ViCue Video Codec Analyzer 2018

Release Notes

(Version 3.7.0)

Overview

The **ViCue Video Codec Analyzer** (VCA) is a graphical coded video bitstream analysis tool, supporting the following coding standards:

- HEVC: (ISO/IEC 23008-2 MPEG-H Part 2 or ITU-T H.265) , 8/10-bit
- HEVC: RExt extension, 8/10/12-bit, 4:0:0/4:2:0/4:2:2/4:4:4
- HEVC: SCC extension, conform to HM + SCM 8.6 in reference code
- HEVC Scalable/Multiview Extension
- AV1, SHA-1 (Version 1.0): d14c5bb4f336ef1842046089849dee4a301fbbf0
- Google's VP9, profiles 0,1,2,3, 4:2:0/4:2:2/4:4:0/4:4:4, 8/10/12-bit
- AVC: (H.264/AVC, ISO/IEC 14496-10, MPEG-4 Part 10), except SVC/MVC
- MPEG2 (ISO/IEC 13818-2 Part 2), 4:2:0/4:2:2, 8-bit
- MKV container
- MP4 container
- MPEG2 TS/PS container
- AVI container
- Mpeg Media Transport container (ARIB STD-B60 1.0)

Once a bitstream is loaded, the tool allows the user to inspect each major step of the decode process visually and numerically, and the structure of the coded image can be explored. This data can be used as a visual reference when learning about HEVC/VP9/AVC/MPEG2/AV1 or when debugging a particular encoder or decoder.

New Features

- add overral stream's stats for console mode (last row - stream's stats)
- Y4M: add console output at y4m format (-y4m option, yuv420/422/444 8-bits only).
- Heat map: add viridis & jet colors maps. Screenshot below
- AV1:
  - add tooltips for syntax elements. Screenshot below
  - add palette details support and collection of palette stats.
  - add disable LR type (if LR_FRAME_TYPE == NONE).
  - add metadata and tile_list syntax elements.
  - add annexB support for containers
- VP9: add level constraints checks ("Run full stream analysis" button). Screenshot below
- add "Keyboard Sequence" dialog at help menu. It contains current hotkeys. Screenshot below
- redesign activation dialog, add more meaningful error messages, add proxy's tab.

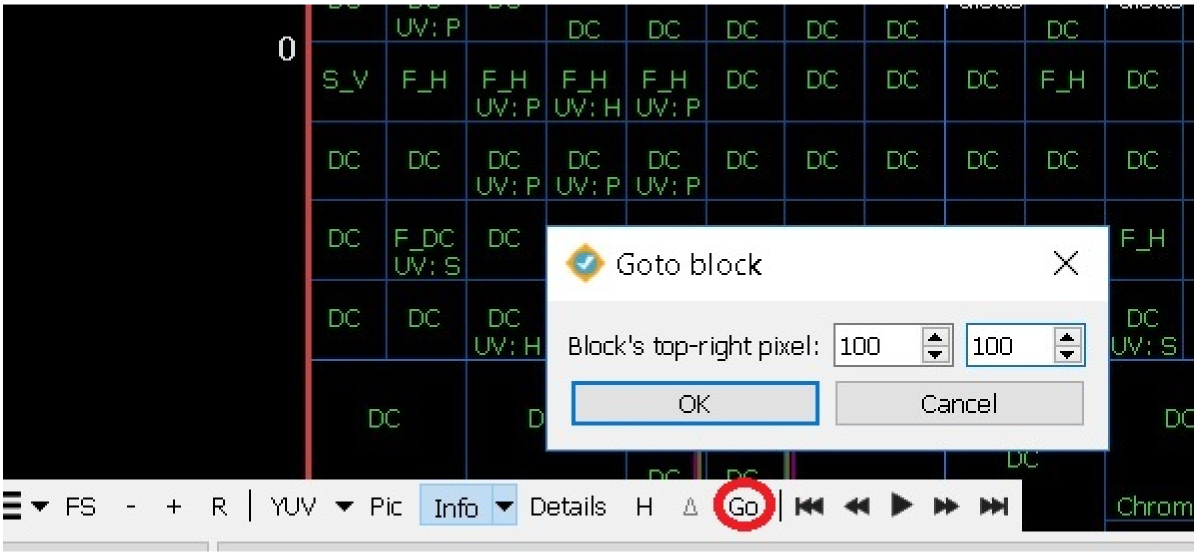
- add "Info" filtering for all codecs (show/hide blocks splits, tiles/slices, mv, etc). "Options/Show MV in Prediction mode" was removed. Use "Info/MV" in prediction mode.Screenshot below
- add "Go" to block at info panel toolbar for all codecs. Screenshot below
- add "Open dependent bitstream as..." option.
- change console "-stats" option to "-stats filename".

Bug Fixes

- HEVC: fix comparing with dependent view at loop filter mode.
- AV1:
  - fix annexB support.
  - fix collection of mv stats. Include intraBC case.
- HEVC/AVC:
  - fix drawing and calculation HRD for streams with HRD violence.
  - restore performance of console mode (avoid to decoding unnecessary frames).
- prevent stack overflow: use loop instead of recursive method for calculation requered pictures (There was crash if tried to navigate > 5000 frame on single idr clip).

Screenshots

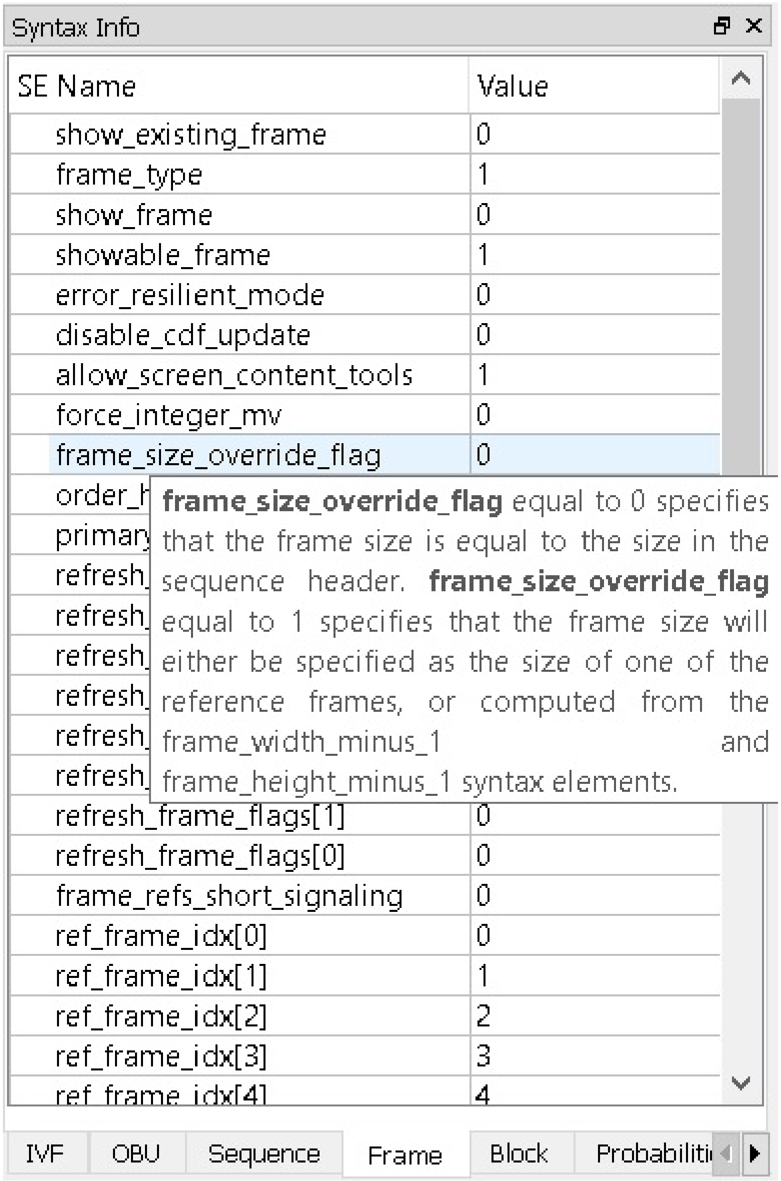
Goto block



Goto block

Block's top-right pixel: 100 100

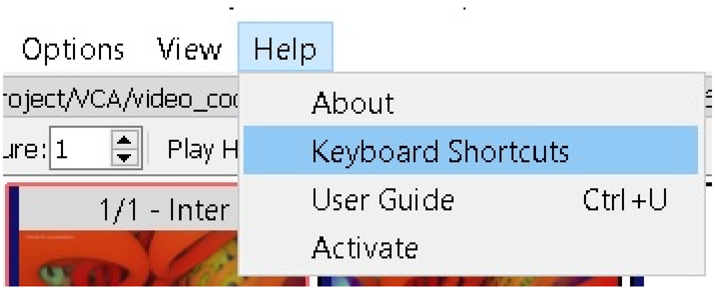
Syntax elements tooltips



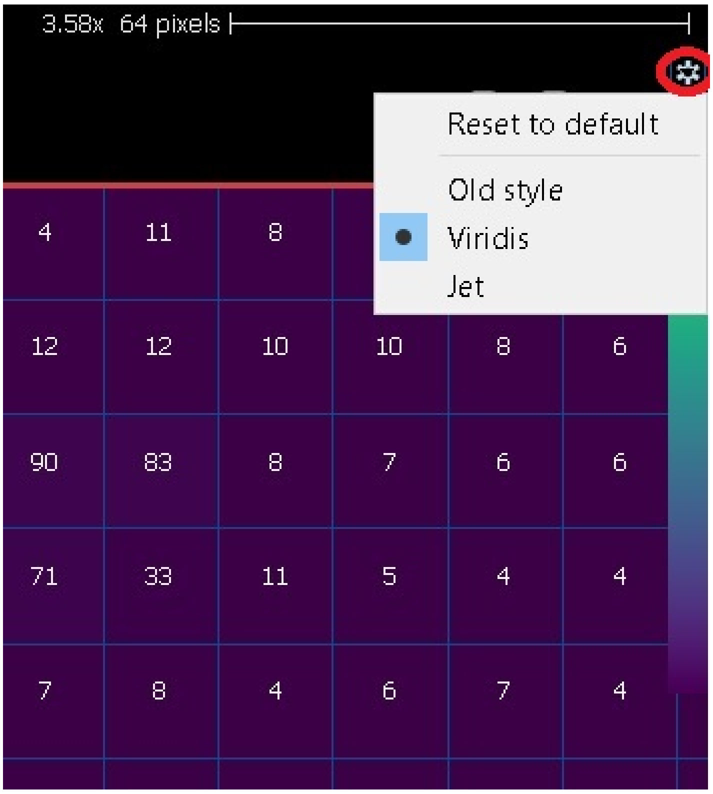
SE Name	Value
show_existing_frame	0
frame_type	1
show_frame	0
showable_frame	1
error_resilient_mode	0
disable_cdf_update	0
allow_screen_content_tools	1
force_integer_mv	0
frame_size_override_flag	0
order_h	
primary	
refresh_	
refresh_	
refresh_	
refresh_	
refresh_	
refresh_	
refresh_frame_flags[1]	0
refresh_frame_flags[0]	0
frame_refs_short_signaling	0
ref_frame_idx[0]	0
ref_frame_idx[1]	1
ref_frame_idx[2]	2
ref_frame_idx[3]	3
ref_frame_idx[4]	4

frame_size_override_flag equal to 0 specifies that the frame size is equal to the size in the sequence header. **frame_size_override_flag** equal to 1 specifies that the frame size will either be specified as the size of one of the reference frames, or computed from the frame_width_minus_1 and frame_height_minus_1 syntax elements.

Keyboard Sequence

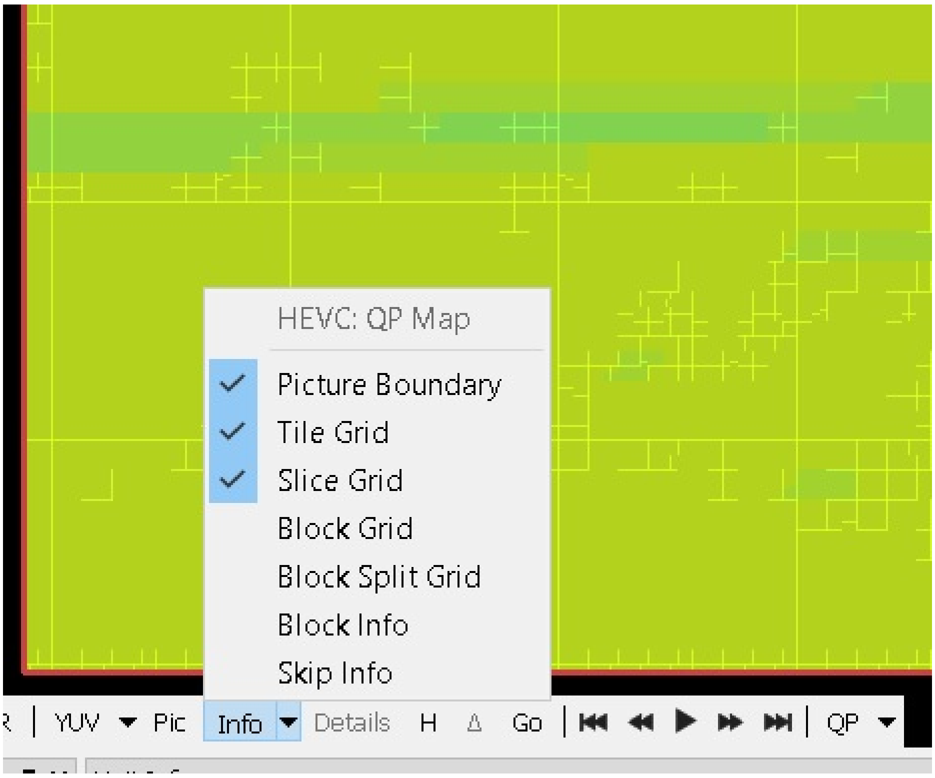


Heat map colors

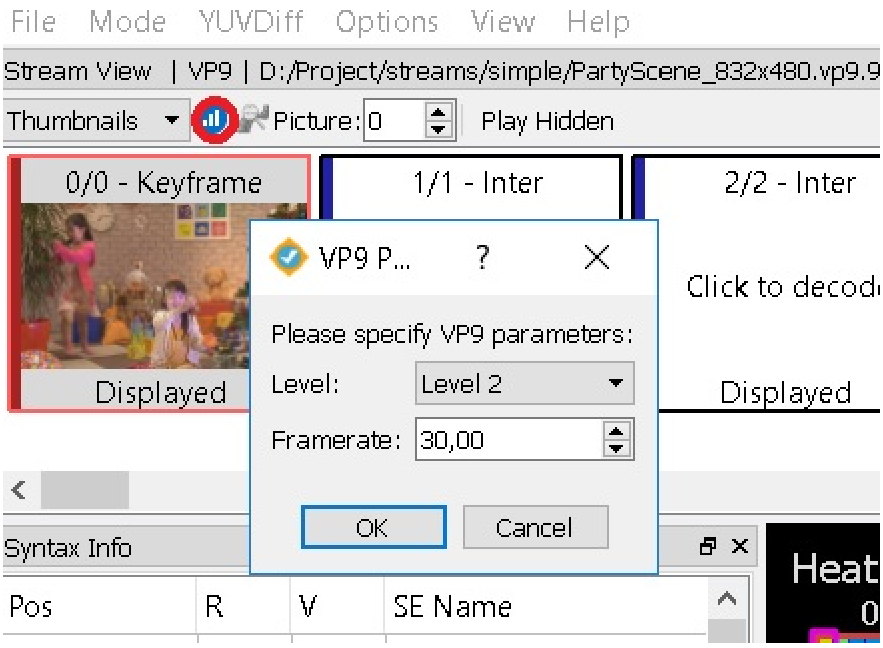


4	11	8			
12	12	10	10	8	6
90	83	8	7	6	6
71	33	11	5	4	4
7	8	4	6	7	4

Info filtering



VP9 level checks



Please specify VP9 parameters:

Level: Level 2

Framerate: 30,00

System Requirements

Hardware

- 1GB RAM minimum, 4GB recommended when loading 4K pictures.

Software

- Microsoft* Windows* 7, Microsoft* Windows* 8, Microsoft* Windows* 8.1, Microsoft* Windows* 10, 32-bit/64-bit.
- Ubuntu* Linux* 16.04 and higher, SLES* 12 and higher, CentOS* 7.2 and higher
- MacOS* 10.9 and higher

Package Contents

content: ViCue Video Codec Analyzer Release Notes (this file), ViCue Video Codec Analyzer User Guide, licenses, binary executables, tool's internal and third-party binaries.

Installation

To begin installation:

- Double-click on the executable file to begin installation.

Activation during installation You must activate the product to finish installation. Use one of these methods:

- Activation using serial number. Internet connection is required;
- Remote activation using serial number. Used when your computer is not connected to the internet. You can use another computer with internet access;
- Activation using license file;
- Activation using license server.

- You can also evaluate the product for trial period.
  If you have problems with installation under MacOS*, please, check System Preferences -> Security & Privacy options under “Allow apps downloaded from:”. The option “Mac App Store and identified developers” should be checked.

Known Limitations

The ViCue Video Codec Analyzer has the following known limitations:

- Supported containers for AV1/VP9 are IVF and WebM (MKV) only.
- AVC/HEVC HRD buffer fullness uses only Type II bitstream for visualization.
- Archive yuv files (*.zip, *gz) fully unpacked in memory currently, huge files could require a lot of memory.
- For zip files only first file in archive is considered to be used as compressed yuv file. Only deflate and store compression methods are supported.
- Archived y4m files are not supported yet.
- AVC decoding does not support gaps in frame_num and slice groups currently.
- Dual view mode is in preview implementation now. It can be unstable and buggy.

AV1 limitations:

- Memory consumption can be high on 4K resolutions.
- large scale tile is not supported.
- "Extract selected units" from OBU panel doesn't work properly for raw obu annexb case.

HEVC Conformance checks:

- Bitstream conformance to active profile, tier and level representation and HRD parameters is tested on general representation only. (PTL syntax elements with prefix ’general_’)
- Bitstream conformance to HRD parameters is tested using VUI parameters specified in active SPS only.
- For Intra only profiles picture reordering is not tested.
- Common Multilayer HEVC: external base layer configurations are not supported.
- HEVC Slice data errors: bitstream restrictions specified SPS VUI or VPS VUI parameters are not supported.

HEVC Decoder behavior:

- For Range exntensions, Highthroughput and Screen-Extended (SCC) profiles in order to detect sub-profile (e.g. Monochrome, Main 444 12bit, ...) bitstream constraint flags, coded in PTL, must be set according to the tables specified in clause A.3 of the spec.
- For Still Picture profiles test for number of pictures present in bitstream is applied to whole bitstream file.
- For Screen-Extended (SCC) profiles in order to detect these profiles, general_profile_idc must be set to 31 or general_compatibility_flag[31] must be set to 1. (same as of HM16.16+SCC8.5) (Will be changed to 9 (according to spec) in future releases)
- Multilayer bitstreams are decoded with highest available index of Layer Set (TargetOlsIdx is set to NumOutputLayerSets – 1). All available layers will be output.
- Different layers in Multiview HEVC stream must have the same resolution and chroma format for inter-layer prediction. (Configurations with different chroma format or resolution are still decodable, but resampling process for inter-layer prediction will be invoked in that cases)
- SCC extensions presence in bitstream is tested on both position 3 and 6 of sps_extension_bit and pps_extension_bit.

Legal Information

Qt

Qt licensed under the GNU Lesser General Public License (LGPL) version 2.1. It is appropriate for the development of Qt applications provided you can comply with the terms and conditions of the GNU LGPL version 2.1. For details, please refer to http://qt-project.org/doc/qt-5/licensing.html for specific details.

Zlib

zlib.h -- interface of the 'zlib' general purpose compression library version 1.2.8, April 28th, 2013

Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org Mark Adler madler@alumni.caltech.edu

Qwt

Qwt License Version 1.0, January 1, 2003 The Qwt library and included programs are provided under the terms of the GNU LESSER GENERAL PUBLIC LICENSE (LGPL) with the following exceptions:

1. Widgets that are subclassed from Qwt widgets do not constitute a derivative work.
2. Static linking of applications and widgets to the Qwt library does not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared Qwt libraries, or link their applications or widgets against a user-supplied version of Qwt. If you link the application or widget to a modified version of Qwt, then the changes to Qwt must be provided under the terms of the LGPL in sections 1, 2, and 4.
3. You do not have to provide a copy of the Qwt license with programs that are linked to the Qwt library, nor do you have to identify the Qwt license in your program or documentation as required by section 6 of the LGPL. However, programs must still identify their use of Qwt. The following example statement can be included in user documentation to satisfy this requirement: [program/widget] is based in part on the work of the Qwt project http://qwt.sf.net