb_vava_32f_3x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_3x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_4x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_4x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_4x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_4x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_4x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_4x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_5x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_5x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_5x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_5x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_5x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_5x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_6x1** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_6x1_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_6x1_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_6x1_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_6x1_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_32f_6x1_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f *scale1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f *scale2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

LComb 関数 : 64f データのベクトル配列 - ベクトル配列の演算

```
IppStatus ippmLComb_vava_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp64f scale1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp64f
    scale2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vava_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp64f scale1, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp64f scale2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vava_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f scale2, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vava_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f scale1, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f scale2,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vava_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f scale2, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u len, Ipp32u count);
IppStatus ippmLComb_vava_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f scale1, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f scale2,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u len,
    Ipp32u count);
IppStatus ippmLComb_vava_64f_3x1 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp64f scale1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp64f scale2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmLComb_vava_64f_3x1_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp64f scale1, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp64f scale2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLComb_vava_64f_3x1_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f scale1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f scale2, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u count);
IppStatus ippmLComb_vava_64f_3x1_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f scale1, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f scale2,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u
    count);
```

IppStatus **ippmLComb_vava_64f_3x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_3x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_4x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_4x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_4x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_4x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_4x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_4x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_5x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_5x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_5x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_5x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_5x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_5x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_6x1** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_6x1_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_6x1_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_6x1_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_6x1_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLComb_vava_64f_6x1_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f *scale1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f *scale2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul

Mul 関数は、ベクトルに定数を掛ける。この関数の説明とすべての引数のリストは、Mul を参照のこと。

Mul 関数 : 32f データのベクトル - 定数の演算

```
IppStatus ippmMul_vc_32f (const Ipp32f* pSrc, Ipp32f val, Ipp32f* pDst,
    Ipp32u len);
IppStatus ippmMul_vc_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2, Ipp32u len);
IppStatus ippmMul_vc_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmMul_vc_32f_3x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmMul_vc_32f_3x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_32f_3x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vc_32f_4x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmMul_vc_32f_4x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_32f_4x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vc_32f_5x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmMul_vc_32f_5x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_32f_5x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vc_32f_6x1 (const Ipp32f* pSrc, Ipp32f val, Ipp32f*
    pDst);
IppStatus ippmMul_vc_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s srcStride2,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データのベクトル - 定数の演算

```
IppStatus ippmMul_vc_64f (const Ipp64f* pSrc, Ipp64f val, Ipp64f* pDst,
    Ipp32u len);
IppStatus ippmMul_vc_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2, Ipp32u len);
```

```
IppStatus ippmMul_vc_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len);
IppStatus ippmMul_vc_64f_3x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmMul_vc_64f_3x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_64f_3x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vc_64f_4x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmMul_vc_64f_4x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_64f_4x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vc_64f_5x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmMul_vc_64f_5x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_64f_5x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vc_64f_6x1 (const Ipp64f* pSrc, Ipp64f val, Ipp64f*
    pDst);
IppStatus ippmMul_vc_64f_6x1_S2 (const Ipp64f* pSrc, Ipp32s srcStride2,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vc_64f_6x1_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

Mul 関数 : 32f データのベクトル配列 - 定数の演算

```
IppStatus ippmMul_vac_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u
    count);
IppStatus ippmMul_vac_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmMul_vac_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmMul_vac_32f_PS2 (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmMul_vac_32f_L (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
```

IppStatus **ippmMul_vac_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_3x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_3x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_3x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_3x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_3x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_3x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_4x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_4x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_4x1_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_4x1_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_4x1_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_4x1_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_5x1** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_32f_5x1_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmMul_vac_32f_5x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_vac_32f_5x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vac_32f_5x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_vac_32f_5x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vac_32f_6x1 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmMul_vac_32f_6x1_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vac_32f_6x1_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_vac_32f_6x1_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vac_32f_6x1_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_vac_32f_6x1_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 64f データのベクトル配列 - 定数の演算

```
IppStatus ippmMul_vac_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u len, Ipp32u
    count);
IppStatus ippmMul_vac_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
IppStatus ippmMul_vac_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u len, Ipp32u
    count);
IppStatus ippmMul_vac_64f_PS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u len, Ipp32u count);
```

IppStatus **ippmMul_vac_64f_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *len*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_3x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_3x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_3x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_3x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_3x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_3x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_4x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_4x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_4x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_4x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_4x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_4x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_5x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_5x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_5x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_5x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_5x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_5x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_6x1** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_6x1_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_6x1_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_6x1_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_6x1_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vac_64f_6x1_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

行列代数関数のプロトタイプ

C

Transpose

Transpose 関数は、行列の転置を実行する。この関数の説明とすべての引数のリストは、Transpose を参照のこと。

Transpose 関数 : 32f データの行列の演算

```
IppStatus ippmTranspose_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32u width, Ipp32u height, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u width, Ipp32u height, Ipp32f*
    pDst, Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmTranspose_m_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmTranspose_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmTranspose_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmTranspose_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmTranspose_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmTranspose_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmTranspose_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
```

```
IppStatus ippmTranspose_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmTranspose_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
```

Transpose 関数 : 32f データの行列配列の演算

```
IppStatus ippmTranspose_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u width, Ipp32u height, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmTranspose_ma_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmTranspose_ma_32f_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u width, Ipp32u height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmTranspose_ma_32f_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u width, Ipp32u height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u
    count);
IppStatus ippmTranspose_ma_32f_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmTranspose_ma_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmTranspose_ma_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmTranspose_ma_32f_3x3_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmTranspose_ma_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_5x5_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_5x5_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_5x5_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_5x5_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_6x6** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_6x6_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_6x6_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_6x6_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_6x6_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_32f_6x6_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Transpose 関数 : 64f データの行列の演算

IppStatus **ippmTranspose_m_64f** (const Ipp64f* *pSrc*, Ipp32s *srcStride1*, Ipp32u *width*, Ipp32u *height*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmTranspose_m_64f_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u *width*, Ipp32u *height*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmTranspose_m_64f_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmTranspose_m_64f_3x3** (const Ipp64f* *pSrc*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmTranspose_m_64f_3x3_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmTranspose_m_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmTranspose_m_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmTranspose_m_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmTranspose_m_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

```
IppStatus ippmTranspose_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmTranspose_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmTranspose_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmTranspose_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmTranspose_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
```

Transpose 関数 : 64f データの行列配列の演算

```
IppStatus ippmTranspose_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u width, Ipp32u height, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmTranspose_ma_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmTranspose_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u width, Ipp32u height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmTranspose_ma_64f_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u width, Ipp32u height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u
    count);
IppStatus ippmTranspose_ma_64f_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride1, Ipp32s dstStride2, Ipp32u count,);
IppStatus ippmTranspose_ma_64f_3x3 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
```

IppStatus **ippmTranspose_ma_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmTranspose_ma_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmTranspose_ma_64f_5x5_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_64f_5x5_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmTranspose_ma_64f_6x6 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmTranspose_ma_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmTranspose_ma_64f_6x6_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmTranspose_ma_64f_6x6_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmTranspose_ma_64f_6x6_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
```

Invert

Invert 関数は、行列の逆行列を計算する。この関数の説明とすべての引数のリストは、Invert を参照のこと。

Invert 関数 : 32f データの行列の演算

```
IppStatus ippmInvert_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmInvert_m_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f*
    pDst, Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmInvert_m_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmInvert_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst, Ipp32s dstStride1);
```

```
IppStatus ippmInvert_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmInvert_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmInvert_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmInvert_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmInvert_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
```

Invert 関数 : 64f データの行列の演算

```
IppStatus ippmInvert_m_64f (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmInvert_m_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f*
    pDst, Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmInvert_m_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f** pDst, Ipp32s
    dstROIShift);
```

```
IppStatus ippmInvert_m_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmInvert_m_64f_4x4 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmInvert_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmInvert_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmInvert_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
```

Invert 関数 : 32f データの行列配列の演算

```
IppStatus ippmInvert_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f*
    pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmInvert_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f*
    pBuffer, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmInvert_ma_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmInvert_ma_32f_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f*
    pBuffer, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmInvert_ma_32f_L (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmInvert_ma_32f_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u
    widthHeight, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmInvert_ma_32f_3x3** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_3x3_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_3x3_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_3x3_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmInvert_ma_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmInvert_ma_32f_5x5_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmInvert_ma_32f_5x5_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmInvert_ma_32f_5x5_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmInvert_ma_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s
    dstStride1, Ipp32u count);
IppStatus ippmInvert_ma_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmInvert_ma_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmInvert_ma_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmInvert_ma_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmInvert_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
```

Invert 関数 : 64f データの行列配列の演算

```
IppStatus ippmInvert_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f*
    pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmInvert_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f*
    pBuffer, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmInvert_ma_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
```

IppStatus **ippmInvert_ma_64f_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_3x3** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_3x3_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_5x5_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_5x5_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_6x6** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_6x6_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_6x6_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_6x6_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*,);

IppStatus **ippmInvert_ma_64f_6x6_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmInvert_ma_64f_6x6_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

FrobNorm

FrobNorm 関数は、行列の Frobenius ノルムを計算する。この関数の説明とすべての引数のリストは、FrobNorm を参照のこと。

FrobNorm 関数 : 32f データの行列の演算

```
IppStatus ippmFrobNorm_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32u width, Ipp32u height, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u width, Ipp32u height, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmFrobNorm_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
```

FrobNorm 関数 : 64f データの行列の演算

```
IppStatus ippmFrobNorm_m_64f (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32u width, Ipp32u height, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u width, Ipp32u height, Ipp64f* pDst);
```

```
IppStatus ippmFrobNorm_m_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_3x3 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_4x4 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmFrobNorm_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
```

FrobNorm 関数 : 32f データの行列配列の演算

```
IppStatus ippmFrobNorm_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u width, Ipp32u height, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp32f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp32f* pDst, Ipp32u
    count);
IppStatus ippmFrobNorm_ma_32f_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u width, Ipp32u height,
    Ipp32f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u width, Ipp32u height,
    Ipp32f* pDst, Ipp32u count);
```

IppStatus **ippmFrobNorm_ma_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_3x3** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_3x3_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_3x3_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_3x3_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_5x5_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_5x5_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmFrobNorm_ma_32f_5x5_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

```
IppStatus ippmFrobNorm_ma_32f_5x5_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
```

FrobNorm 関数 : 64f データの行列配列の演算

```
IppStatus ippmFrobNorm_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u width, Ipp32u height, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u width, Ipp32u height, Ipp64f* pDst, Ipp32u
    count);
IppStatus ippmFrobNorm_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u width, Ipp32u height,
    Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u width, Ipp32u height,
    Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u width,
    Ipp32u height, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_3x3 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
```

```
IppStatus ippmFrobNorm_ma_64f_3x3_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_3x3_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_3x3_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_4x4 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_4x4_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_4x4_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_4x4_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_5x5 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_5x5_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_5x5_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_5x5_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_6x6 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
```

```
IppStatus ippmFrobNorm_ma_64f_6x6_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_6x6_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmFrobNorm_ma_64f_6x6_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
```

Det

Det 関数は、行列の行列式を計算する。この関数の説明とすべての引数のリストは、Det を参照のこと。

Det 関数 : 32f データの行列の演算

```
IppStatus ippmDet_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1, Ipp32u
    widthHeight, Ipp32f* pBuffer, Ipp32f* pDst);
IppStatus ippmDet_m_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f*
    pDst);
IppStatus ippmDet_m_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f* pDst);
IppStatus ippmDet_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
IppStatus ippmDet_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmDet_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst);
IppStatus ippmDet_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
IppStatus ippmDet_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmDet_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst);
IppStatus ippmDet_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
IppStatus ippmDet_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmDet_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst);
IppStatus ippmDet_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
```

```
IppStatus ippmDet_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmDet_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f* pDst);
```

Det 関数 : 64f データの行列の演算

```
IppStatus ippmDet_m_64f (const Ipp64f* pSrc, Ipp32s srcStride1, Ipp32u
    widthHeight, Ipp32f* pBuffer, Ipp64f* pDst);
IppStatus ippmDet_m_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f*
    pDst);
IppStatus ippmDet_m_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f* pDst);
IppStatus ippmDet_m_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmDet_m_64f_3x3_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_4x4 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmDet_m_64f_4x4_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmDet_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmDet_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmDet_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f* pDst);
```

Det 関数 : 32f データの行列配列の演算

```
IppStatus ippmDet_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0, Ipp32s
    srcStride1, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32u count);
```

IppStatus **ippmDet_ma_32f_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_3x3** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_3x3_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_3x3_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_3x3_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmDet_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

```
IppStatus ippmDet_ma_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_5x5_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_5x5_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_5x5_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmDet_ma_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmDet_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
```

Det 関数 : 64f データの行列配列の演算

```
IppStatus ippmDet_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0, Ipp32s
    srcStride1, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmDet_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f*
    pBuffer, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f*
    pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_L (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp32f* pBuffer, Ipp64f*
    pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_LS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f*
    pBuffer, Ipp64f* pDst, Ipp32u count);
```

```
IppStatus ippmDet_ma_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_3x3_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_3x3_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_3x3_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmDet_ma_64f_4x4 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_4x4_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_4x4_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_4x4_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmDet_ma_64f_5x5 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_5x5_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_5x5_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_5x5_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmDet_ma_64f_6x6 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
```

```
IppStatus ippmDet_ma_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_6x6_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_6x6_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmDet_ma_64f_6x6_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
```

Trace

Trace 関数は、行列のトレースを計算する。この関数の説明とすべての引数のリストは、Trace を参照のこと。

Trace 関数 : 32f データの行列の演算

```
IppStatus ippmTrace_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32u widthHeight, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
IppStatus ippmTrace_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
IppStatus ippmTrace_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
IppStatus ippmTrace_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pDst);
```

```
IppStatus ippmTrace_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pDst);
IppStatus ippmTrace_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst);
```

Trace 関数 : 64f データの行列の演算

```
IppStatus ippmTrace_m_64f (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32u widthHeight, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32u widthHeight, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmTrace_m_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_4x4 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmTrace_m_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmTrace_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pDst);
IppStatus ippmTrace_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pDst);
IppStatus ippmTrace_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst);
```

Trace 関数 : 32f データの行列配列の演算

```
IppStatus ippmTrace_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight, Ipp32f*
    pDst, Ipp32u count);
```

IppStatus **ippmTrace_ma_32f_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u *widthHeight*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u *widthHeight*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u *widthHeight*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_3x3** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_3x3_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_3x3_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_3x3_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pDst*, Ipp32u *count*);

```
IppStatus ippmTrace_ma_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_5x5_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_5x5_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_5x5_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmTrace_ma_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
IppStatus ippmTrace_ma_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f* pDst,
    Ipp32u count);
```

Trace 関数 : 64f データの行列配列の演算

```
IppStatus ippmTrace_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u widthHeight, Ipp64f*
    pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32u widthHeight, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u widthHeight, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmTrace_ma_64f_L (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32u widthHeight, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u
    widthHeight, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
```

IppStatus **ippmTrace_ma_64f_3x3_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_5x5_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_5x5_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pDst*, Ipp32u *count*);

IppStatus **ippmTrace_ma_64f_6x6** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pDst*, Ipp32u *count*);

```
IppStatus ippmTrace_ma_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
IppStatus ippmTrace_ma_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_6x6_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_6x6_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pDst, Ipp32u count);
IppStatus ippmTrace_ma_64f_6x6_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f* pDst,
    Ipp32u count);
```

Mul

Mul 関数は、行列に定数、ベクトル、または他の行列を掛ける。この関数の説明とすべての引数のリストは、Mul を参照のこと。

Mul 関数 : 32f データの行列 - 定数の演算

```
IppStatus ippmMul_mc_32f (const Ipp32f* pSrc, Ipp32s srcStride1, Ipp32f
    val, Ipp32f* pDst, Ipp32s dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmMul_mc_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height);
IppStatus ippmMul_mc_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u
    height);
IppStatus ippmMul_mc_32f_3x3 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mc_32f_4x4 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mc_32f_5x5 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
```

```
IppStatus ippmMul_mc_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mc_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データの行列 - 定数の演算

```
IppStatus ippmMul_mc_64f (const Ipp64f* pSrc, Ipp32s srcStride1, Ipp64f
    val, Ipp64f* pDst, Ipp32s dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmMul_mc_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height);
IppStatus ippmMul_mc_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u
    height);
IppStatus ippmMul_mc_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mc_64f_4x4 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mc_64f_5x5 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

```
IppStatus ippmMul_mc_64f_6x6 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mc_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mc_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

Mul 関数 : 32f データの転置行列 - 定数の演算

```
IppStatus ippmMul_mTc_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1, Ipp32u width, Ipp32u
    height);
IppStatus ippmMul_mTc_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height);
IppStatus ippmMul_mTc_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u
    height);
IppStatus ippmMul_mTc_32f_3x3 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTc_32f_4x4 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTc_32f_5x5 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTc_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f val, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
```

```
IppStatus ippmMul_mTc_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データの転置行列 - 定数の演算

```
IppStatus ippmMul_mTc_64f (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1, Ipp32u width, Ipp32u
    height);
IppStatus ippmMul_mTc_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height);
IppStatus ippmMul_mTc_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u
    height);
IppStatus ippmMul_mTc_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTc_64f_4x4 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTc_64f_5x5 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTc_64f_6x6 (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f val, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmMul_mTc_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32s dstStride2);
IppStatus ippmMul_mTc_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift);
```

Mul 関数 : 32f データの行列配列 - 定数の演算

```
IppStatus ippmMul_mac_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_32f_S2 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_32f_P (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u
    height, Ipp32u count);
IppStatus ippmMul_mac_32f_PS2 (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_32f_L (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_32f_LS2 (const Ipp32f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_32f_3x3 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mac_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mac_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_mac_32f_3x3_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mac_32f_3x3_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mac_32f_3x3_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mac_32f_4x4 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
```

IppStatus **ippmMul_mac_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_5x5_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_5x5_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_5x5_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_5x5_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_6x6** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_32f_6x6_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmMul_mac_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_mac_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mac_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mac_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
```

Mul 関数 : 64f データの行列配列 - 定数の演算

```
IppStatus ippmMul_mac_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_64f_S2 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f val, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_64f_P (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u
    height, Ipp32u count);
IppStatus ippmMul_mac_64f_PS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_64f_L (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_64f_LS2 (const Ipp64f** pSrc, Ipp32s srcROIShift,
    Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f val, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmMul_mac_64f_3x3 (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mac_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f val,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mac_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f val, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
```

IppStatus **ippmMul_mac_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_5x5_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_5x5_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_6x6** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_6x6_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_6x6_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_6x6_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_6x6_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mac_64f_6x6_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - 定数の演算

IppStatus **ippmMul_maTc_32f** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_3x3** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_3x3_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_3x3_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_3x3_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f *val*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmMul_maTc_32f_5x5 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTc_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTc_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_maTc_32f_5x5_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTc_32f_5x5_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTc_32f_5x5_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTc_32f_6x6 (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f val, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTc_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTc_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f val, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    count);
IppStatus ippmMul_maTc_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTc_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f val, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTc_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f val,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
```

Mul 関数 : 64f データの転置行列配列 - 定数の演算

```
IppStatus ippmMul_maTc_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f val, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u count);
```

IppStatus **ippmMul_maTc_64f_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_3x3** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_3x3_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_5x5_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_5x5_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_6x6** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_6x6_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_6x6_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_6x6_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_6x6_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTc_64f_6x6_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f *val*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列 - ベクトルの演算

IppStatus **ippmMul_mv_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32u *vecLen*, Ipp32f* *pDst*);

IppStatus **ippmMul_mv_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mv_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mv_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*);

IppStatus **ippmMul_mv_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mv_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mv_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*);

IppStatus **ippmMul_mv_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mv_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mv_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*);

IppStatus **ippmMul_mv_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

```
IppStatus ippmMul_mv_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mv_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmMul_mv_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_mv_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データの行列 - ベクトルの演算

```
IppStatus ippmMul_mv_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u matrWidth, Ipp32u matrHeight, const Ipp64f* pSrc2, Ipp32u
    vecLen, Ipp64f* pDst);
IppStatus ippmMul_mv_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u matrWidth, Ipp32u matrHeight, const
    Ipp64f* pSrc2, Ipp32s src2Stride2, Ipp32u vecLen, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmMul_mv_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u matrWidth, Ipp32u matrHeight, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u vecLen, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mv_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmMul_mv_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_mv_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mv_64f_4x4 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmMul_mv_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_mv_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mv_64f_5x5 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmMul_mv_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
```

```
IppStatus ippmMul_mv_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mv_64f_6x6 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmMul_mv_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_mv_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 32f データの転置行列 - ベクトルの演算

```
IppStatus ippmMul_mTv_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u matrWidth, Ipp32u matrHeight, const Ipp32f* pSrc2, Ipp32u
    vecLen, Ipp32f* pDst);
IppStatus ippmMul_mTv_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u matrWidth, Ipp32u matrHeight, const
    Ipp32f* pSrc2, Ipp32s src2Stride2, Ipp32u vecLen, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmMul_mTv_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u matrWidth, Ipp32u matrHeight, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u vecLen, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTv_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmMul_mTv_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_mTv_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTv_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmMul_mTv_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_mTv_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTv_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmMul_mTv_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
```

IppStatus **ippmMul_mTv_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTv_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*);

IppStatus **ippmMul_mTv_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTv_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 64f データの転置行列 - ベクトルの演算

IppStatus **ippmMul_mTv_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32u *vecLen*, Ipp64f* *pDst*);

IppStatus **ippmMul_mTv_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTv_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTv_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*);

IppStatus **ippmMul_mTv_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTv_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTv_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*);

IppStatus **ippmMul_mTv_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTv_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTv_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*);

IppStatus **ippmMul_mTv_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTv_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTv_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*);

IppStatus **ippmMul_mTv_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTv_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 32f データの行列 - ベクトル配列の演算

IppStatus **ippmMul_mva_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmMul_mva_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mva_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mva_32f_5x5_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mva_32f_5x5_L (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mva_32f_5x5_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mva_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32u count);
IppStatus ippmMul_mva_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mva_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mva_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift,  const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmMul_mva_32f_6x6_L (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mva_32f_6x6_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 64f データの行列 - ベクトル配列の演算

```
IppStatus ippmMul_mva_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u matrWidth, Ipp32u matrHeight, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32u vecLen, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    count);
```

IppStatus **ippmMul_mva_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mva_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列 - ベクトル配列の演算

IppStatus **ippmMul_mTva_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列 - ベクトル配列の演算

IppStatus **ippmMul_mTva_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTva_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列配列 - ベクトルの演算

IppStatus **ippmMul_mav_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列配列 - ベクトルの演算

IppStatus **ippmMul_mav_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mav_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - ベクトルの演算

IppStatus **ippmMul_maTv_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列配列 - ベクトルの演算

IppStatus **ippmMul_maTv_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTv_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列配列 - ベクトル配列の演算

IppStatus **ippmMul_mava_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列配列 - ベクトル配列の演算

IppStatus **ippmMul_mava_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mava_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - ベクトル配列の演算

IppStatus **ippmMul_maTva_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmMul_maTva_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTva_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTva_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTva_32f_6x6_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTva_32f_6x6_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 64f データの転置行列配列 - ベクトル配列の演算

```
IppStatus ippmMul_maTva_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u matrWidth, Ipp32u matrHeight, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32u vecLen, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmMul_maTva_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    matrWidth, Ipp32u matrHeight, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, Ipp32u vecLen, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTva_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u matrWidth, Ipp32u matrHeight, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u vecLen, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmMul_maTva_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u matrWidth, Ipp32u
    matrHeight, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u vecLen, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTva_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u matrWidth, Ipp32u
    matrHeight, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32u
    vecLen, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTva_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *vecLen*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTva_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データのベクトル - 行列の演算

IppStatus **ippmMul_vm_32f** (const Ipp32f* *pSrc1*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*);

IppStatus **ippmMul_vm_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vm_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vm_32f_3x3** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*);

IppStatus **ippmMul_vm_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vm_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vm_32f_4x4** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*);

IppStatus **ippmMul_vm_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vm_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vm_32f_5x5** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*);

IppStatus **ippmMul_vm_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vm_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vm_32f_6x6** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*);

IppStatus **ippmMul_vm_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

```
IppStatus ippmMul_vm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データのベクトル - 行列の演算

```
IppStatus ippmMul_vm_64f (const Ipp64f* pSrc1, Ipp32u vecLen, const
    Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32u matrWidth, Ipp32u
    matrHeight, Ipp64f* pDst);
IppStatus ippmMul_vm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride2,
    Ipp32u vecLen, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u matrWidth, Ipp32u matrHeight, Ipp64f* pDst,
    Ipp32s dstStride2);
IppStatus ippmMul_vm_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u vecLen, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32u
    matrWidth, Ipp32u matrHeight, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vm_64f_3x3 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp32s src2Stride1, Ipp64f* pDst);
IppStatus ippmMul_vm_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vm_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vm_64f_4x4 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp32s src2Stride1, Ipp64f* pDst);
IppStatus ippmMul_vm_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vm_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vm_64f_5x5 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp32s src2Stride1, Ipp64f* pDst);
IppStatus ippmMul_vm_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vm_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vm_64f_6x6 (const Ipp64f* pSrc1, const Ipp64f* pSrc2,
    Ipp32s src2Stride1, Ipp64f* pDst);
IppStatus ippmMul_vm_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
```

```
IppStatus ippmMul_vm_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 32f データのベクトル - 転置行列の演算

```
IppStatus ippmMul_vmT_32f (const Ipp32f* pSrc1, Ipp32u vecLen, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32u matrWidth, Ipp32u
    matrHeight, Ipp32f* pDst);
IppStatus ippmMul_vmT_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride2,
    Ipp32u vecLen, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u matrWidth, Ipp32u matrHeight, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmMul_vmT_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u vecLen, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32u
    matrWidth, Ipp32u matrHeight, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vmT_32f_3x3 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32f* pDst);
IppStatus ippmMul_vmT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vmT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vmT_32f_4x4 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32f* pDst);
IppStatus ippmMul_vmT_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vmT_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vmT_32f_5x5 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32f* pDst);
IppStatus ippmMul_vmT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmMul_vmT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_vmT_32f_6x6 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32f* pDst);
IppStatus ippmMul_vmT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
```

IppStatus **ippmMul_vmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 64f データのベクトル - 転置行列の演算

IppStatus **ippmMul_vmT_64f** (const Ipp64f* *pSrc1*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*);

IppStatus **ippmMul_vmT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vmT_64f_3x3** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*);

IppStatus **ippmMul_vmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vmT_64f_4x4** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*);

IppStatus **ippmMul_vmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vmT_64f_5x5** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*);

IppStatus **ippmMul_vmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_vmT_64f_6x6** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*);

IppStatus **ippmMul_vmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmMul_vmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 32f データのベクトル配列 - 行列の演算

IppStatus **ippmMul_vam_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, IIpp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データのベクトル配列 - 行列の演算

IppStatus **ippmMul_vam_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vam_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データのベクトル配列 - 転置行列の演算

IppStatus **ippmMul_vamT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データのベクトル配列 - 転置行列の演算

IppStatus **ippmMul_vamT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データのベクトル - 行列配列の演算

IppStatus **ippmMul_vma_32f** (const Ipp32f* *pSrc1*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_L** (const Ipp32f* *pSrc1*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_3x3** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_3x3_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_4x4** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_4x4_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_5x5** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_5x5_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_6x6** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_6x6_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

Mul 関数 : 64f データのベクトル - 行列配列の演算

IppStatus **ippmMul_vma_64f** (const Ipp64f* *pSrc1*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_L** (const Ipp64f* *pSrc1*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_3x3** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_3x3_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_4x4** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_4x4_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_5x5** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_5x5_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_6x6** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_6x6_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vma_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データのベクトル - 転置行列配列の演算

IppStatus **ippmMul_vmaT_32f** (const Ipp32f* *pSrc1*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_L** (const Ipp32f* *pSrc1*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_3x3** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_3x3_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_4x4** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_4x4_L** (const Ipp32f* *pSrc1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_32f_5x5** (const Ipp32f* *pSrc1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmMul_vmaT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vmaT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vmaT_32f_5x5_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmMul_vmaT_32f_5x5_L (const Ipp32f* pSrc1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vmaT_32f_5x5_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vmaT_32f_6x6 (const Ipp32f* pSrc1, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32u count);
IppStatus ippmMul_vmaT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vmaT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vmaT_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmMul_vmaT_32f_6x6_L (const Ipp32f* pSrc1, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vmaT_32f_6x6_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride2, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 64f データのベクトル - 転置行列配列の演算

```
IppStatus ippmMul_vmaT_64f (const Ipp64f* pSrc1, Ipp32u vecLen, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    matrWidth, Ipp32u matrHeight, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32u count);
```

IppStatus **ippmMul_vmaT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_L** (const Ipp64f* *pSrc1*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_3x3** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_3x3_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_4x4** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_4x4_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_5x5** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_5x5_L** (const Ipp64f* *pSrc1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vmaT_64f_6x6** (const Ipp64f* *pSrc1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmMul_vmaT_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vmaT_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vmaT_64f_6x6_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmMul_vmaT_64f_6x6_L (const Ipp64f* pSrc1, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vmaT_64f_6x6_LS2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride2, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 32f データのベクトル配列 - 行列配列の演算

```
IppStatus ippmMul_vama_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32u vecLen, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32u matrWidth, Ipp32u matrHeight, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmMul_vama_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride2, Ipp32u vecLen, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    matrWidth, Ipp32u matrHeight, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u vecLen, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u matrWidth, Ipp32u matrHeight, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vama_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u vecLen, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u matrWidth,
    Ipp32u matrHeight, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u vecLen, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u matrWidth, Ipp32u
    matrHeight, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_vama_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmMul_vama_32f_4x4_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_5x5 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmMul_vama_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vama_32f_5x5_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_5x5_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vama_32f_5x5_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32u count);
IppStatus ippmMul_vama_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vama_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vama_32f_6x6_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_vama_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データのベクトル配列 - 行列配列の演算

IppStatus **ippmMul_vama_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vama_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データのベクトル配列 - 転置行列配列の演算

IppStatus **ippmMul_vamaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vma_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

```
IppStatus ippmMul_vamaT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vamaT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vamaT_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vamaT_32f_6x6_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vamaT_32f_6x6_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 64f データのベクトル配列 - 転置行列配列の演算

```
IppStatus ippmMul_vamaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32u vecLen, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32u matrWidth, Ipp32u matrHeight, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32u count);
IppStatus ippmMul_vamaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride2, Ipp32u vecLen, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32u matrWidth, Ipp32u matrHeight, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_vamaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u vecLen, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u matrWidth, Ipp32u matrHeight, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_vamaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u vecLen, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u matrWidth,
    Ipp32u matrHeight, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_vamaT_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u vecLen, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u matrWidth, Ipp32u
    matrHeight, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_vamaT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *vecLen*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *matrWidth*, Ipp32u *matrHeight*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_vamaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmMul_vamaT_64f_6x6_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Mul 関数 : 32f データの行列 - 行列の演算

```
IppStatus ippmMul_mm_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmMul_mm_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift);
IppStatus ippmMul_mm_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mm_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mm_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
```

```
IppStatus ippmMul_mm_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mm_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データの行列 - 行列の演算

```
IppStatus ippmMul_mm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp64f* pDst,
    Ipp32s dstStride1);
IppStatus ippmMul_mm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift);
IppStatus ippmMul_mm_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mm_64f_4x4 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
```

```
IppStatus ippmMul_mm_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mm_64f_5x5 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mm_64f_6x6 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mm_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mm_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 32f データの転置行列 - 行列の演算

```
IppStatus ippmMul_mTm_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmMul_mTm_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mTm_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift);
IppStatus ippmMul_mTm_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mTm_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
```

```
IppStatus ippmMul_mTm_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTm_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mTm_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mTm_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTm_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mTm_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mTm_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mTm_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mTm_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mTm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Mul 関数 : 64f データの転置行列 - 行列の演算

```
IppStatus ippmMul_mTm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp64f* pDst,
    Ipp32s dstStride1);
IppStatus ippmMul_mTm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
```

IppStatus **ippmMul_mTm_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTm_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTm_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTm_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTm_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTm_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTm_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTm_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTm_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTm_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTm_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTm_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTm_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 32f データの行列 - 転置行列の演算

```
IppStatus ippmMul_mmT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmMul_mmT_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mmT_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift);
IppStatus ippmMul_mmT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mmT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mmT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mmT_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mmT_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mmT_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmMul_mmT_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmMul_mmT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmMul_mmT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

IppStatus **ippmMul_mmT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mmT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 64f データの行列 - 転置行列の演算

IppStatus **ippmMul_mmT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mmT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mmT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 32f データの転置行列 - 転置行列の演算

IppStatus **ippmMul_mTmT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 64f データの転置行列 - 転置行列の演算

IppStatus **ippmMul_mTmT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmMul_mTmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmMul_mTmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmMul_mTmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Mul 関数 : 32f データの行列 - 行列配列の演算

```
IppStatus ippmMul_mma_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mma_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mma_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mma_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mma_32f_L (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mma_32f_LS2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mma_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mma_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mma_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列 - 行列配列の演算

```
IppStatus ippmMul_mma_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mma_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mma_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mma_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mma_64f_L (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mma_64f_LS2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mma_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mma_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mma_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mma_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列 - 行列配列の演算

```
IppStatus ippmMul_mTma_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTma_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mTma_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mTma_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mTma_32f_L (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTma_32f_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mTma_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTma_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mTma_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mTma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列 - 行列配列の演算

```
IppStatus ippmMul_mTma_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTma_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mTma_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mTma_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mTma_64f_L (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTma_64f_LS2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mTma_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTma_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mTma_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mTma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTma_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列 - 転置行列配列の演算

```
IppStatus ippmMul_mmaT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mmaT_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mmaT_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mmaT_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mmaT_32f_L (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mmaT_32f_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mmaT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mmaT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mmaT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列 - 転置行列配列の演算

```
IppStatus ippmMul_mmaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mmaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mmaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mmaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mmaT_64f_L (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mmaT_64f_LS2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mmaT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mmaT_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mmaT_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mmaT_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列 - 転置行列配列の演算

```
IppStatus ippmMul_mTmaT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTmaT_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mTmaT_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mTmaT_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mTmaT_32f_L (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTmaT_32f_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mTmaT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTmaT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mTmaT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mTmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列 - 転置行列配列の演算

```
IppStatus ippmMul_mTmaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTmaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mTmaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mTmaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp32u src2Width,
    Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mTmaT_64f_L (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTmaT_64f_LS2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mTmaT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mTmaT_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mTmaT_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mTmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mTmaT_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列配列 - 行列の演算

```
IppStatus ippmMul_mam_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mam_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmMul_mam_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32u src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mam_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mam_32f_L (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_mam_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mam_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mam_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mam_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mam_32f_3x3_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
```

IppStatus **ippmMul_mam_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列配列 - 行列の演算

IppStatus **ippmMul_mam_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mam_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - 行列の演算

IppStatus **ippmMul_maTm_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列配列 - 行列の演算

IppStatus **ippmMul_maTm_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTm_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列配列 - 転置行列の演算

IppStatus **ippmMul_mamT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列配列 - 転置行列の演算

IppStatus **ippmMul_mamT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u *src1Width*, Ipp32u *src1Height*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32u *src2Width*, Ipp32u *src2Height*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - 転置行列の演算

```
IppStatus ippmMul_maTmT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_maTmT_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTmT_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmT_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_maTmT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTmT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列配列 - 転置行列の演算

```
IppStatus ippmMul_maTmT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_maTmT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTmT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmT_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmMul_maTmT_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmT_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTmT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列配列 - 行列配列の演算

```
IppStatus ippmMul_mama_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mama_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mama_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mama_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mama_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mama_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mama_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列配列 - 行列配列の演算

```
IppStatus ippmMul_mama_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mama_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32u src2Width, Ipp32u
    src2Height, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mama_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mama_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mama_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mama_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mama_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mama_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mama_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - 行列配列の演算

```
IppStatus ippmMul_maTma_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTma_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTma_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTma_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTma_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列配列 - 行列配列の演算

```
IppStatus ippmMul_maTma_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTma_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTma_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTma_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTma_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTma_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTma_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの行列配列 - 転置行列配列の演算

```
IppStatus ippmMul_mamaT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mamaT_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mamaT_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mamaT_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mamaT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mamaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの行列配列 - 転置行列配列の演算

```
IppStatus ippmMul_mamaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mamaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_mamaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mamaT_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_mamaT_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_mamaT_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_mamaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_mamaT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 32f データの転置行列配列 - 転置行列配列の演算

```
IppStatus ippmMul_maTmaT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp32f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Mul 関数 : 64f データの転置行列配列 - 転置行列配列の演算

```
IppStatus ippmMul_maTmaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u src1Height, const
    Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u src1Width, Ipp32u src1Height, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32u src2Width, Ipp32u src2Height,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u src1Width, Ipp32u
    src1Height, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32u src2Width, Ipp32u src2Height, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u
    src1Width, Ipp32u src1Height, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32u
    src2Width, Ipp32u src2Height, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmMul_maTmaT_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
```

IppStatus **ippmMul_maTmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmMul_maTmaT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Gaxpy

Gaxpy 関数は、行列の GAXPY 演算を実行する。この関数の説明とすべての引数のリストは、Gaxpy を参照のこと。

Gaxpy 関数 : 32f データのベクトルの演算

```
IppStatus ippmGaxpy_v_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, const Ipp32f* pSrc3, Ipp32f* pDst, Ipp32u
    width, Ipp32u height);
IppStatus ippmGaxpy_v_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride2, const
    Ipp32f* pSrc3, Ipp32s src3Stride2, Ipp32f* pDst, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmGaxpy_v_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, const Ipp32f** pSrc3,
    Ipp32s src3ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    width, Ipp32u height);
IppStatus ippmGaxpy_v_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, const Ipp32f* pSrc3, Ipp32f* pDst);
IppStatus ippmGaxpy_v_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, const Ipp32f* pSrc3, Ipp32s src3Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmGaxpy_v_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, const
    Ipp32f** pSrc3, Ipp32s src3ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmGaxpy_v_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, const Ipp32f* pSrc3, Ipp32f* pDst);
IppStatus ippmGaxpy_v_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, const Ipp32f* pSrc3, Ipp32s src3Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
IppStatus ippmGaxpy_v_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, const
    Ipp32f** pSrc3, Ipp32s src3ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmGaxpy_v_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, const Ipp32f* pSrc3, Ipp32f* pDst);
IppStatus ippmGaxpy_v_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride2, const Ipp32f* pSrc3, Ipp32s src3Stride2, Ipp32f* pDst,
    Ipp32s dstStride2);
```

IppStatus **ippmGaxpy_v_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmGaxpy_v_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, const Ipp32f* *pSrc3*, Ipp32f* *pDst*);

IppStatus **ippmGaxpy_v_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride2*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmGaxpy_v_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

Gaxpy 関数 : 64f データのベクトルの演算

IppStatus **ippmGaxpy_v_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, const Ipp64f* *pSrc3*, Ipp64f* *pDst*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmGaxpy_v_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmGaxpy_v_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmGaxpy_v_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, const Ipp64f* *pSrc3*, Ipp64f* *pDst*);

IppStatus **ippmGaxpy_v_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmGaxpy_v_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmGaxpy_v_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, const Ipp64f* *pSrc3*, Ipp64f* *pDst*);

IppStatus **ippmGaxpy_v_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmGaxpy_v_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmGaxpy_v_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, const Ipp64f* *pSrc3*, Ipp64f* *pDst*);

IppStatus **ippmGaxpy_v_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmGaxpy_v_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmGaxpy_v_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, const Ipp64f* *pSrc3*, Ipp64f* *pDst*);

IppStatus **ippmGaxpy_v_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmGaxpy_v_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Gaxpy 関数 : 32f データのベクトル配列の演算

IppStatus **ippmGaxpy_va_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp32f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp32f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

Gaxpy 関数 : 64f データのベクトル配列の演算

IppStatus **ippmGaxpy_va_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_L** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, const Ipp64f* *pSrc3*, Ipp32s *src3Stride0*, Ipp32s *src3Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmGaxpy_va_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, const Ipp64f** *pSrc3*, Ipp32s *src3ROIShift*, Ipp32s *src3Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmGaxpy_va_64f_6x6 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, const Ipp64f*
    pSrc3, Ipp32s src3Stride0, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmGaxpy_va_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride2, const Ipp64f* pSrc3, Ipp32s
    src3Stride0, Ipp32s src3Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmGaxpy_va_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, const
    Ipp64f** pSrc3, Ipp32s src3ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmGaxpy_va_64f_6x6_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, const Ipp64f** pSrc3, Ipp32s src3ROIShift, Ipp32s
    src3Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmGaxpy_va_64f_6x6_L (const Ipp64f* pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, const Ipp64f** pSrc3, Ipp32s src3ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmGaxpy_va_64f_6x6_LS2 (const Ipp64f* pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, const
    Ipp64f** pSrc3, Ipp32s src3ROIShift, Ipp32s src3Stride2, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

Add

Add 関数は、1 つの行列を他の行列に加算する。この関数の説明とすべての引数のリストは、Add を参照のこと。

Add 関数 : 32f データの行列 - 行列の演算

```
IppStatus ippmAdd_mm_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mm_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mm_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
```

```
IppStatus ippmAdd_mm_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mm_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mm_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mm_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mm_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mm_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mm_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mm_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mm_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mm_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mm_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Add 関数 : 64f データの行列 - 行列の演算

```
IppStatus ippmAdd_mm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
```

IppStatus **ippmAdd_mm_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mm_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mm_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mm_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mm_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mm_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mm_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mm_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mm_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mm_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mm_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mm_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mm_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mm_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Add 関数 : 32f データの転置行列 - 行列の演算

IppStatus **ippmAdd_mTm_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTm_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTm_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTm_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTm_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTm_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mTm_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTm_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTm_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mTm_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTm_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTm_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

```
IppStatus ippmAdd_mTm_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mTm_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Add 関数 : 64f データの転置行列 - 行列の演算

```
IppStatus ippmAdd_mTm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mTm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mTm_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mTm_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mTm_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTm_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mTm_64f_4x4 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mTm_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTm_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mTm_64f_5x5 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
```

IppStatus **ippmAdd_mTm_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTm_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mTm_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTm_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTm_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Add 関数 : 32f データの行列 - 転置行列の演算

IppStatus **ippmAdd_mmT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mmT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mmT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mmT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mmT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mmT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mmT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mmT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

```
IppStatus ippmAdd_mmT_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mmT_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mmT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mmT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mmT_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mmT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mmT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Add 関数 : 64f データの行列 - 転置行列の演算

```
IppStatus ippmAdd_mmT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mmT_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mmT_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mmT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmAdd_mmT_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mmT_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

IppStatus **ippmAdd_mmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Add 関数 : 32f データの転置行列 - 転置行列の演算

IppStatus **ippmAdd_mTmT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTmT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTmT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTmT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

```
IppStatus ippmAdd_mTmT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTmT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mTmT_32f_4x4 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmAdd_mTmT_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTmT_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mTmT_32f_5x5 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmAdd_mTmT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTmT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmAdd_mTmT_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmAdd_mTmT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmAdd_mTmT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Add 関数 : 64f データの転置行列 - 転置行列の演算

```
IppStatus ippmAdd_mTmT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmAdd_mTmT_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
```

IppStatus **ippmAdd_mTmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmAdd_mTmT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mTmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mTmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmAdd_mTmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmAdd_mTmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmAdd_mTmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

Add 関数 : 32f データの行列配列 - 行列の演算

```
IppStatus ippmAdd_mam_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mam_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmAdd_mam_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mam_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mam_32f_L (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mam_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mam_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmAdd_mam_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_mam_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_mam_32f_3x3_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmAdd_mam_32f_3x3_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
```

IppStatus **ippmAdd_mam_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの行列配列 - 行列の演算

IppStatus **ippmAdd_mam_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mam_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの転置行列配列 - 行列の演算

IppStatus **ippmAdd_maTm_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの転置行列配列 - 行列の演算

```
IppStatus ippmAdd_maTm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmAdd_maTm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmAdd_maTm_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmAdd_maTm_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_maTm_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_maTm_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_maTm_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmAdd_maTm_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_maTm_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_maTm_64f_3x3_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmAdd_maTm_64f_3x3_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
```

IppStatus **ippmAdd_maTm_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTm_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの行列配列 - 転置行列の演算

IppStatus **ippmAdd_mamT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの行列配列 - 転置行列の演算

IppStatus **ippmAdd_mamT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの転置行列配列 - 転置行列の演算

IppStatus **ippmAdd_maTmT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの転置行列配列 - 転置行列の演算

IppStatus **ippmAdd_maTmT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの行列配列 - 行列配列の演算

```
IppStatus ippmAdd_mama_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mama_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mama_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmAdd_mama_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmAdd_mama_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmAdd_mama_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmAdd_mama_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmAdd_mama_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmAdd_mama_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_mama_32f_3x3_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmAdd_mama_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmAdd_mama_32f_5x5_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_mama_32f_5x5_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmAdd_mama_32f_5x5_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_mama_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmAdd_mama_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmAdd_mama_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmAdd_mama_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmAdd_mama_32f_6x6_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmAdd_mama_32f_6x6_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
```

Add 関数 : 64f データの行列配列 - 行列配列の演算

```
IppStatus ippmAdd_mama_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
```

IppStatus **ippmAdd_mama_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mama_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの転置行列配列 - 行列配列の演算

IppStatus **ippmAdd_maTma_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの転置行列配列 - 行列配列の演算

IppStatus **ippmAdd_maTma_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTma_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの行列配列 - 転置行列配列の演算

IppStatus **ippmAdd_mamaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの行列配列 - 転置行列配列の演算

IppStatus **ippmAdd_mamaT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_mamaT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 32f データの転置行列配列 - 転置行列配列の演算

IppStatus **ippmAdd_maTmaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Add 関数 : 64f データの転置行列配列 - 転置行列配列の演算

IppStatus **ippmAdd_maTmaT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmAdd_maTmaT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub

Sub 関数は、1 つの行列を他の行列から引く。この関数の説明とすべての引数のリストは、Sub を参照のこと。

Sub 関数 : 32f データの行列 - 行列の演算

```
IppStatus ippmSub_mm_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mm_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmSub_mm_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mm_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mm_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mm_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
```

```
IppStatus ippmSub_mm_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mm_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 64f データの行列 - 行列の演算

```
IppStatus ippmSub_mm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmSub_mm_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mm_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mm_64f_4x4 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

```
IppStatus ippmSub_mm_64f_5x5 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mm_64f_6x6 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mm_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mm_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 32f データの転置行列 - 行列の演算

```
IppStatus ippmSub_mTm_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTm_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTm_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTm_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTm_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
```

```
IppStatus ippmSub_mTm_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTm_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTm_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 64f データの転置行列 - 行列の演算

```
IppStatus ippmSub_mTm_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTm_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTm_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTm_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
```

```
IppStatus ippmSub_mTm_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTm_64f_4x4 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTm_64f_5x5 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTm_64f_6x6 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mTm_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTm_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 32f データの行列 - 転置行列の演算

```
IppStatus ippmSub_mmT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mmT_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmSub_mmT_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height);
```

```
IppStatus ippmSub_mmT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mmT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mmT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mmT_32f_4x4 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mmT_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mmT_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mmT_32f_5x5 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mmT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mmT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mmT_32f_6x6 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1);
IppStatus ippmSub_mmT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mmT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 64f データの行列 - 転置行列の演算

```
IppStatus ippmSub_mmT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
```

IppStatus **ippmSub_mmT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmSub_mmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmSub_mmT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmSub_mmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmSub_mmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmSub_mmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmSub_mmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmSub_mmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmSub_mmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

```
IppStatus ippmSub_mmT_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 32f データの転置行列 - 転置行列の演算

```
IppStatus ippmSub_mTmT_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s
    dstStride1, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTmT_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f* pDst, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTmT_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height);
IppStatus ippmSub_mTmT_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmSub_mTmT_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTmT_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTmT_32f_4x4 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmSub_mTmT_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTmT_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTmT_32f_5x5 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1, Ipp32f* pDst,
    Ipp32s dstStride1);
IppStatus ippmSub_mTmT_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTmT_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift);
```

IppStatus **ippmSub_mTmT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mTmT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmSub_mTmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*);

Sub 関数 : 64f データの転置行列 - 転置行列の演算

IppStatus **ippmSub_mTmT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmSub_mTmT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmSub_mTmT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*);

IppStatus **ippmSub_mTmT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mTmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmSub_mTmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmSub_mTmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

IppStatus **ippmSub_mTmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride1*, Ipp32s *dstStride2*);

IppStatus **ippmSub_mTmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

IppStatus **ippmSub_mTmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride1*);

```
IppStatus ippmSub_mTmT_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTmT_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
IppStatus ippmSub_mTmT_64f_6x6 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride1, Ipp64f* pDst,
    Ipp32s dstStride1);
IppStatus ippmSub_mTmT_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride1,
    Ipp32s dstStride2);
IppStatus ippmSub_mTmT_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift);
```

Sub 関数 : 32f データの行列 - 行列配列の演算

```
IppStatus ippmSub_mma_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmSub_mma_32f_P (const Ipp32f** pSrc1, Ipp32s src1ROIShift,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_32f_L (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_32f_LS2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s src2ROIShift,
    Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
```

IppStatus **ippmSub_mma_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmSub_mma_32f_6x6_L (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mma_32f_6x6_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u count);
```

Sub 関数 : 64f データの行列 - 行列配列の演算

```
IppStatus ippmSub_mma_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmSub_mma_64f_P (const Ipp64f** pSrc1, Ipp32s src1ROIShift,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_64f_L (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1,
    Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_64f_LS2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s src2ROIShift,
    Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mma_64f_3x3 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mma_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
```

IppStatus **ippmSub_mma_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mma_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの転置行列 - 行列配列の演算

```
IppStatus ippmSub_mTma_32f (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mTma_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_mTma_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mTma_32f_L (const Ipp32f* pSrc1, Ipp32s src1Stride1,
    const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1,
    Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mTma_32f_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mTma_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmSub_mTma_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mTma_32f_3x3_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
```

IppStatus **ippmSub_mTma_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_mTma_32f_5x5_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_5x5_L (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_5x5_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmSub_mTma_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mTma_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_6x6_L (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mTma_32f_6x6_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u count);
```

Sub 関数 : 64f データの転置行列 - 行列配列の演算

```
IppStatus ippmSub_mTma_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
```

IppStatus **ippmSub_mTma_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTma_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの行列 - 転置行列配列の演算

IppStatus **ippmSub_mmaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmSub_mmaT_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mmaT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
IppStatus ippmSub_mmaT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mmaT_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_mmaT_32f_6x6_L (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mmaT_32f_6x6_LS2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2,
    Ipp32u count);
```

Sub 関数 : 64f データの行列 - 転置行列配列の演算

```
IppStatus ippmSub_mmaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_mmaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride1,
    Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmSub_mmaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_mmaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
```

IppStatus **ippmSub_mmaT_64f_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mmaT_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの転置行列 - 転置行列配列の演算

IppStatus **ippmSub_mTmaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_3x3_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_3x3_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_4x4_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_4x4_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_5x5_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_5x5_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_6x6_L** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_32f_6x6_LS2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの転置行列 - 転置行列配列の演算

IppStatus **ippmSub_mTmaT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_3x3_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_3x3_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_4x4_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_4x4_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_5x5_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_5x5_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_6x6_L** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mTmaT_64f_6x6_LS2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの行列配列 - 行列の演算

IppStatus **ippmSub_mam_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの行列配列 - 行列の演算

IppStatus **ippmSub_mam_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mam_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_mam_64f_6x6_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_mam_64f_6x6_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_mam_64f_6x6_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
```

Sub 関数 : 32f データの転置行列配列 - 行列の演算

```
IppStatus ippmSub_maTm_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride1,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTm_32f_S2 (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmSub_maTm_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_maTm_32f_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTm_32f_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTm_32f_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTm_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
```

IppStatus **ippmSub_maTm_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの転置行列配列 - 行列の演算

IppStatus **ippmSub_maTm_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTm_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの行列配列 - 転置行列の演算

IppStatus **ippmSub_mamT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの行列配列 - 転置行列の演算

IppStatus **ippmSub_mamT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの転置行列配列 - 転置行列の演算

IppStatus **ippmSub_maTmT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_maTmT_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmSub_maTmT_32f_6x6_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride1, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u count);
IppStatus ippmSub_maTmT_32f_6x6_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    count);
```

Sub 関数 : 64f データの転置行列配列 - 転置行列の演算

```
IppStatus ippmSub_maTmT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride1,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTmT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTmT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_maTmT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTmT_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTmT_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_maTmT_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u count);
```

IppStatus **ippmSub_maTmT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの行列配列 - 行列配列の演算

IppStatus **ippmSub_mama_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_mama_32f_3x3_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_32f_3x3_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_32f_3x3_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_32f_4x4 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSub_mama_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mama_32f_4x4_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_32f_4x4_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_32f_4x4_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_32f_5x5 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
```

IppStatus **ippmSub_mama_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの行列配列 - 行列配列の演算

```
IppStatus ippmSub_mama_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mama_64f_S2 (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2, Ipp64f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mama_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_mama_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_mama_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_mama_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mama_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSub_mama_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mama_64f_3x3_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmSub_mama_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mama_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

```
IppStatus ippmSub_mama_64f_5x5_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_64f_5x5_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_64f_5x5_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_64f_6x6 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp64f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_64f_6x6_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSub_mama_64f_6x6_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mama_64f_6x6_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mama_64f_6x6_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mama_64f_6x6_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 32f データの転置行列配列 - 行列配列の演算

```
IppStatus ippmSub_maTma_32f (const Ipp32f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
```

IppStatus **ippmSub_maTma_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの転置行列配列 - 行列配列の演算

IppStatus **ippmSub_maTma_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTma_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの行列配列 - 転置行列配列の演算

IppStatus **ippmSub_mamaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmSub_mamaT_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, const Ipp32f* pSrc2, Ipp32s
    src2Stride0, Ipp32s src2Stride1, Ipp32f* pDst, Ipp32s dstStride0,
    Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mamaT_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp32f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u count);
IppStatus ippmSub_mamaT_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, const Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32f**
    pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmSub_mamaT_32f_6x6_PS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmSub_mamaT_32f_6x6_L (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride1, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmSub_mamaT_32f_6x6_LS2 (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp32f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
```

Sub 関数 : 64f データの行列配列 - 転置行列配列の演算

```
IppStatus ippmSub_mamaT_64f (const Ipp64f* pSrc1, Ipp32s src1Stride0,
    Ipp32s src1Stride1, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride1, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mamaT_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp32s src2Stride1, Ipp32s src2Stride2,
    Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1, Ipp32s
    dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmSub_mamaT_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32u width, Ipp32u height, Ipp32u
    count);
IppStatus ippmSub_mamaT_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u
    count);
```

IppStatus **ippmSub_mamaT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_mamaT_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 32f データの転置行列配列 - 転置行列配列の演算

IppStatus **ippmSub_maTmaT_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

Sub 関数 : 64f データの転置行列配列 - 転置行列配列の演算

IppStatus **ippmSub_maTmaT_64f** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_3x3** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_3x3_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_3x3_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_3x3_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride1*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmSub_maTmaT_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride1*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

```
IppStatus ippmSub_maTmaT_64f_6x6_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride1, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
```

連立 1 次方程式の解の関数のプロトタイプ

D

LUDecomp

この関数は、正方行列を上三角行列と下三角行列の積に分解する。この関数の説明とすべての引数のリストは、LUDecomp を参照のこと。

LUDecomp 関数 : 32f データの行列の演算

```
IppStatus ippmLUDecomp_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32u* pDstIndex, Ipp32f* pDst, Ipp32s dstStride1, Ipp32u
    widthHeight);
IppStatus ippmLUDecomp_m_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u widthHeight);
IppStatus ippmLUDecomp_m_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u widthHeight);
IppStatus ippmLUDecomp_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmLUDecomp_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmLUDecomp_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmLUDecomp_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmLUDecomp_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
```

```
IppStatus ippmLUDecomp_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmLUDecomp_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmLUDecomp_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift);
```

LUDecomp 関数 : 64f データの行列の演算

```
IppStatus ippmLUDecomp_m_64f (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp32u* pDstIndex, Ipp64f* pDst, Ipp32s dstStride1, Ipp32u
    widthHeight);
IppStatus ippmLUDecomp_m_64f_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u widthHeight);
IppStatus ippmLUDecomp_m_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u widthHeight);
IppStatus ippmLUDecomp_m_64f_3x3 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmLUDecomp_m_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp64f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmLUDecomp_m_64f_4x4 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmLUDecomp_m_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp64f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmLUDecomp_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
```

```
IppStatus ippmLUDecomp_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp64f** pDst, Ipp32s
    dstROIShift);
IppStatus ippmLUDecomp_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32u* pDstIndex, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmLUDecomp_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmLUDecomp_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp64f** pDst, Ipp32s
    dstROIShift);
```

LUDecomp 関数 : 32f データの行列配列の演算

```
IppStatus ippmLUDecomp_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u widthHeight, Ipp32u
    count);
IppStatus ippmLUDecomp_ma_32f_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u widthHeight, Ipp32u
    count);
IppStatus ippmLUDecomp_ma_32f_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
```

IppStatus **ippmLUDecomp_ma_32f_3x3_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_3x3_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_5x5_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_32f_5x5_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmLUDecomp_ma_32f_5x5_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_5x5_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp32f** pDst, Ipp32s
    dstROIShift, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmLUDecomp_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
```

LUDecomp 関数 : 64f データの行列配列の演算

```
IppStatus ippmLUDecomp_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32u* pDstIndex, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32u*
    pDstIndex, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32u* pDstIndex, Ipp64f** pDst, Ipp32s
    dstROIShift, Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUDecomp_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32u* pDstIndex, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u widthHeight, Ipp32u
    count);
```

IppStatus **ippmLUDecomp_ma_64f_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_3x3** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_3x3_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_5x5_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_5x5_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_6x6** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_6x6_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_6x6_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_6x6_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_6x6_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmLUDecomp_ma_64f_6x6_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32u* *pDstIndex*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

LUBackSubst

この関数は、LU 因数分解された正方行列を使用して連立 1 次方程式を解く。この関数の説明とすべての引数のリストは、LUBackSubst を参照のこと。

LUBackSubst 関数 : 32f データの行列 - ベクトルの演算

```
IppStatus ippmLUBackSubst_mv_32f (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp32f* pSrc2, Ipp32f* pDst,
    Ipp32u widthHeight);
IppStatus ippmLUBackSubst_mv_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2, Ipp32u
    widthHeight);
IppStatus ippmLUBackSubst_mv_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u
    widthHeight);
IppStatus ippmLUBackSubst_mv_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmLUBackSubst_mv_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmLUBackSubst_mv_32f_4x4 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmLUBackSubst_mv_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmLUBackSubst_mv_32f_5x5 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmLUBackSubst_mv_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmLUBackSubst_mv_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp32f* pSrc2, Ipp32f* pDst);
```

```
IppStatus ippmLUBackSubst_mv_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
```

LUBackSubst 関数 : 64f データの行列 - ベクトルの演算

```
IppStatus ippmLUBackSubst_mv_64f (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp64f* pSrc2, Ipp64f* pDst,
    Ipp32u widthHeight);
IppStatus ippmLUBackSubst_mv_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2, Ipp32u
    widthHeight);
IppStatus ippmLUBackSubst_mv_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    widthHeight);
IppStatus ippmLUBackSubst_mv_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmLUBackSubst_mv_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmLUBackSubst_mv_64f_4x4 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmLUBackSubst_mv_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmLUBackSubst_mv_64f_5x5 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmLUBackSubst_mv_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmLUBackSubst_mv_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmLUBackSubst_mv_64f_6x6 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32u* pSrcIndex, const Ipp64f* pSrc2, Ipp64f* pDst);
```

IppStatus **ippmLUBackSubst_mv_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmLUBackSubst_mv_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

LUBackSubst 関数 : 32f データの行列 - ベクトル配列の演算

IppStatus **ippmLUBackSubst_mva_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *widthHeight*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

LUBackSubst 関数 : 64f データの行列 - ベクトル配列の演算

```
IppStatus ippmLUBackSubst_mva_64f (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32u* pSrcIndex, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    widthHeight, Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u*
    pSrcIndex, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u
    widthHeight, Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u widthHeight, Ipp32u
    count);
IppStatus ippmLUBackSubst_mva_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32u* pSrcIndex, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u*
    pSrcIndex, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u widthHeight, Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32u* pSrcIndex, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmLUBackSubst_mva_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp32u*
    pSrcIndex, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32u* pSrcIndex, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmLUBackSubst_mva_64f_3x3_PS2 (const Ipp64f** pSrc1,
    Ipp32s src1ROIShift, Ipp32s src1Stride2, Ipp32u* pSrcIndex, const
    Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f**
    pDst, Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmLUBackSubst_mva_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmLUBackSubst_mva_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32u* *pSrcIndex*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

最小 2 乗問題の解の関数のプロトタイプ

E

QRDecomp

この関数は、ピボット演算を使用せずに、行列の QR 分解を計算する。この関数の説明とすべての引数のリストは、QRDecomp を参照のこと。

QRDecomp 関数 : 32f データの行列の演算

```
IppStatus ippmQRDecomp_m_32f (const Ipp32f* pSrc, Ipp32s srcStride1,
    Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height);
IppStatus ippmQRDecomp_m_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u
    height);
IppStatus ippmQRDecomp_m_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u width, Ipp32u height);
IppStatus ippmQRDecomp_m_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRDecomp_m_32f_4x4 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_32f_4x4_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_32f_4x4_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRDecomp_m_32f_5x5 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_32f_5x5_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
```

```
IppStatus ippmQRDecomp_m_32f_5x5_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRDecomp_m_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift);
```

QRDecomp 関数 : 64f データの行列の演算

```
IppStatus ippmQRDecomp_m_64f (const Ipp64f* pSrc, Ipp32s srcStride1,
    Ipp64f* pBuffer, Ipp64f* pDst, Ipp32s dstStride1, Ipp32u width,
    Ipp32u height);
IppStatus ippmQRDecomp_m_64f_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pBuffer, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2, Ipp32u width, Ipp32u
    height);
IppStatus ippmQRDecomp_m_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pBuffer, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u width, Ipp32u height);
IppStatus ippmQRDecomp_m_64f_3x3 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pBuffer, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_64f_3x3_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pBuffer, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_64f_3x3_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pBuffer, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRDecomp_m_64f_4x4 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pBuffer, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_64f_4x4_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pBuffer, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_64f_4x4_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pBuffer, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRDecomp_m_64f_5x5 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pBuffer, Ipp64f* pDst, Ipp32s dstStride1);
IppStatus ippmQRDecomp_m_64f_5x5_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pBuffer, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_64f_5x5_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pBuffer, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRDecomp_m_64f_6x6 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp64f* pBuffer, Ipp64f* pDst, Ipp32s dstStride1);
```

```
IppStatus ippmQRDecomp_m_64f_6x6_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride1, Ipp32s srcStride2, Ipp64f* pBuffer, Ipp64f* pDst,
    Ipp32s dstStride1, Ipp32s dstStride2);
IppStatus ippmQRDecomp_m_64f_6x6_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pBuffer, Ipp64f** pDst, Ipp32s dstROIShift);
```

QRDecomp 関数 : 32f データの行列配列の演算

```
IppStatus ippmQRDecomp_ma_32f (const Ipp32f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp32f* pBuffer, Ipp32f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f*
    pBuffer, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u
    height, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pBuffer, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u width, Ipp32u
    height, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f*
    pBuffer, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_3x3 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_3x3_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f*
    pBuffer, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_3x3_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_3x3_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_3x3_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pBuffer, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
```

IppStatus **ippmQRDecomp_ma_32f_3x3_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_4x4** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_4x4_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_4x4_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_4x4_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_4x4_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_4x4_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_5x5** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_5x5_S2** (const Ipp32f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_5x5_P** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_5x5_PS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_5x5_L** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_32f_5x5_LS2** (const Ipp32f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp32f* *pBuffer*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

```
IppStatus ippmQRDecomp_ma_32f_6x6 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32f* pBuffer, Ipp32f* pDst,
    Ipp32s dstStride0, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_6x6_S2 (const Ipp32f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f*
    pBuffer, Ipp32f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_6x6_P (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32f* pBuffer, Ipp32f** pDst, Ipp32s dstROIShift,
    Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_6x6_PS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp32f* pBuffer, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_6x6_L (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32f* pBuffer, Ipp32f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u count);
IppStatus ippmQRDecomp_ma_32f_6x6_LS2 (const Ipp32f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp32f*
    pBuffer, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u count);
```

QRDecomp 関数 : 64f データの行列配列の演算

```
IppStatus ippmQRDecomp_ma_64f (const Ipp64f* pSrc, Ipp32s srcStride0,
    Ipp32s srcStride1, Ipp64f* pBuffer, Ipp64f* pDst, Ipp32s
    dstStride0, Ipp32s dstStride1, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmQRDecomp_ma_64f_S2 (const Ipp64f* pSrc, Ipp32s
    srcStride0, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f*
    pBuffer, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRDecomp_ma_64f_P (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp64f* pBuffer, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRDecomp_ma_64f_PS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride2, Ipp64f* pBuffer, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u
    height, Ipp32u count);
IppStatus ippmQRDecomp_ma_64f_L (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp64f* pBuffer, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride1, Ipp32u width, Ipp32u
    height, Ipp32u count);
IppStatus ippmQRDecomp_ma_64f_LS2 (const Ipp64f** pSrc, Ipp32s
    srcROIShift, Ipp32s srcStride1, Ipp32s srcStride2, Ipp64f*
    pBuffer, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride1,
    Ipp32s dstStride2, Ipp32u width, Ipp32u height, Ipp32u count);
```

IppStatus **ippmQRDecomp_ma_64f_3x3** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_3x3_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_3x3_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_3x3_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_3x3_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_3x3_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_4x4** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_4x4_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_4x4_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_4x4_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_4x4_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_4x4_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_5x5** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_5x5_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_5x5_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_5x5_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_5x5_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_5x5_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_6x6** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_6x6_S2** (const Ipp64f* *pSrc*, Ipp32s *srcStride0*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_6x6_P** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_6x6_PS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_6x6_L** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32u *count*);

IppStatus **ippmQRDecomp_ma_64f_6x6_LS2** (const Ipp64f** *pSrc*, Ipp32s *srcROIShift*, Ipp32s *srcStride1*, Ipp32s *srcStride2*, Ipp64f* *pBuffer*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride1*, Ipp32s *dstStride2*, Ipp32u *count*);

QRBackSubst

この関数は、ピボット演算を使用せずに、QR 分解された行列の最小 2 乗問題を解く。この関数の説明とすべての引数のリストは、QRBackSubst を参照のこと。

QRBackSubst 関数 : 32f データの行列 - ベクトルの演算

```
IppStatus ippmQRBackSubst_mv_32f (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32f* pBuffer, const Ipp32f* pSrc2, Ipp32f* pDst,
    Ipp32u width, Ipp32u height);
IppStatus ippmQRBackSubst_mv_32f_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32f* pBuffer, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height);
IppStatus ippmQRBackSubst_mv_32f_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f* pBuffer, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift, Ipp32u width,
    Ipp32u height);
IppStatus ippmQRBackSubst_mv_32f_3x3 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32f* pBuffer, const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmQRBackSubst_mv_32f_3x3_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32f* pBuffer, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_32f_3x3_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f* pBuffer, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRBackSubst_mv_32f_4x4 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32f* pBuffer, const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmQRBackSubst_mv_32f_4x4_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32f* pBuffer, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_32f_4x4_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f* pBuffer, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRBackSubst_mv_32f_5x5 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32f* pBuffer, const Ipp32f* pSrc2, Ipp32f* pDst);
IppStatus ippmQRBackSubst_mv_32f_5x5_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32f* pBuffer, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_32f_5x5_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f* pBuffer, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRBackSubst_mv_32f_6x6 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32f* pBuffer, const Ipp32f* pSrc2, Ipp32f* pDst);
```

```
IppStatus ippmQRBackSubst_mv_32f_6x6_S2 (const Ipp32f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp32f* pBuffer, const Ipp32f*
    pSrc2, Ipp32s src2Stride2, Ipp32f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_32f_6x6_P (const Ipp32f** pSrc1, Ipp32s
    src1ROIShift, Ipp32f* pBuffer, const Ipp32f** pSrc2, Ipp32s
    src2ROIShift, Ipp32f** pDst, Ipp32s dstROIShift);
```

QRBackSubst 関数 : 64f データの行列 - ベクトルの演算

```
IppStatus ippmQRBackSubst_mv_64f (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp64f* pBuffer, const Ipp64f* pSrc2, Ipp64f* pDst,
    Ipp32u width, Ipp32u height);
IppStatus ippmQRBackSubst_mv_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp64f* pBuffer, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2, Ipp32u
    width, Ipp32u height);
IppStatus ippmQRBackSubst_mv_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f* pBuffer, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u width,
    Ipp32u height);
IppStatus ippmQRBackSubst_mv_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp64f* pBuffer, const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmQRBackSubst_mv_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp64f* pBuffer, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f* pBuffer, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRBackSubst_mv_64f_4x4 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp64f* pBuffer, const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmQRBackSubst_mv_64f_4x4_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp64f* pBuffer, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_64f_4x4_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f* pBuffer, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRBackSubst_mv_64f_5x5 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp64f* pBuffer, const Ipp64f* pSrc2, Ipp64f* pDst);
IppStatus ippmQRBackSubst_mv_64f_5x5_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp32s src1Stride2, Ipp64f* pBuffer, const Ipp64f*
    pSrc2, Ipp32s src2Stride2, Ipp64f* pDst, Ipp32s dstStride2);
IppStatus ippmQRBackSubst_mv_64f_5x5_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f* pBuffer, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift);
IppStatus ippmQRBackSubst_mv_64f_6x6 (const Ipp64f* pSrc1, Ipp32s
    src1Stride1, Ipp64f* pBuffer, const Ipp64f* pSrc2, Ipp64f* pDst);
```

IppStatus **ippmQRBackSubst_mv_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride2*);

IppStatus **ippmQRBackSubst_mv_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*);

QRBackSubst 関数 : 32f データの行列 - ベクトル配列の演算

IppStatus **ippmQRBackSubst_mva_32f** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *width*, Ipp32u *height*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_3x3** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_3x3_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_3x3_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_3x3_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_3x3_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_3x3_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_4x4** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_4x4_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_4x4_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_4x4_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_4x4_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_4x4_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_5x5** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_5x5_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_5x5_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_5x5_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_5x5_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_5x5_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_6x6** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_6x6_S2** (const Ipp32f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp32f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_6x6_P** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_6x6_PS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_6x6_L** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_32f_6x6_LS2** (const Ipp32f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp32f* *pBuffer*, const Ipp32f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp32f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

QRBackSubst 関数 : 64f データの行列 - ベクトル配列の演算

```
IppStatus ippmQRBackSubst_mva_64f (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp64f* pBuffer, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    width, Ipp32u height, Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp64f*
    pBuffer, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f* pBuffer, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u width,
    Ipp32u height, Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f* pBuffer, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u width, Ipp32u height,
    Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_L (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp64f* pBuffer, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_LS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp64f*
    pBuffer, const Ipp64f** pSrc2, Ipp32s src2ROIShift, Ipp32s
    src2Stride2, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32s dstStride2,
    Ipp32u width, Ipp32u height, Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_3x3 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp64f* pBuffer, const Ipp64f*
    pSrc2, Ipp32s src2Stride0, Ipp64f* pDst, Ipp32s dstStride0, Ipp32u
    count);
IppStatus ippmQRBackSubst_mva_64f_3x3_S2 (const Ipp64f* pSrc1, Ipp32s
    src1Stride0, Ipp32s src1Stride1, Ipp32s src1Stride2, Ipp64f*
    pBuffer, const Ipp64f* pSrc2, Ipp32s src2Stride0, Ipp32s
    src2Stride2, Ipp64f* pDst, Ipp32s dstStride0, Ipp32s dstStride2,
    Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_3x3_P (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp64f* pBuffer, const Ipp64f** pSrc2, Ipp32s
    src2ROIShift, Ipp64f** pDst, Ipp32s dstROIShift, Ipp32u count);
IppStatus ippmQRBackSubst_mva_64f_3x3_PS2 (const Ipp64f** pSrc1, Ipp32s
    src1ROIShift, Ipp32s src1Stride2, Ipp64f* pBuffer, const Ipp64f**
    pSrc2, Ipp32s src2ROIShift, Ipp32s src2Stride2, Ipp64f** pDst,
    Ipp32s dstROIShift, Ipp32s dstStride2, Ipp32u count);
```

IppStatus **ippmQRBackSubst_mva_64f_3x3_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_3x3_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_4x4** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_4x4_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_4x4_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_4x4_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_4x4_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_4x4_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_5x5** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_5x5_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_5x5_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_5x5_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_5x5_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_5x5_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_6x6** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_6x6_S2** (const Ipp64f* *pSrc1*, Ipp32s *src1Stride0*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f* *pSrc2*, Ipp32s *src2Stride0*, Ipp32s *src2Stride2*, Ipp64f* *pDst*, Ipp32s *dstStride0*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_6x6_P** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_6x6_PS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_6x6_L** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32u *count*);

IppStatus **ippmQRBackSubst_mva_64f_6x6_LS2** (const Ipp64f** *pSrc1*, Ipp32s *src1ROIShift*, Ipp32s *src1Stride1*, Ipp32s *src1Stride2*, Ipp64f* *pBuffer*, const Ipp64f** *pSrc2*, Ipp32s *src2ROIShift*, Ipp32s *src2Stride2*, Ipp64f** *pDst*, Ipp32s *dstROIShift*, Ipp32s *dstStride2*, Ipp32u *count*);

索 引

A

Add (行列), 5-12
Add (ベクトル), 4-3

C

Copy, 3-1
CrossProduct, 4-8

D

Det, 5-5
DotProduct, 4-10

E

Extract, 3-3

F

FrobNorm, 5-4

G

Gaxpy, 5-16

I

Invert, 5-3
ippmAdd (行列), 5-12
ippmAdd (ベクトル), 4-3
ippmCopy, 3-1
ippmCrossProduct, 4-8
ippmDet, 5-5
ippmDotProduct, 4-10
ippmExtract, 3-3
ippmFrobNorm, 5-4
ippmGaxpy, 5-16
ippmInvert, 5-3
ippmL2Norm, 4-11
ippmLComb, 4-13
ippmLoadIdentity, 3-4
ippmMul (行列), 5-8
ippmMul (ベクトル), 4-7
ippmSaxpy, 4-1
ippmSub (行列), 5-14
ippmSub (ベクトル), 4-5
ippmTrace, 5-6
ippmTranspose, 5-1
IPP ソフトウェア , 1-1
IPP の関数の命名 , 2-7

L

L2Norm, 4-11
LComb, 4-13
LoadIdentity, 3-4
L- ポインタ , 2-3

M

Mul (行列), 5-8
Mul (ベクトル), 4-7

P

P- ポインタ , 2-3

R

ROI (Region of Interest), 2-1
ROI シフト , 2-2

S

Saxpy, 4-1
Sub (行列), 5-14
Sub (ベクトル), 4-5

S- ポインタ , 2-3

T

Trace, 5-6
Transpose, 5-1

あ

アドレス指定 , 2-3
暗黙のパラメータ , 2-8

お

オブジェクト、関数に渡される , 2-7
オブジェクトのシーケンシャルな配列 , 2-4

か

概要 , 1-1
関数の基本名 , 2-7
関数の説明、本書の各章の , 1-2
関数の引数 , 2-9

き

行列演算用 IPP の目的 , 2-1
行列代数関数
- Add, 5-12
- Det, 5-5
- FrobNorm, 5-4
- Gaxpy, 5-16
- Invert, 5-3
- Mul, 5-8
- Sub, 5-14
- Trace, 5-6
- Transpose, 5-1

さ

最小 2 乗問題関数
- QRBackSubst, 7-3
- QRDecomp, 7-1

サイズ、行列の , 2-8
サイズ、ベクトルの , 2-8

し

字体の規則 , 1-3
順序、パラメータの , 2-9

す

ストライド , 2-3

て

ディスクリプタ , 2-8
データ・タイプ , 2-8

に

任意のオブジェクト配列 , 2-5

は

はじめに , 2-1

ひ

引数、IPP 関数の , 2-9
必要なハードウェアとソフトウェアの構成 , 1-1
表記規則 , 1-3

へ

ベクトル代数関数
- Add, 4-3
- CrossProduct, 4-8
- DotProduct, 4-10
- L2Norm, 4-11
- LComb, 4-13
- Mul, 4-7
- Saxpy, 4-1
- Sub, 4-5

ほ

ポインタ , 2-3
本書について , 1-2
本書の構成 , 1-2
本書の対象読者 , 1-3

め

命名規則 , 1-3

ゆ

ユーティリティ関数
Copy, 3-1
Extract, 3-3
LoadIdentity, 3-4

れ

連立 1 次方程式の解の関数
LUBackSubst, 6-3
LUDecomp, 6-1