Intel® Video Pro Analyzer User Guide

Version 1.1.0

LEGAL DISCLAIMER

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web Site.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

Intel, the Intel logo, Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Revision History:

Revision Number	Date	Description
1.0	10/2/13	Initial Draft
1.1	1/2/14	Updated text and screenshots for new features

Contents

1 Overview

Intel® Video Pro Analyzer is a graphical coded video bitstream analysis tool, supporting both major next-generation coding standards:

- HEVC: (ISO/IEC 23008-2 MPEG-H Part 2 or ITU-T H.265)
- Google* VP9*

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 or when debugging a particular encoder or decoder.

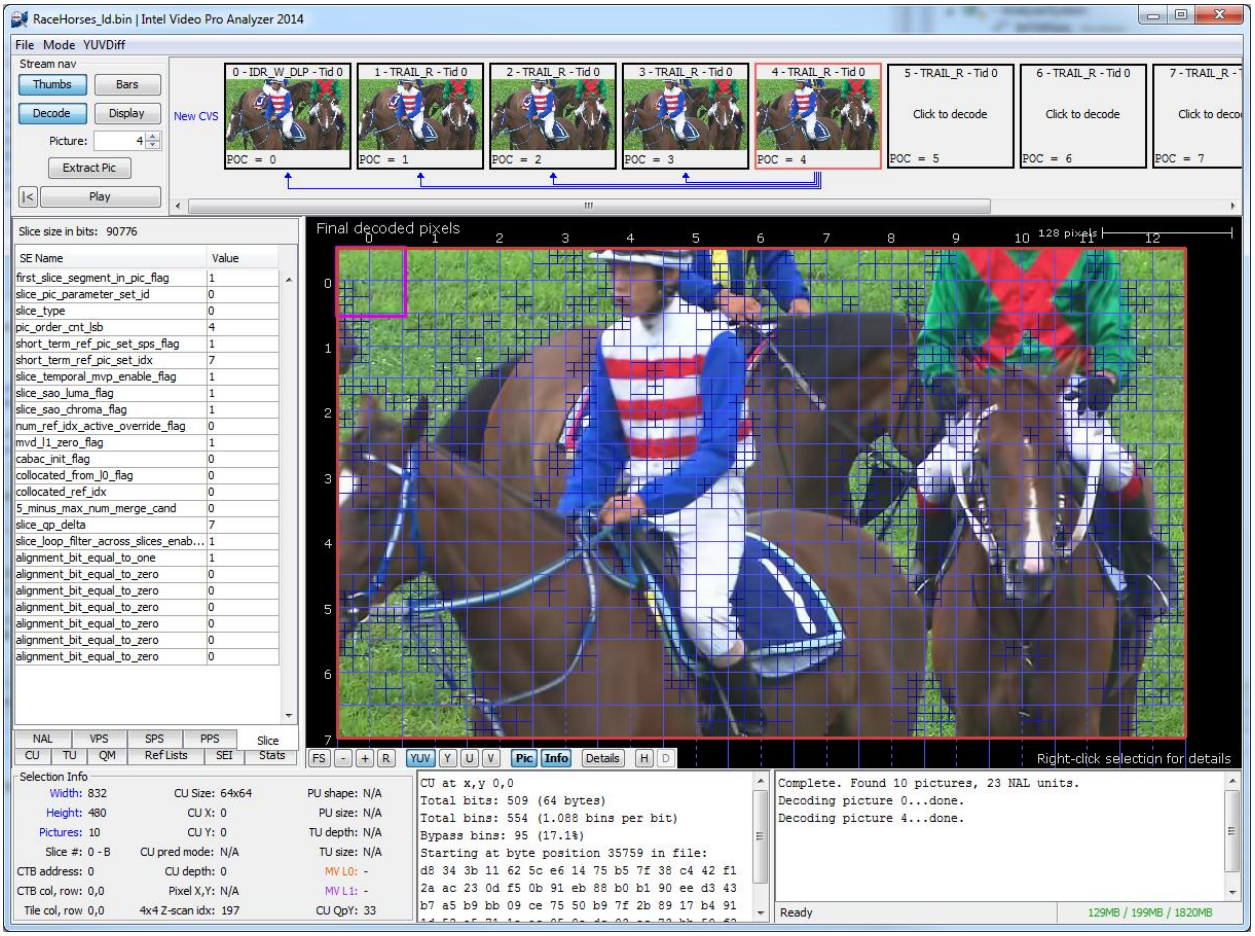
Intel Video Pro Analyzer is written in Java* and requires a Java Runtime Environment (JRE) version 7 or higher to be present on the system. Because of this the tool can run on Windows*, Linux*, Mac OS* or any other operating system that supports Java 7. A 64-bit JRE is highly recommended.

2 Document Conventions, Symbols, and Terms

Term	Definition
CABAC	Context-adaptive binary arithmetic coding
CTB	Coding Tree Block
CU	Coding Unit
CVC	Coded Video Stream
GUI/UI	(Graphical) User Interface
HEVC	High Efficiency Video Coding
NAL	Network Abstraction Layer
PPS	Picture Parameter Set
PU	Prediction Unit
QM	Quantization Matrix
SAO	Sample adaptive offset
SEI	Supplemental Enhancement Information
SPS	Sequence Parameter Set
TU	Transform Unit
TU	Transform Unit
VP9*	Google* video codec, VP9 is a successor to VP8*.
VPS	Video Parameter Set
YUV	Color space (YUV)

3 HEVC

The following sections describe all available features when loading an HEVC bitstream. The supported format of a bitstream is the raw bitstream with no surrounding container. Output from the publicly available HEVC reference software HM is in this format and can be opened directly. Intel® Video Pro Analyzer supports Main, Main 10, and Main Still Picture profiles. Sequences that go beyond the profile limits maybe be supported as well.



3.1 UI Components

The following sections describe the various components in the UI when an HEVC bitstream is loaded.

3.1.1 Top Filmstrip

The top filmstrip is a horizontally scrolling overview of the pictures in the bitstream. When a new sequence begins (CVS from the HEVC spec), a label indicating this is inserted before the first picture of the sequence. The current picture is highlighted with a red border. On top is the picture number - the decode order index in the bitstream - and the picture type as indicated by the NAL unit types, and on bottom is the POC value. Referenced pictures for the current picture are indicated with arrows. Green arrows indicate the long term references.

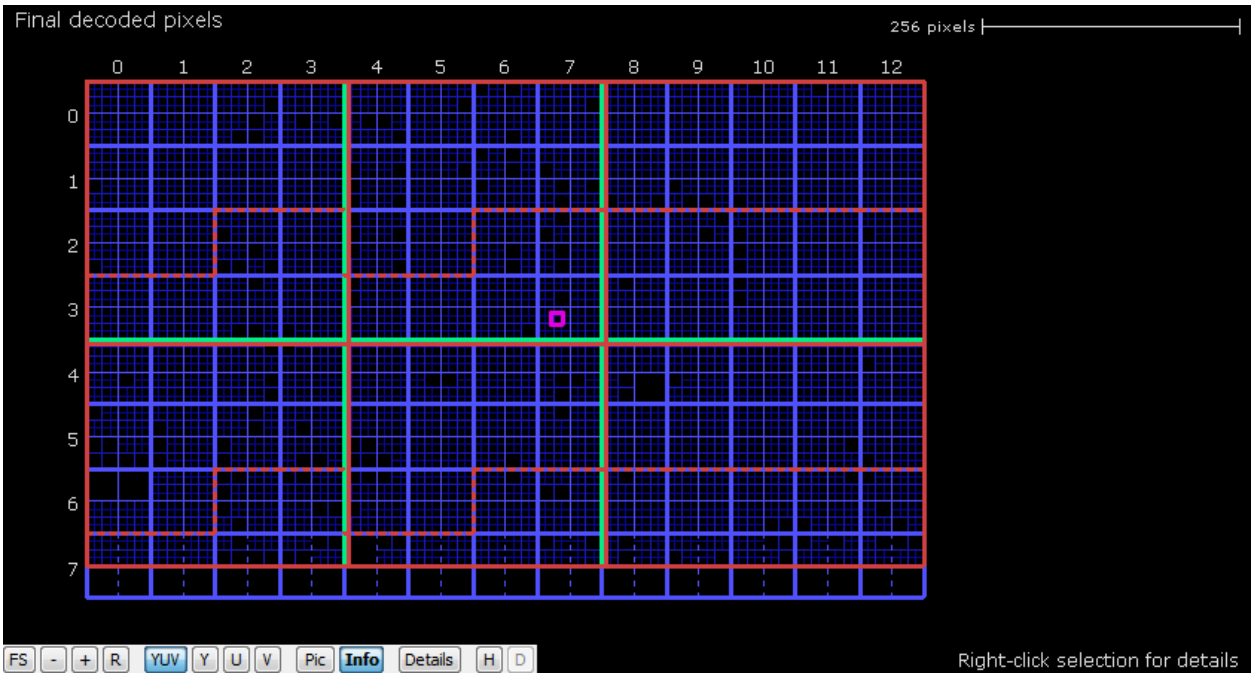
Clicking on a picture will cause Intel Video Pro Analyzer to decode it and mark it as the current picture, updating the rest of the UI correspondingly. If the new picture requires other pictures to be available as reference that have not yet been decoded, those will automatically be decoded.

Pictures are decoded on demand since it is impractical to store all details of all pictures in memory, especially for HD sequences. The current picture can also be selected by typing in a picture number in the box on the left of the panel. When using this box, the filmstrip is scrolled such that the newly selected current picture is visible.

On the left of the filmstrip there are buttons that let you choose between display order and decode order, and switch between thumbnails or bar graphs. A button called "Extract Pic" will write out a bitstream that contains the minimum number of pictures needed to decode the current picture. Typically this will consist of the current picture and all previous pictures that are reference pictures up to the nearest IRAP picture. This can be useful for debugging issues in long sequences.

The button labeled "|<" will rewind, causing the current picture to be set to the first picture in the loaded bitstream. The "Play" button will start playback of loaded bitstream at a reasonable rate, limited by CPU performance. All normal mode operations are possible during playback. Note that playback mode always operates in decode order.

3.1.2 Main Panel



The main panel displays the selected picture with visual annotations. The type of annotations and associated interactive behavior depends on the current mode, see 3.2 HEVC Modes. Click-dragging moves the picture around, and the mouse wheel zooms in or out about the mouse cursor.

At the top left of the panel, the current mode is displayed. Along the top right the current scale is shown. In all modes, the CTB row and column values are displayed along the left and top border of the picture, slices are indicated with thick red lines, dependent slice boundaries are dashed red lines and tile boundaries are thick green lines. The screenshot above shows an example of a stream that is 13x8 CTBs, has 6 tiles and a number of slice segments, some of which are dependent. The selected CU is outlined with a pink box.

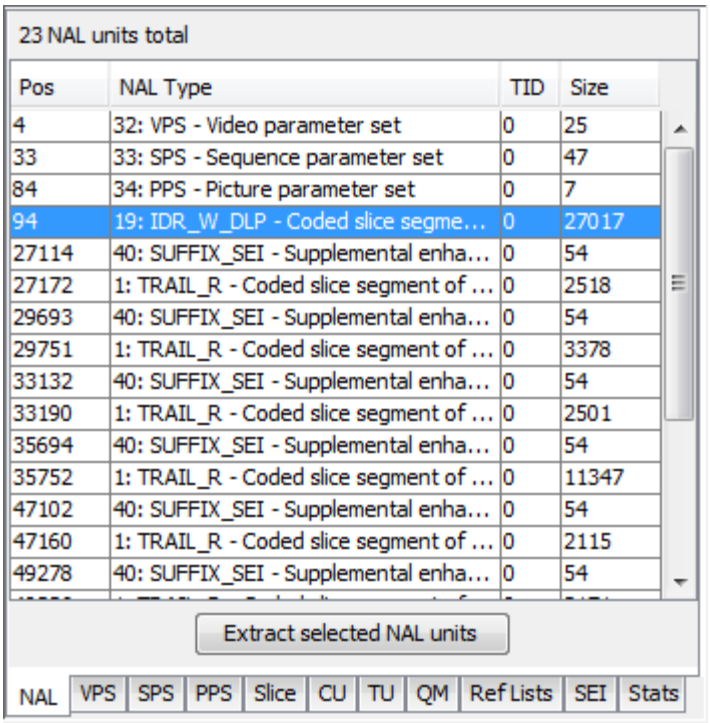
The bottom left has a cluster of buttons:

- FS: Toggles full-screen mode, which hides all UI elements except the main panel. Keyboard shortcut is F.
- +/-: Zooms in or out, centered on the center of the panel. Note that zooming with the mousewheel is much easier than using these buttons. Keyboard shortcuts are + and -.
- R: Resets the current zoom to fit the entire picture in the main panel. Keyboard shortcut is R.
- YUV, Y, U, V: Toggles between per-component image, or full-color YUV.
- Pic: Toggles the actual picture underneath the annotations. The picture is mode-dependent. For example in prediction mode, the picture is comprised of the prediction samples before the residual signal has been added in. It is often helpful to turn off the picture in order to make the annotation easier to read. Keyboard shortcut is P.
- Info: Toggles the annotations on or off. Keyboard shortcut is I.
- H/D: Toggles between hexadecimal and decimal display of the annotations. This applies to the values in the Left Tabs as well. Keyboard shortcuts are H and D.

3.1.3 Left Tabs

The left side of the UI is a resizable panel with several tabs that display information about the current selection in one way or another. The following subsections describe each tab.

3.1.3.1 NAL

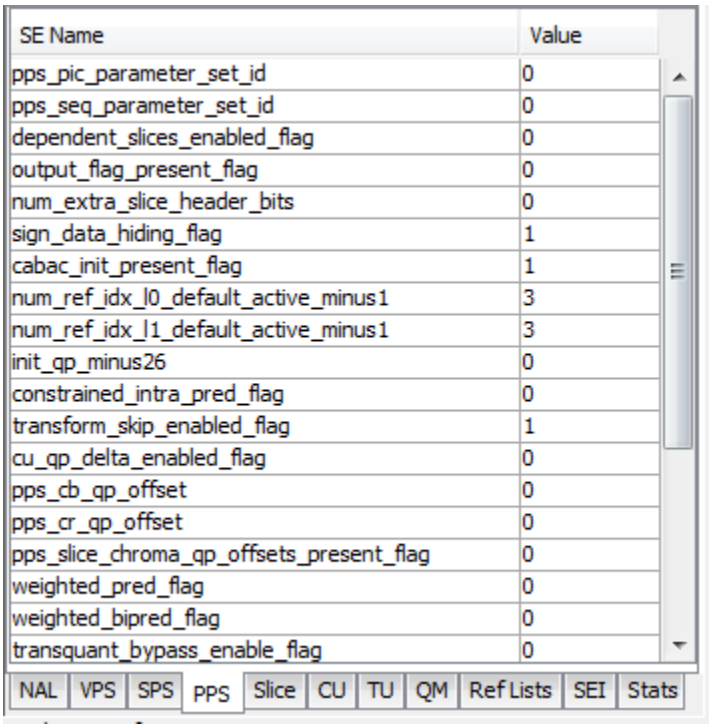


The NAL tab lists all NAL units found in the bitstream, in decode order. Above the list the total number of NAL units is displayed. The list has four columns: "Pos" is the byte position of the NAL unit in the file, "NAL Type" is the textual description of the NAL's type, "TID" is the Temporal ID of the NAL unit, and "Size" indicates the total number of bytes in the NAL unit.

Clicking on any NAL unit will display up to the first 500 raw bytes in the Raw Bytes panel. If a NAL unit is a VCL NAL unit, meaning it is a coded slice segment, it can be double-clicked in order to make the picture that it belongs to the current picture. Conversely, changing the current picture will highlight the NAL unit that contains the first slice segment of that picture. Selecting a CU in the Main Panel will highlight the NAL unit containing the slice that contains the selected CU.

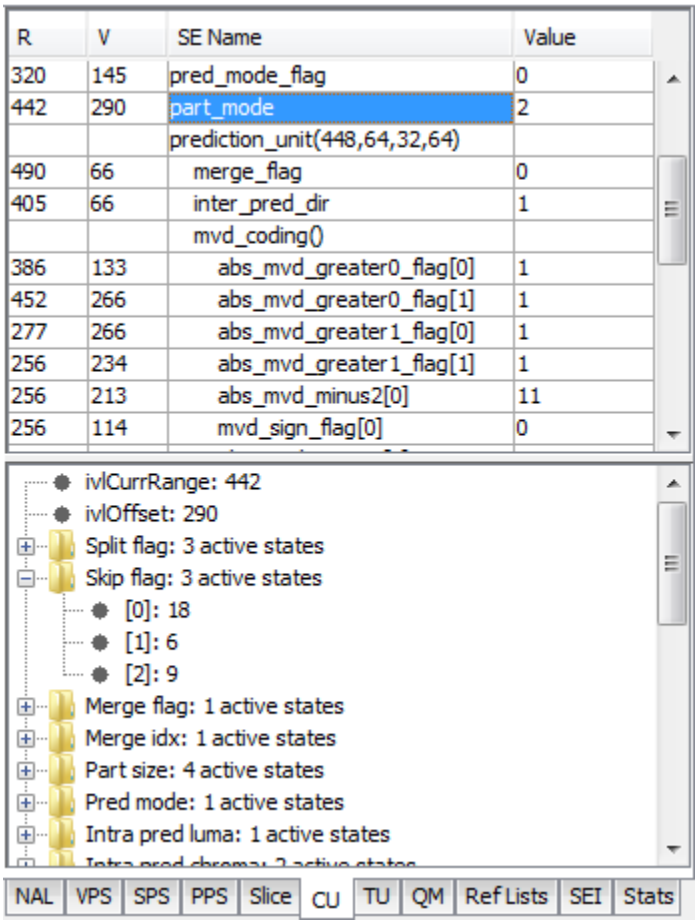
At the bottom of the NAL tab is a button called "Extract selected NAL units". Clicking this will write all selected NAL units out to a file.

3.1.3.2 VPS, SPS, PPS, Slice, SEI



These tabs contain a list of every syntax element, in decode order, of the VPS, SPS, PPS, slice segment header and SEI messages that apply to the currently selected CU in the Main Panel. Syntax elements that come from a sub-function call are indented accordingly, and the function call itself appears in the list as a syntax element with no value. Any part of the list can be selected and copied for pasting in other programs.

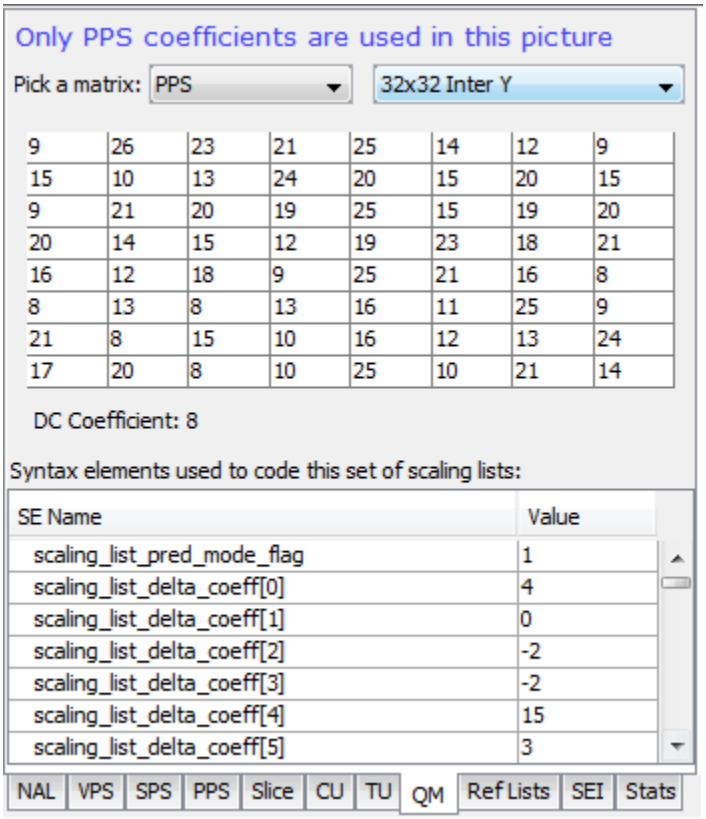
3.1.3.3 CU, TU



These tabs contain a list of the CABAC syntax elements decoded, in decode order, of the currently selected CU or TU in the Main Panel. Note that only the syntax elements pertaining to the currently selected CU/TU are shown, not any of the syntax elements in a higher or lower nodes in the quadtree. So if a CU is selected that is split into 4 smaller CUs, only the split flag will be shown, not the syntax elements of CUs further down the quadtree.
The left side of the list contains two columns, R and V, which denote the state of the CABAC engine's Range (ivlCurrRange) and Value (ivlOffset) prior to the syntax element decoding process. Any part of the list can be selected and copied for pasting in other programs.

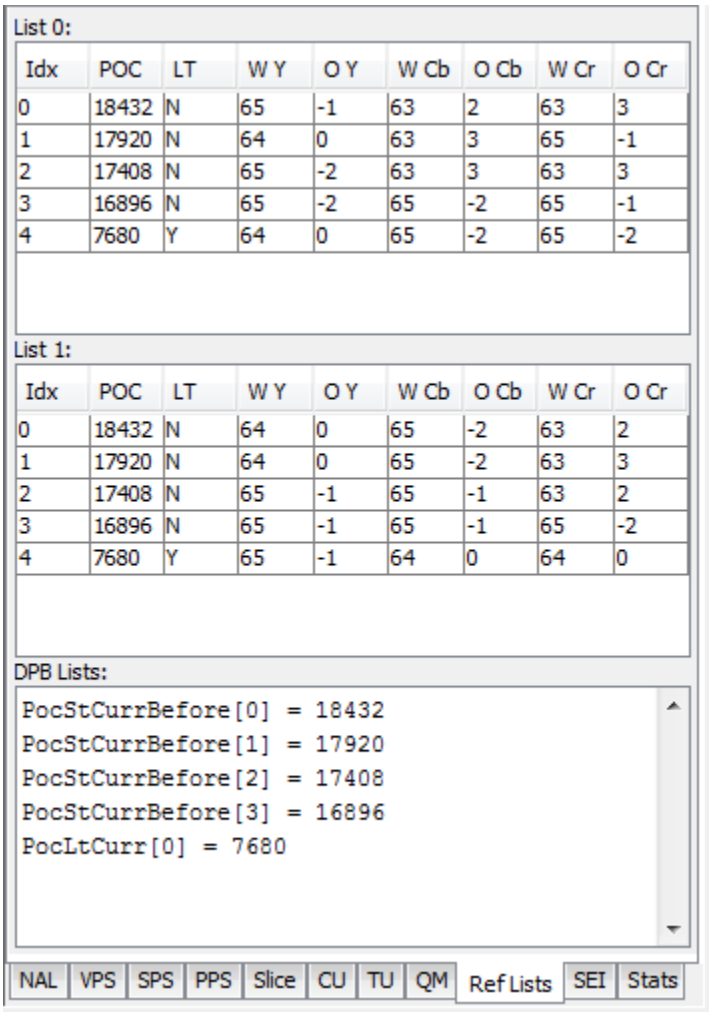
Clicking a syntax element will update the tree in the lower half of this panel. This tree displays the values of all CABAC state variables prior to the decoding of the selected syntax element.

3.1.3.4 QM



This tab displays the scaling lists, or quantizer matrices used by the current picture. Scaling lists can be present in the SPS, PPS, or both. PPS scaling lists take precedence over SPS when present. The top of the panel shows in blue text how the various scaling lists are used in this picture. Below that, one of the 20 scaling list from either the SPS or PPS can be chosen for inspection, and the chosen matrix is displayed in the grid. Matrices for 16x16 and larger TUs have a separate DC coefficient, which is displayed below the grid. Also shown in this panel is the list of syntax elements that is used to code the scaling list set, and is identical to the syntax in the corresponding SPS or PPS tab.

3.1.3.5 Ref Lists

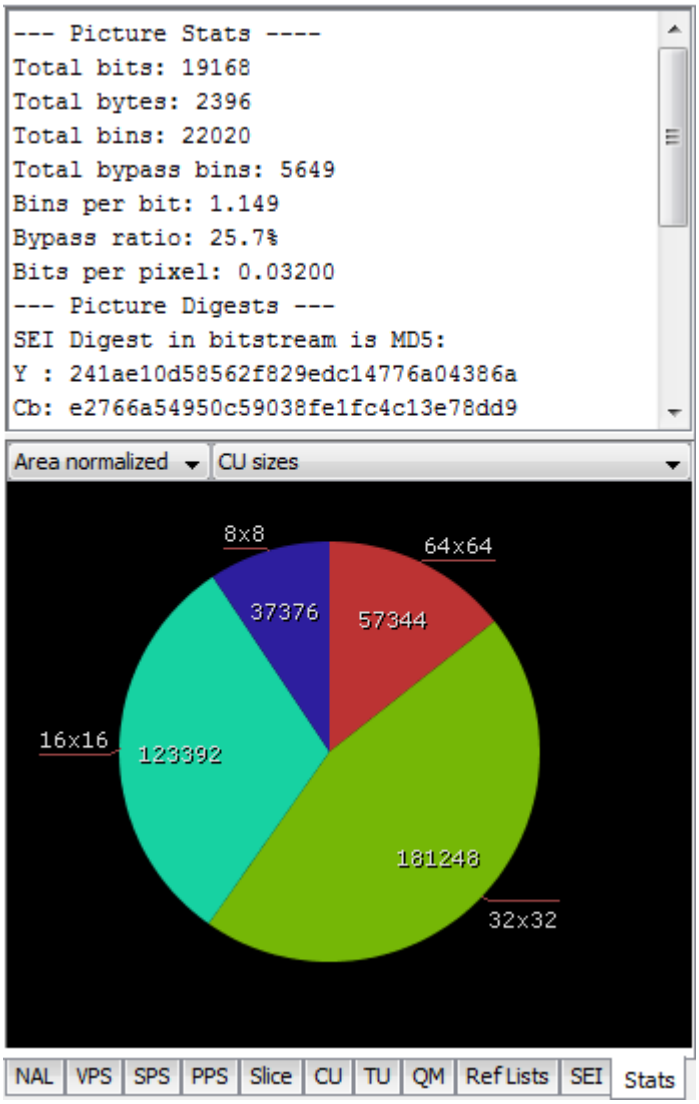


This tab displays the details of the two reference lists L0 and L1, as well as the Reference Picture Set arrays used to construct the L0 and L1 lists. The L0 and L1 lists have the following columns:

- Idx: The index associated with the reference picture
- POC: The Picture Order Count of the reference picture
- LT: A Yes/No flag indicating if the reference picture is a long-term picture or not
- W Y: The derived luma weight used in the weighted prediction process of the reference picture
- Y: The derived luma offset used in the weighted prediction process of the reference picture
- W Cb: The derived chroma Cb weight used in the weighted prediction process of the reference picture
- Cb: The derived chroma Cb offset used in the weighted prediction process of the reference picture
- W Cr: The derived chroma Cr weight used in the weighted prediction process of the reference picture
- Cr: The derived chroma Cr offset used in the weighted prediction process of the reference picture

The third panel contains in textual form the contents of the Reference Picture Set arrays from section 8.3.2 in the HEVC specification.

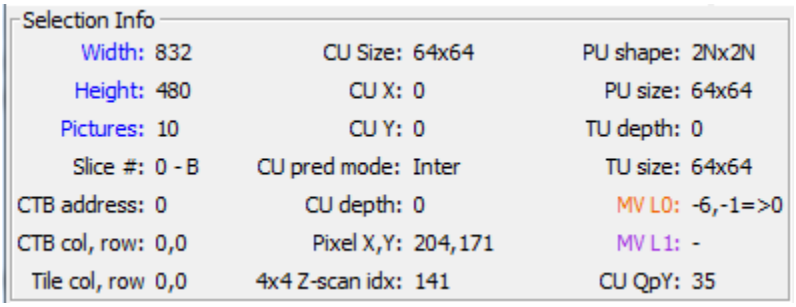
3.1.3.6 Stats



This tab displays various statistics extracted from the current picture. The top half shows some picture size and compression stats as well as the image digest information. The bottom half displays pie charts for a number of metrics. Each pie chart can be drawn normalized or un-normalized. Normalized data is weighted by area or compressed bits. For example there may be a much lower number of 64x64 CUs than 8x8 CUs in the picture (smaller un-normalized pie wedge), but they could still make up the majority of the picture area, making the 64x64 pie wedge large. Normalized numbers in the pie chart are in units of pixels or bits, and un-normalized numbers are raw counts.

The pie chart can be moved by dragging the mouse, and zoomed with the mouse scroll wheel.

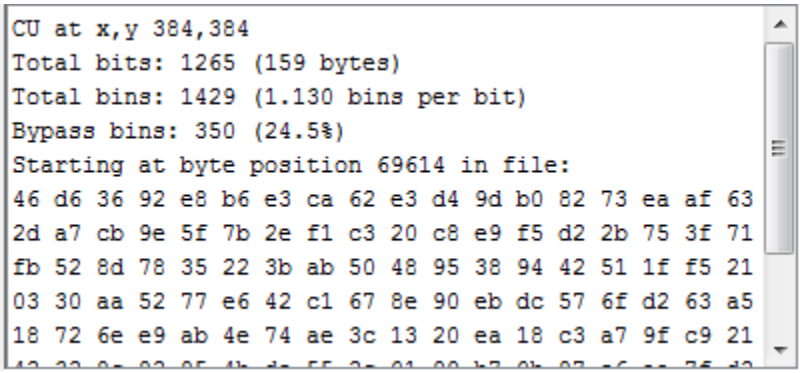
3.1.4 Selection Info



This panel shows a few details about the current selection at a glance. All values are decimal:

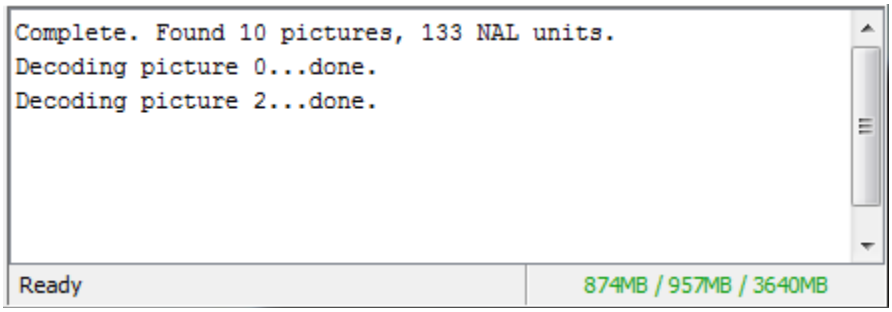
- Picture height & width
- Number of pictures in the bitstream
- Slice segment number of the slice segment that contains the selected CU, and the slice type.
- CTB address and row/col position
- Tile row/col position
- Size of the selected CU. Note it applies to the selection at any point in the quadtree, not necessarily just the leaf nodes.
- X/Y position of the selected CU in pixels
- CU prediction mode. Intra, Inter or Skip
- Depth in the quadtree of the selected CU
- Pixel X/Y position in the picture of the mouse cursor
- Z-order index at 4x4 granularity in the CTB at the mouse cursor location
- PU size and shape. Note that this will be equal to the size of the leaf CU in any mode other than Prediction.
- TU size and depth. Note that this will be equal to the size of the leaf CU (top of TU tree) in any mode other than Residuals.
- The L0 and L1 motion vectors, if any, of the selected PU. The MVs are shown with an arrow => followed by the reference index.
- The Luma QP value associated with the selected CU.

3.1.5 Raw Bytes



The panel in the bottom center of the UI displays the raw bytes that were used to code the selected CU, or NAL unit. A few details about the selection are included as well.

3.1.6 Messages

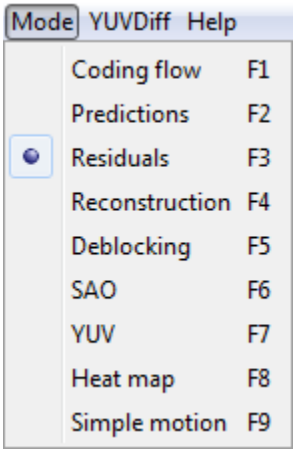


This panel in the lower right of the UI displays messages about the decoding process, and shows progress of any actions that may take a while to complete. If a decoded picture's digest does not match the SEI digest (when present), a warning will be displayed here.

The 3 numbers in the lower right of this panel represent the memory state of JVM. From left to right the 3 numbers are:

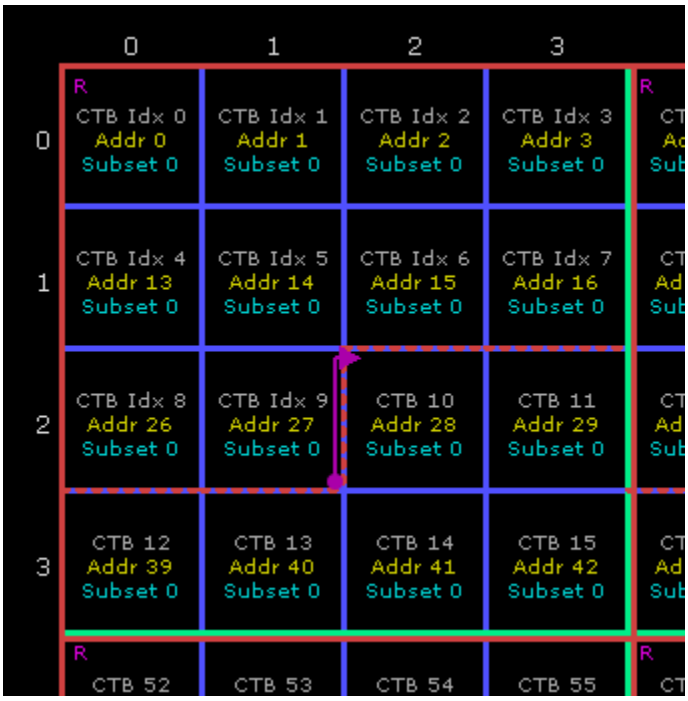
- Amount of memory used
- Total memory claimed by JVM
- Maximum amount the JVM might claim

3.2 Modes



With an HEVC bitstream loaded, Intel Video Pro Analyzer can be put into one of 9 modes using either the F1-F9 keys, or using the Mode menu. The mode selection affects only what is displayed in the Main Panel. Most modes can also show details of the current selection (CU/PU/TU). This can be toggled with the right mouse button or the main panel's button strip.

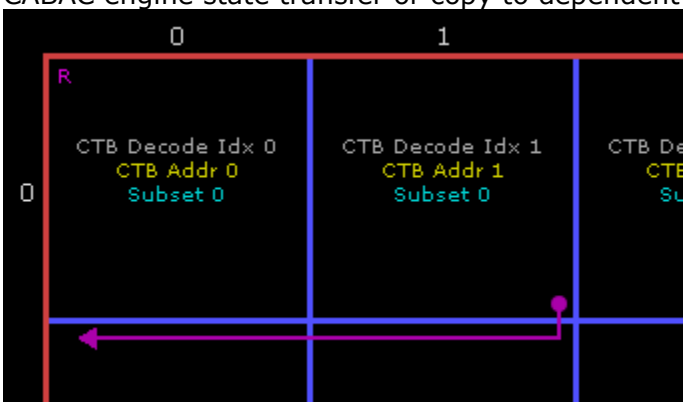
3.2.1 Coding Flow



The coding flow mode gives a visual overview of the ordering of CTBs in the stream, and some information of the decoding process. The blue grid shows the boundaries of the CTBs in the picture. Each CTB contains 3 values:

- The decode index of the CTB, showing the order of decode in the picture.
- The CTB address, which is simply the raster scan index of the CTB.
- The substream that the CTB belongs to. This number will only be greater than 0 if tiles or wavefront tools are employed in the picture.

Operations on the CABAC engine state are displayed as well. A small purple R at the top left of a CTB indicates the CABAC engine is reset before decoding that CTB. A small F indicates a CABAC flush at the end of a substream that is not the end of a slice. Purple arrows indicate CABAC engine state transfer or copy to dependent slices or wavefront rows:



3.2.2 Predictions

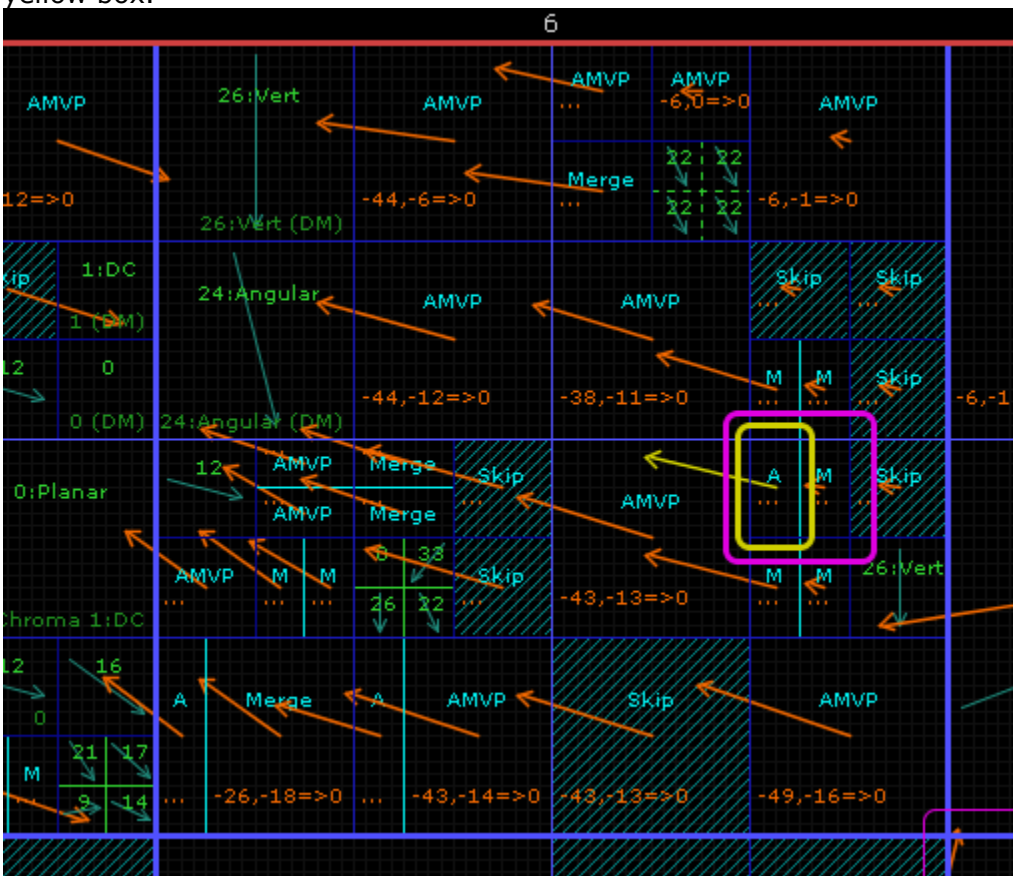
This mode shows the PUs in each CU. CU quad tree splits are indicated with blue lines, which get darker as the split depth increases. When a CU split is implied by HEVC and not directly coded in the bitstream, the split lines are dashed.

Intra modes are indicated with green colors, and directional modes also show an arrow indicating the prediction direction. In the lower right corner of an intra PU the chroma mode is indicated in a darker green. When the prediction shape is not the same as the CU size, the prediction unit shapes are shown with green lines for intra blocks, cyan for inter blocks. Dashed green lines in intra blocks means the PU was split along with the transform tree.

Inter PUs are indicated with cyan colors showing the PU splits and mode. Additionally, skipped blocks are shaded with a lined texture. An inter PU is Skip, AMVP or Merge mode. The L0 motion vectors are drawn with an orange color, the L1 motion vectors (B slices only) are drawn with a purple color. The MV value is shown in the lower left corner along with the reference index.

Clicking a PU will select it, and the syntax elements used to code it and the CU it belongs to are displayed in the CU tab on the left. Clicking repeatedly on the same PU will cycle through the CU hierarchy, showing the parent and child relationship.

Below is shown Prediction mode on a zoomed-in selection along the top edge of a picture. The selected CU is surrounded with a pink box while the selected PU is surrounded with a yellow box.



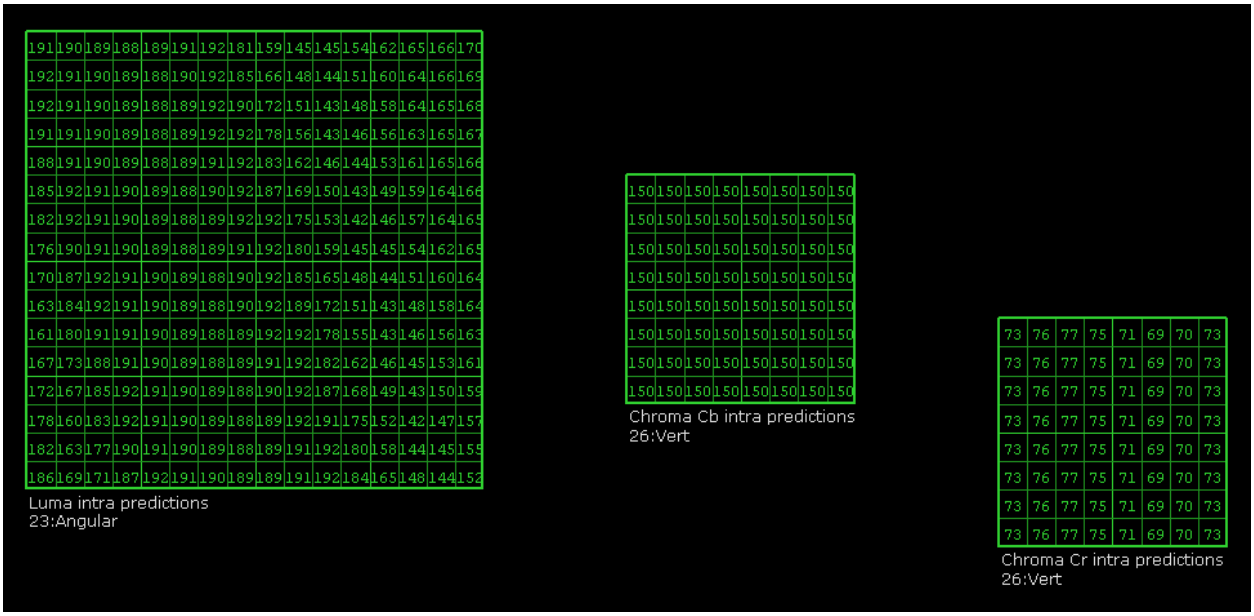
3.2.2.1 Prediction Detail Mode

In Prediction mode, the sample values of a particular PU can be viewed in detail.

Intra PUs are displayed in green, and the left, upper-left and above prediction arrays are drawn adjacent. Luma blocks show 3 versions of the predictions arrays. From furthest out to inner, they represent the arrays in 3 steps:

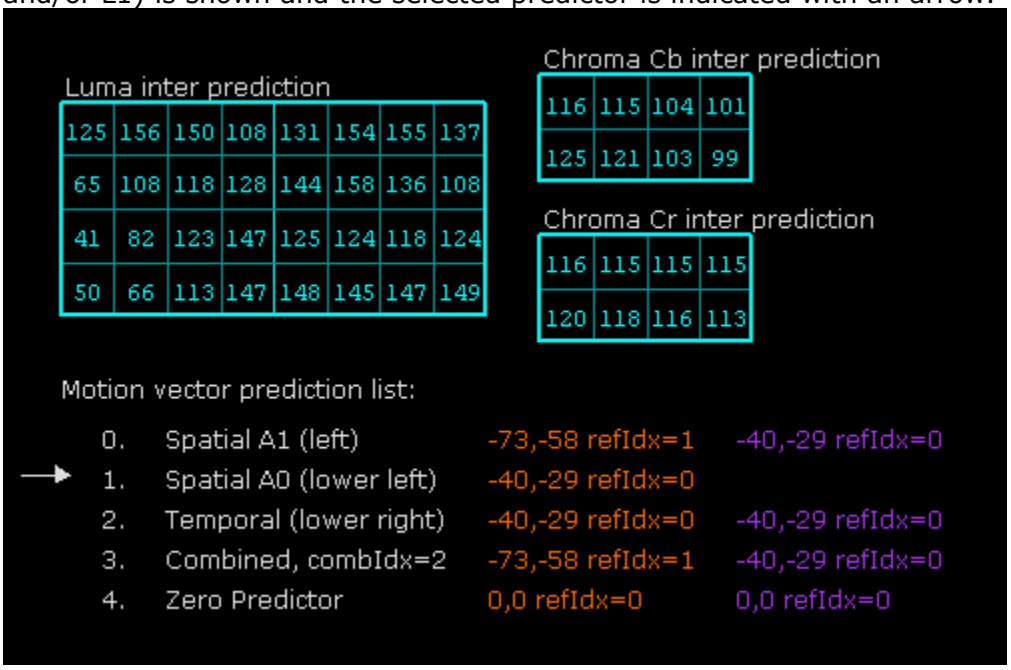
1. Initial neighboring samples. Unavailable samples are given with a red "X".
2. Samples after reference sample substitution process (section 8.4.4.2.2 from the HEVC spec).
3. Samples after filtering process (section 8.4.4.2.3 from the HEVC spec).

Chroma blocks only show the first two steps, since reference sample filtering is not applied to chroma intra prediction.

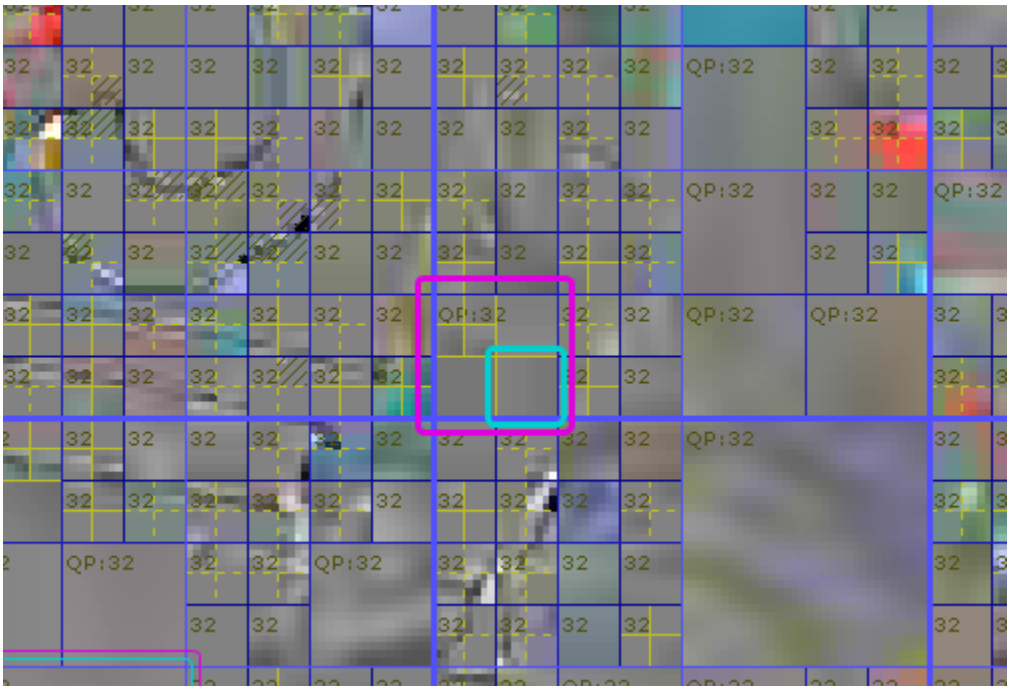


Inter PUs are displayed in cyan (turquoise). Underneath the prediction array values the motion vector predictor lists are displayed. In merge mode, the merge list is shown with the L0 predictor in orange and the L1 predictor in purple. An arrow indicates the merge candidate that is chosen by the bitstream (merge_idx).

When the inter PU does not code motion vectors in merge mode, it is considered AMVP mode (Advance Motion Vector Prediction). The 2-entry list for each motion vector (L0 and/or L1) is shown and the selected predictor is indicated with an arrow.



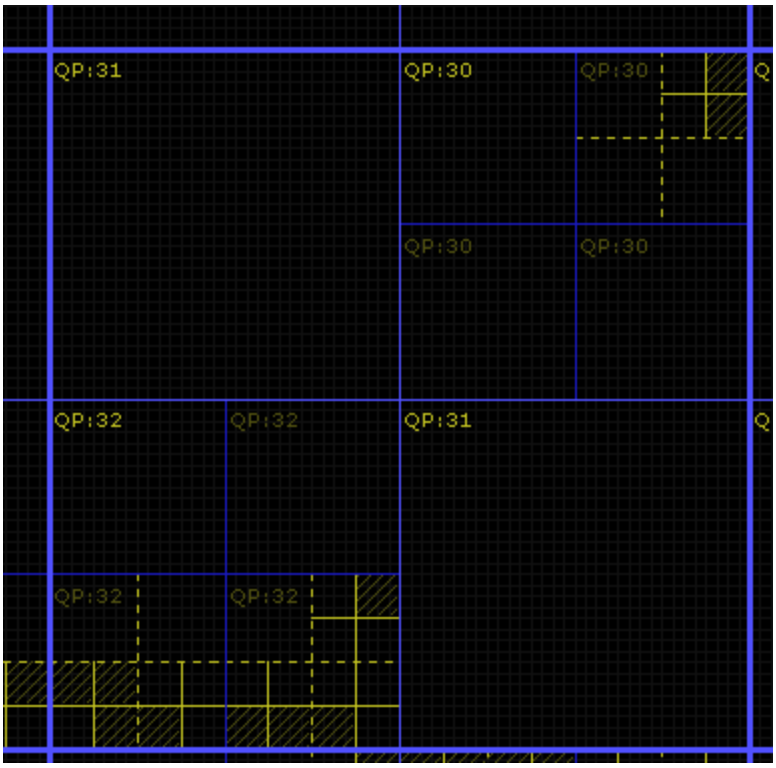
3.2.3 Residuals

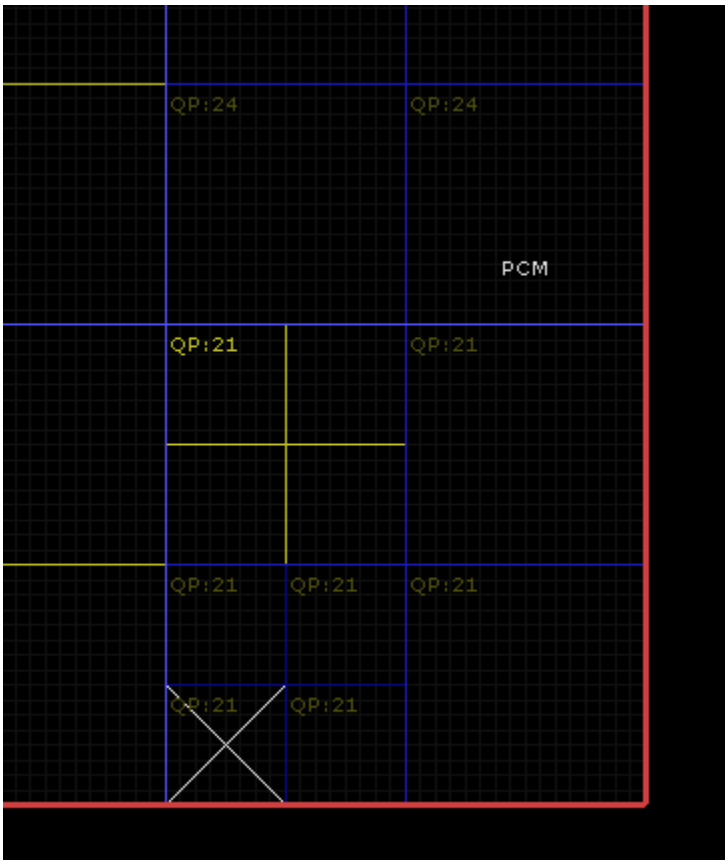


In Residual mode accompanying residual signal of the picture can be seen. As in Predictions mode, CU boundaries are shown with blue lines. The transform splits, when present, are indicated with yellow lines. A dashed yellow line indicates that the TU split was implied. QP values of each CU are shown in the upper-left corner. When this value is bright yellow, it means that a delta-QP was coded in the bitstream during the decode of that particular CU. Otherwise the number is dark. When DQP is enabled by the PPS, this helps visualize the DQP coding depth (diff_cu_qp_delta_depth). TUs that have transform_skip_flag set are shaded. CUs that are PCM coded are marked with "PCM" in white, and CUs that have cu_transquant_bypass_flag set have a white X through them.

If the "Pic" button is turned on, the raw residual signal is shown in image form. Residual values of 0 result in flat grey, negative values are darker and positive values are brighter.

Clicking on TUs causes that TU to be selected with a blue surrounding box and its syntax elements (when present) to be displayed in the TU tab of the left panel. Clicking a TU repeatedly causes the selection to move up and cycle though the TU, then CU quadtree hierarchy. Note that when the selected TU is a 4x4, the chroma coefficients are coded one level up in the TU hierarchy since HEVC doesn't specify a 2x2 chroma TU. So to view chroma TU syntax when the luma TU is 4x4, click the luma TU twice to move up to the 8x8 level where the chroma syntax is decoded.

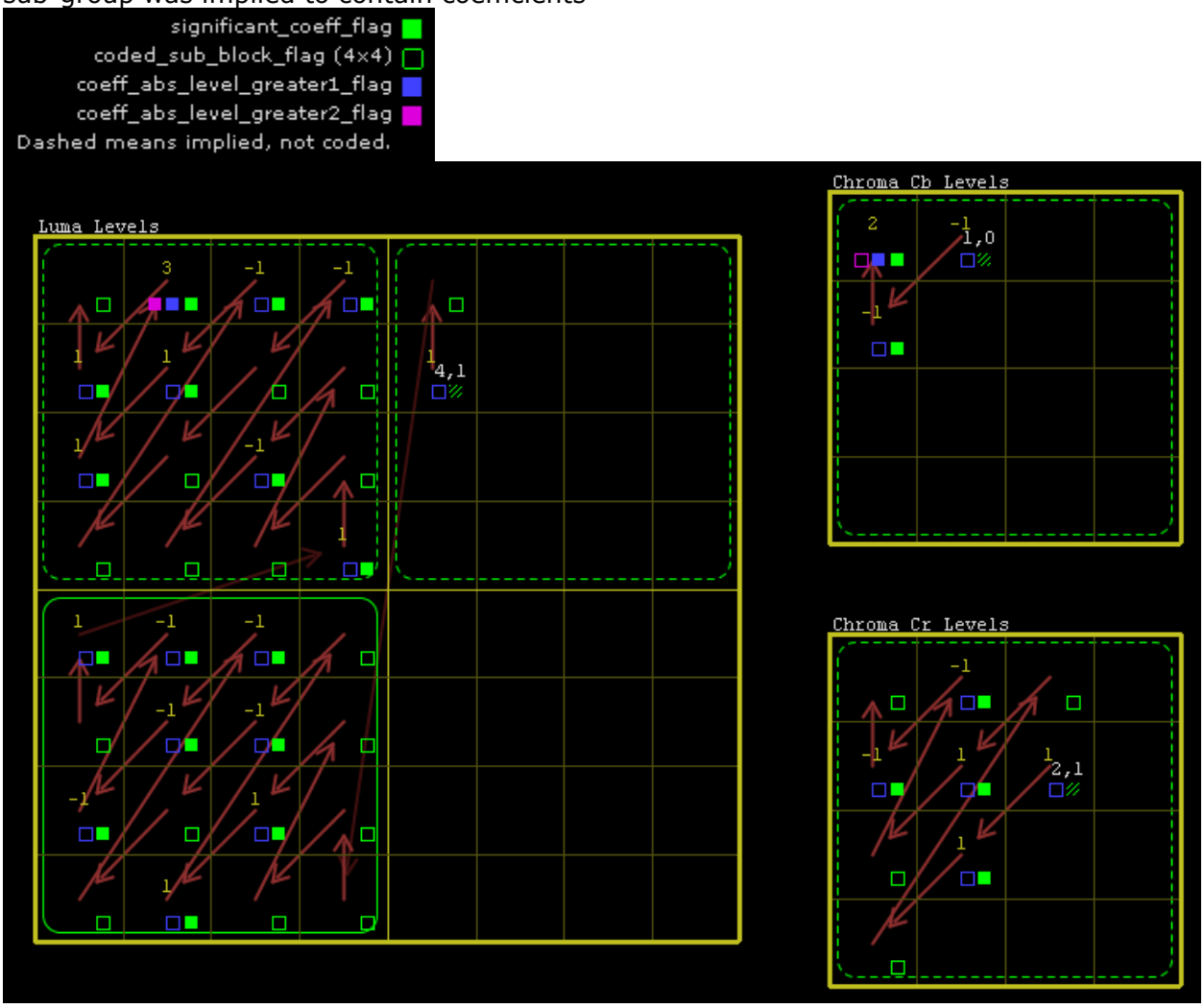

QP:31
QP:30
QP:30
QP:30
QP:30
QP:32
QP:32
QP:31
QP:32
QP:32


QP:24
QP:24
PCM
QP:21
QP:21
QP:21
QP:21
QP:21
QP:21
QP:21

3.2.3.1 Residual Detail Mode

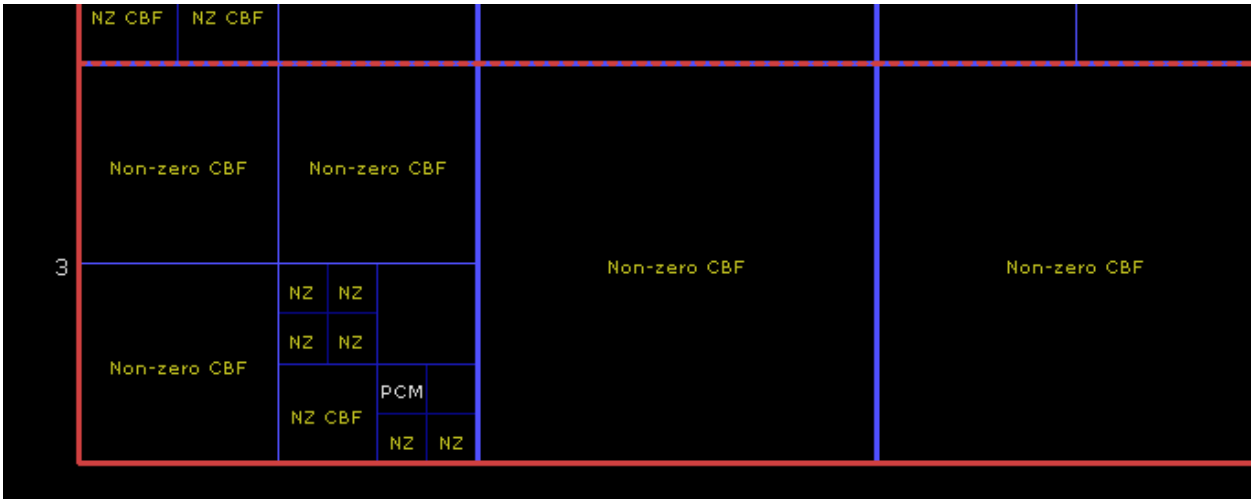
To view the full details of a particular TU or group of TUs, make the TU selection and enter detail mode by right clicking or using the detail mode button at the bottom of the main panel. In this mode the selected TU structure is drawn three times, arranged in a column from top to bottom, showing the 3 major steps in recovering the residual signal:

Coefficient decode. This diagram shows the scan order of the coefficients, the coefficient values and small icons indicating the syntax elements associated with coefficient decode. Refer to the legend shown in the top right of the main panel for the exact details. Coefficient groups are outlined with a green box. This box is bright if the coded_sub_block_flag was equal to 1, dark green if 0. A dashed line means the sub-group was implied to contain coefficients



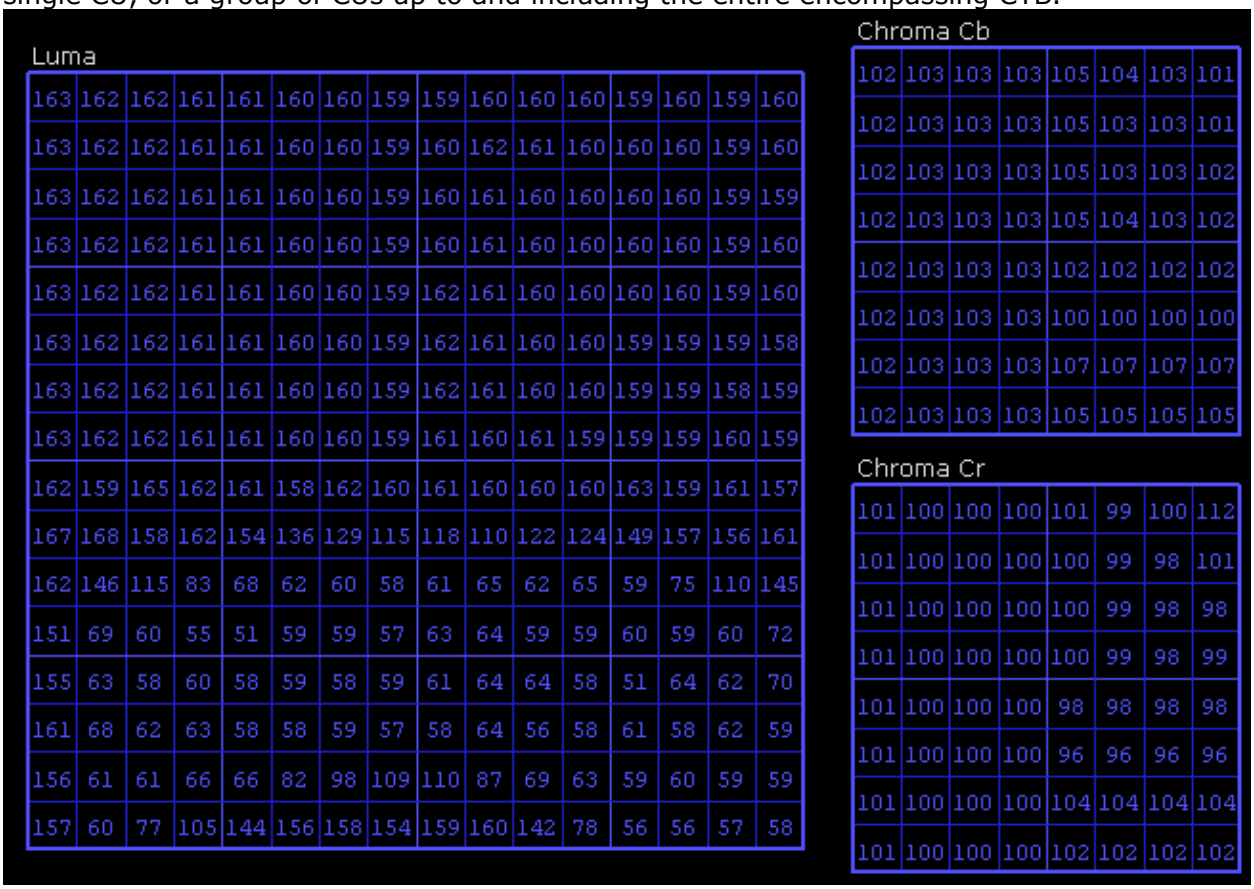
3.2.4 Reconstruction

In Reconstruction mode the reconstructed samples prior to deblocking can be inspected. As with Prediction/Residual mode, CU boundaries are shown with blue lines. CUs that have at least one TU with non-zero coefficients are marked with "Non-zero CBF". PCM-coded CUs are indicated with "PCM" in white.



3.2.4.1 Reconstruction Detail Mode

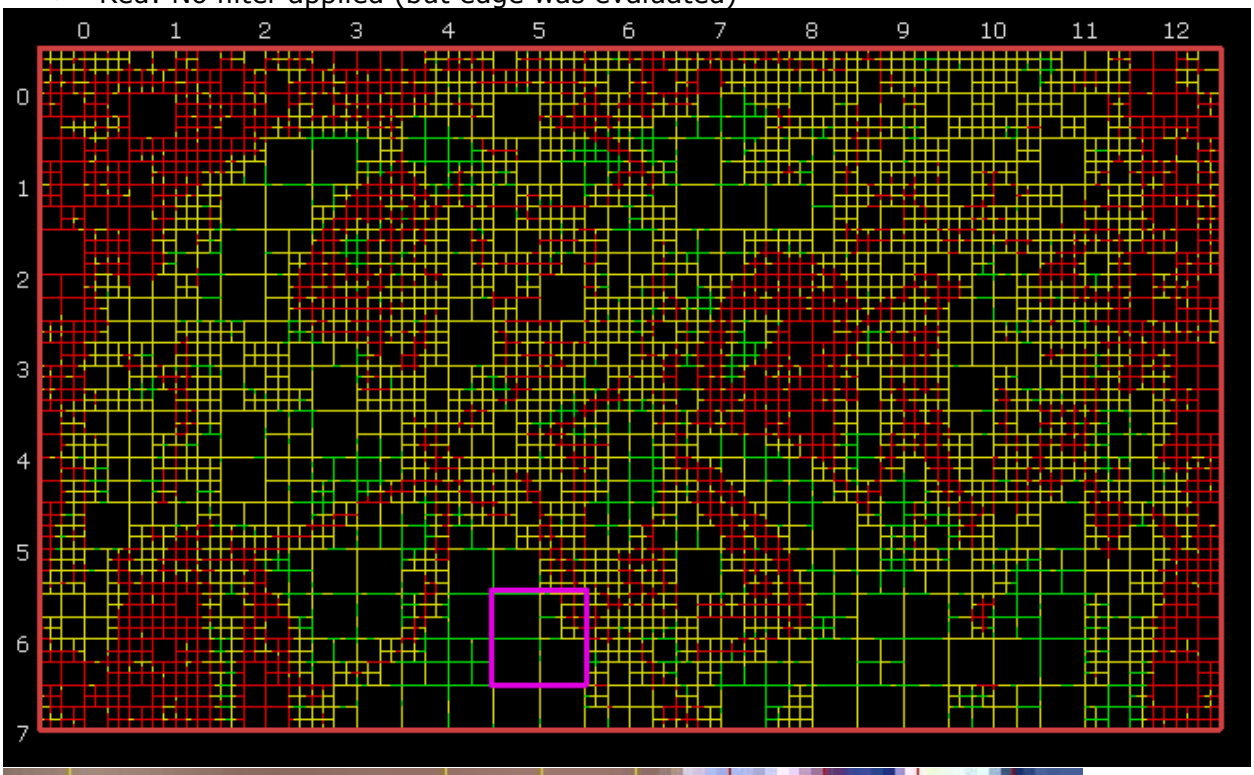
In detail mode, the reconstructed sample values can be inspected. The selection may be a single CU, or a group of CUs up to and including the entire encompassing CTB.

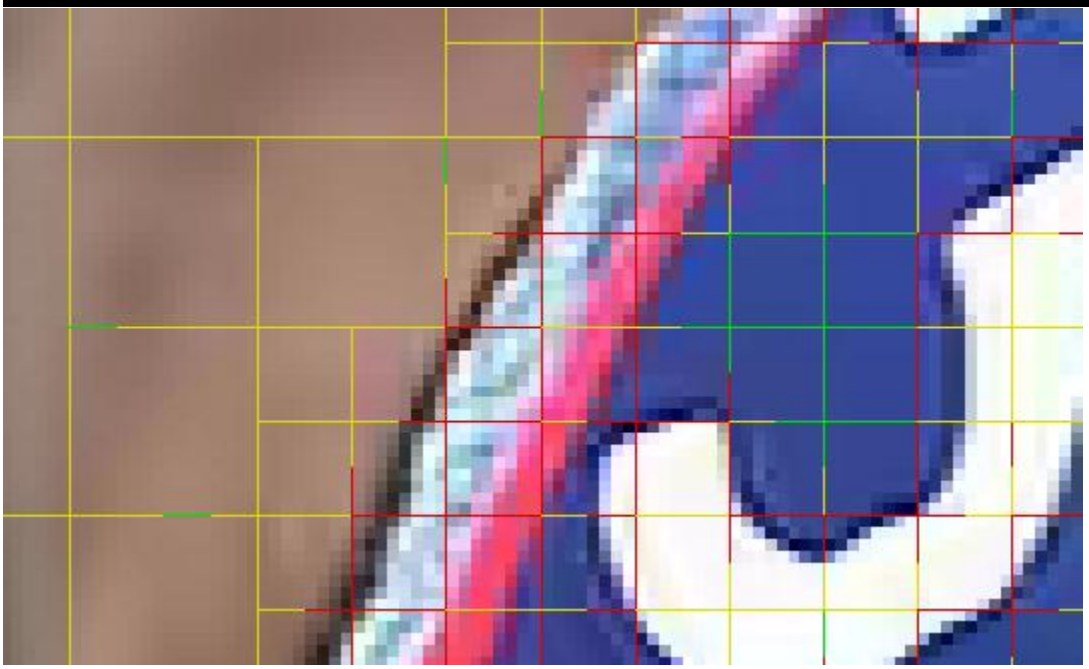


3.2.5 Deblocking

Deblocking mode shows all edges processed by the deblocking filter as described in the HEVC spec section 8.7.2. CU boundaries are not displayed in this mode, however CU selection still functions as in the other modes. Edges shown are for the luma deblocking process and are color coded in the following manner:

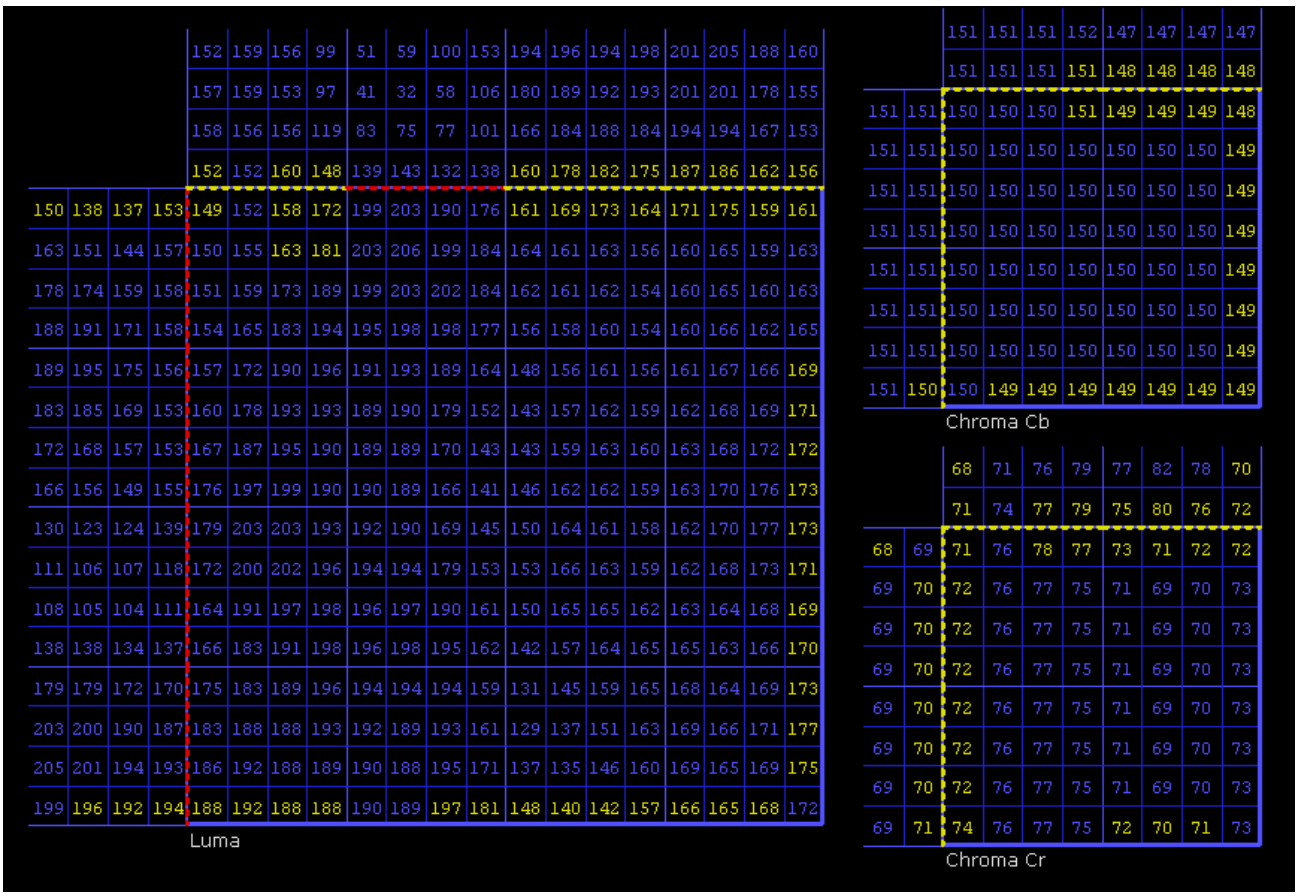
- Green: Strong luma filter applied
- Yellow: Weak luma filter applied
- Red: No filter applied (but edge was evaluated)





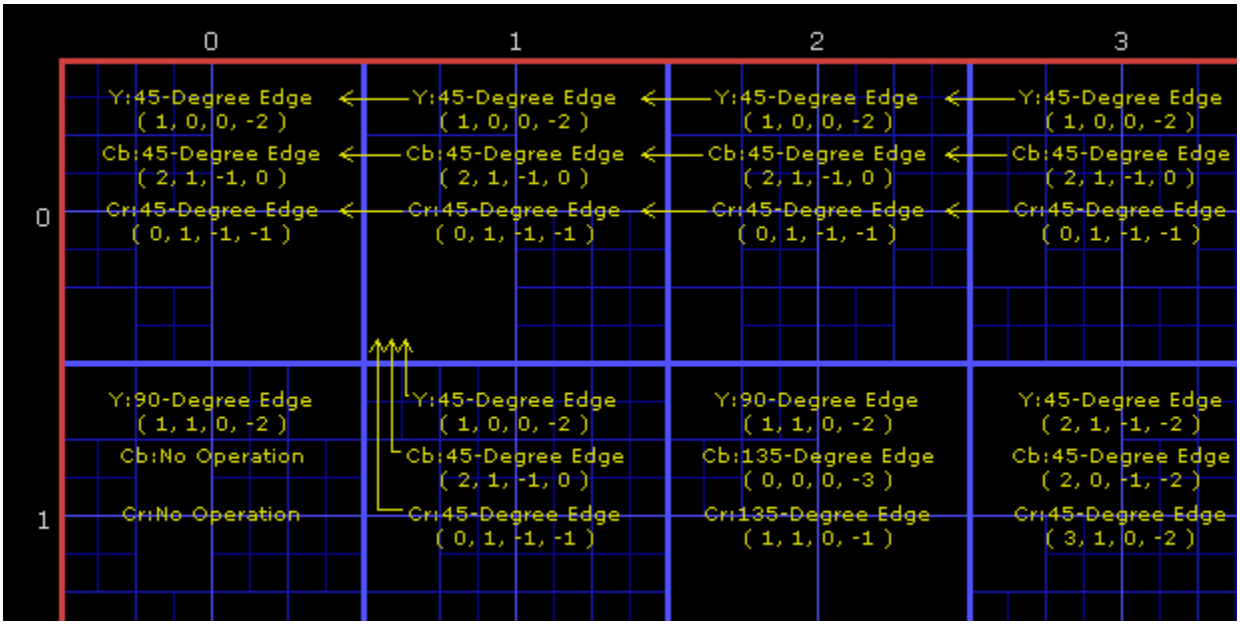
3.2.5.1 Deblocking Detail Mode

The deblocked sample values can be inspected directly in detail mode. As with Reconstruction Details, the inspected area can consist of one or more CUs. Sample values are indicated in yellow if they were modified by the deblocking process. Sample values that cannot be modified due to pcm_flag or cu_transquant_bypass_flag are indicated in red. Filtered edges belonging to the selected CU are shown in thick dashed lines, color coded the same way as in the non-detail mode.



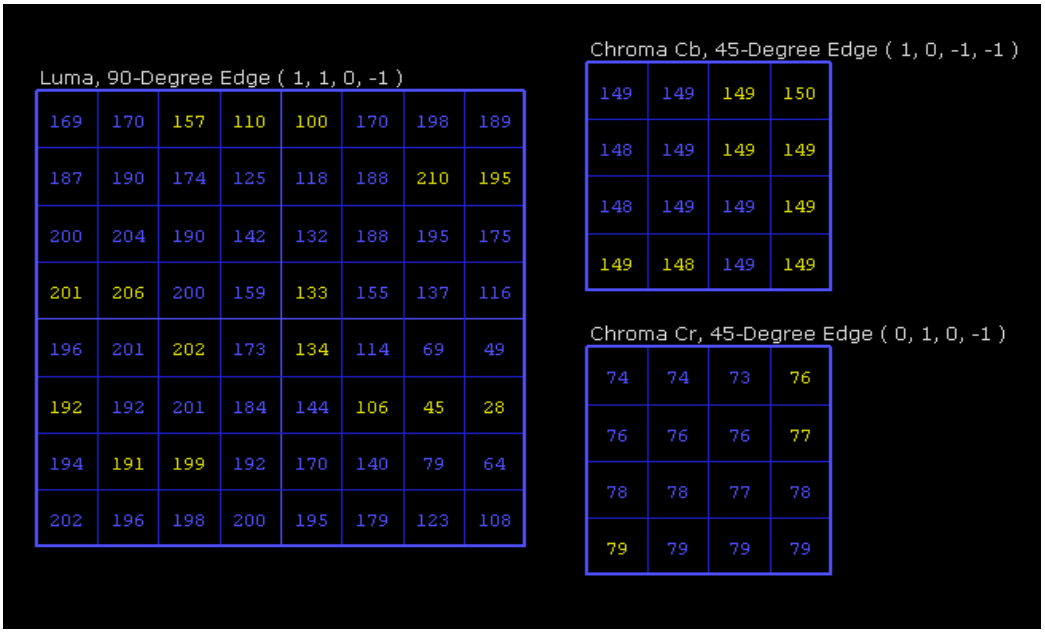
3.2.6 SAO

In SAO mode the SAO filter parameters assigned to each CTB are shown. The mode and associated four offsets for each component are indicated with yellow text, and the merge_up / merge_left flags are indicated with an arrow. The CU quadtree boundaries are shown here as well.

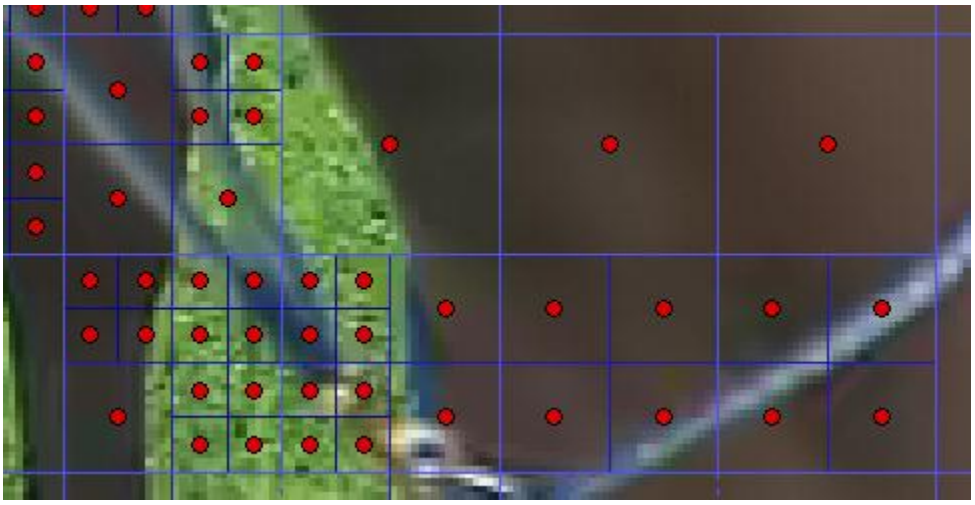


3.2.6.1.1 SAO Detail Mode

Entering detail mode on a selected CU will show all samples after being processed by the SAO filter. Samples values that were actually modified by the SAO operation are highlighted in yellow. Samples that cannot be modified by the filter are indicated in red.

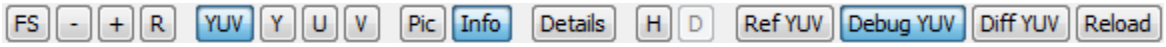


3.2.7 YUV



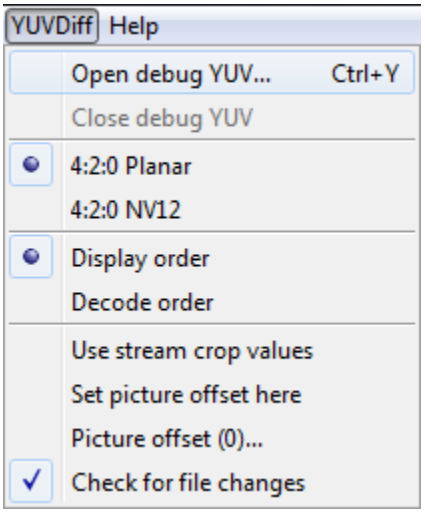
YUV mode allows inspection of the final decoded sample values, without additional overlay data. In this mode, the YUVDiff feature is exposed. This feature allows for comparison of an external decoded YUV file with the decoded bitstream. Supported formats are 4:2:0 planar and NV12. Also, YUV files that are zipped or gzipped may be opened directly without the need to decompress separately. If a zip file contains more than one YUV file, only the first one is used. When a YUV file is loaded, any mismatches will be indicated with a red dot in the CU containing the mismatch. This allows the user to quickly identify the nature of the mismatch which can assist with debug.

When a file is opened, four additional buttons become visible on the lower left of the main window:



- Ref YUV: The original, expected YUV image.
- Debug YUV: The loaded debug YUV file.
- Diff YUV: The delta image. Like in Residual mode, areas with zero delta (i.e. Original and Debug image are identical) are flat gray. Areas where the debug YUV has a lower value are darker, and areas where the debug YUV has a higher value are brighter.
- Reload: Reloads the YUV file, which can be useful as a shortcut to loading via the YUVDiff menu.

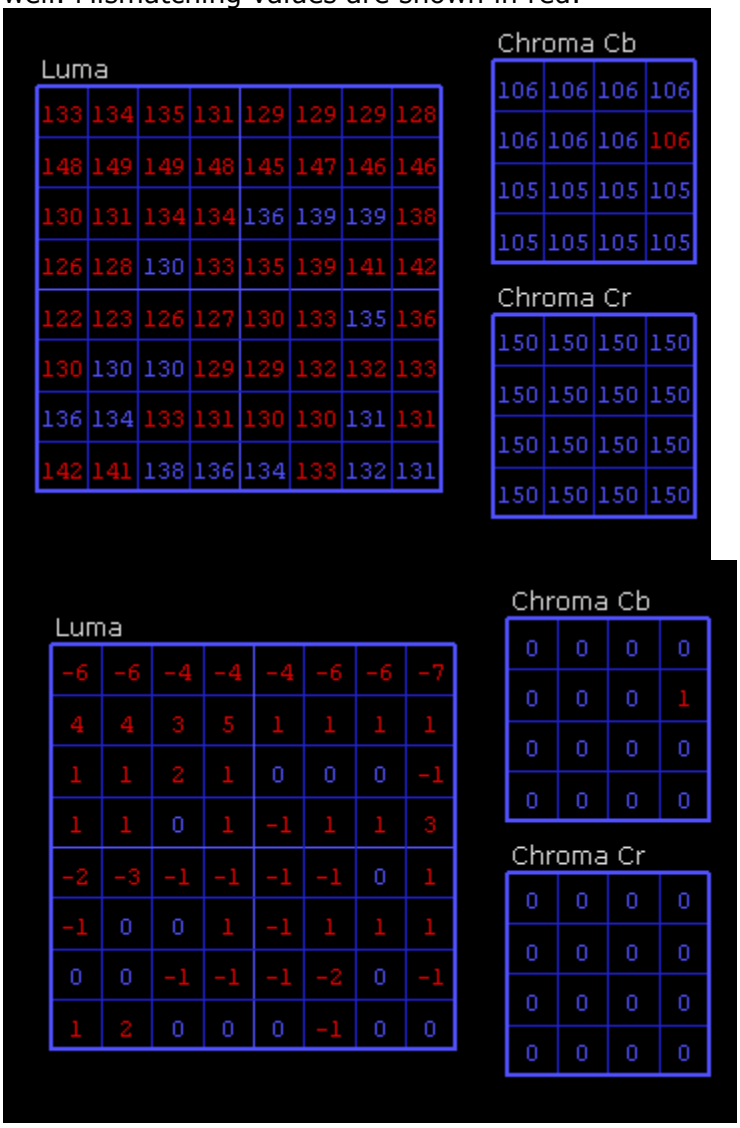
3.2.7.1 YUVDiff Menu



- Open: Brings up a file chooser to open a YUV file
- Close: Closes the currently opened YUV file. Any mismatch indicators will disappear.
- 4:2:0 Planar: Causes Intel Video Pro Analyzer to interpret the loaded YUV file as 4:2:0 Planar.
- 4:2:0 NV12: Causes Intel Video Pro Analyzer to interpret the loaded YUV file as 4:2:0 NV12. This option is chosen automatically when the loaded YUV file has the file extension ".nv12".
- Display order: Causes Intel Video Pro Analyzer to use display order picture numbers when determining how far to seek into the YUV file to extract the image to compare with the current picture.
- Decode order: Causes Intel Video Pro Analyzer to use decode order picture numbers when determining how far to seek into the YUV file to extract the image to compare with the current picture.
- Use stream crop values: When checked, YUV files are assumed to contain samples only within the cropping window as defined by the loaded bitstream. Samples outside this window are not compared, and are assumed 0 when viewing the loaded YUV image directly.
- Set picture offset here: Shortcut for setting the picture offset to the current picture number. See bullet below.
- Picture offset: This brings up a dialog allowing the user to enter the picture number of the first picture in the YUV file. For example if a 100-picture bitstream is loaded but the YUV file only contains pictures 80-99, the user would enter 80 to properly line up the YUV file with the decoded bitstream.
- Check for file changes: When checked, this option causes Intel Video Pro Analyzer to periodically check if the loaded YUV file has changed on disk since it was last loaded. If the file has indeed changed, a dialog pops up offering a chance to reload the YUV file.

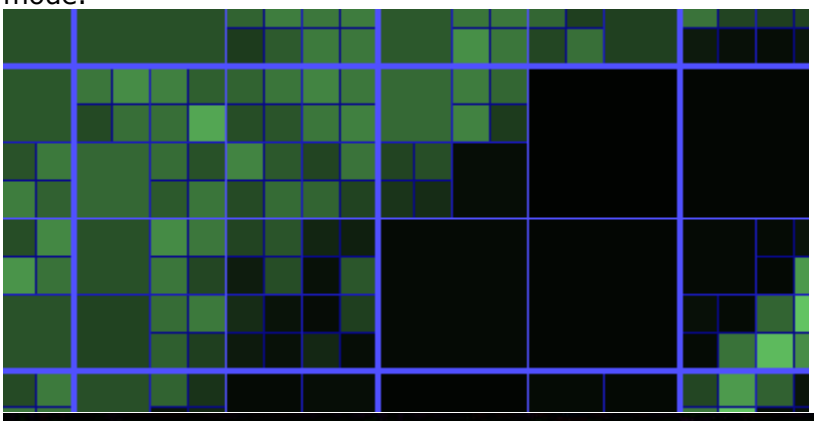
3.2.7.2 YUV Detail Mode

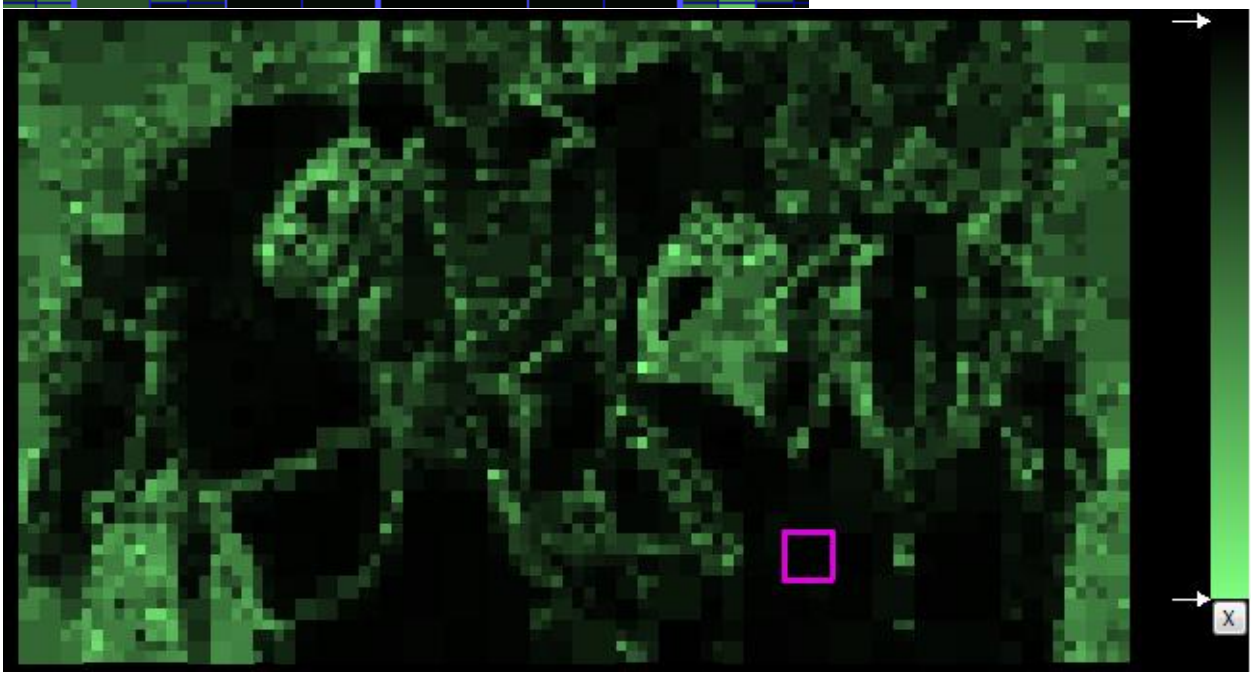
In detail mode the expected YUV values can be inspected. Note that the displayed sample values are the same as those in SAO mode since SAO filtering is the last step in the decode process. When a YUV file is loaded, the debug and delta YUV values can be inspected as well. Mismatching values are shown in red.



3.2.8 Heat Map

Heat map mode shows visually how the compressed bits of the picture are distributed spatially. CUs with more bits per pixel are brighter than CUs with less. By default this mode also shows the CU quadtree boundaries in blue. It may be useful to turn off this overlay using the "Info" button on the lower left of the main panel. Heat Map does not offer a detail mode.





The gradient used to display the heat map may be edited:

- Drag the gradient markers up and down to reposition them.
- Drag a marker away from the gradient to remove it. A red X indicates removal will take place.
- Double-click a marker to change the color.
- Double click the gradient bar to add a new marker.

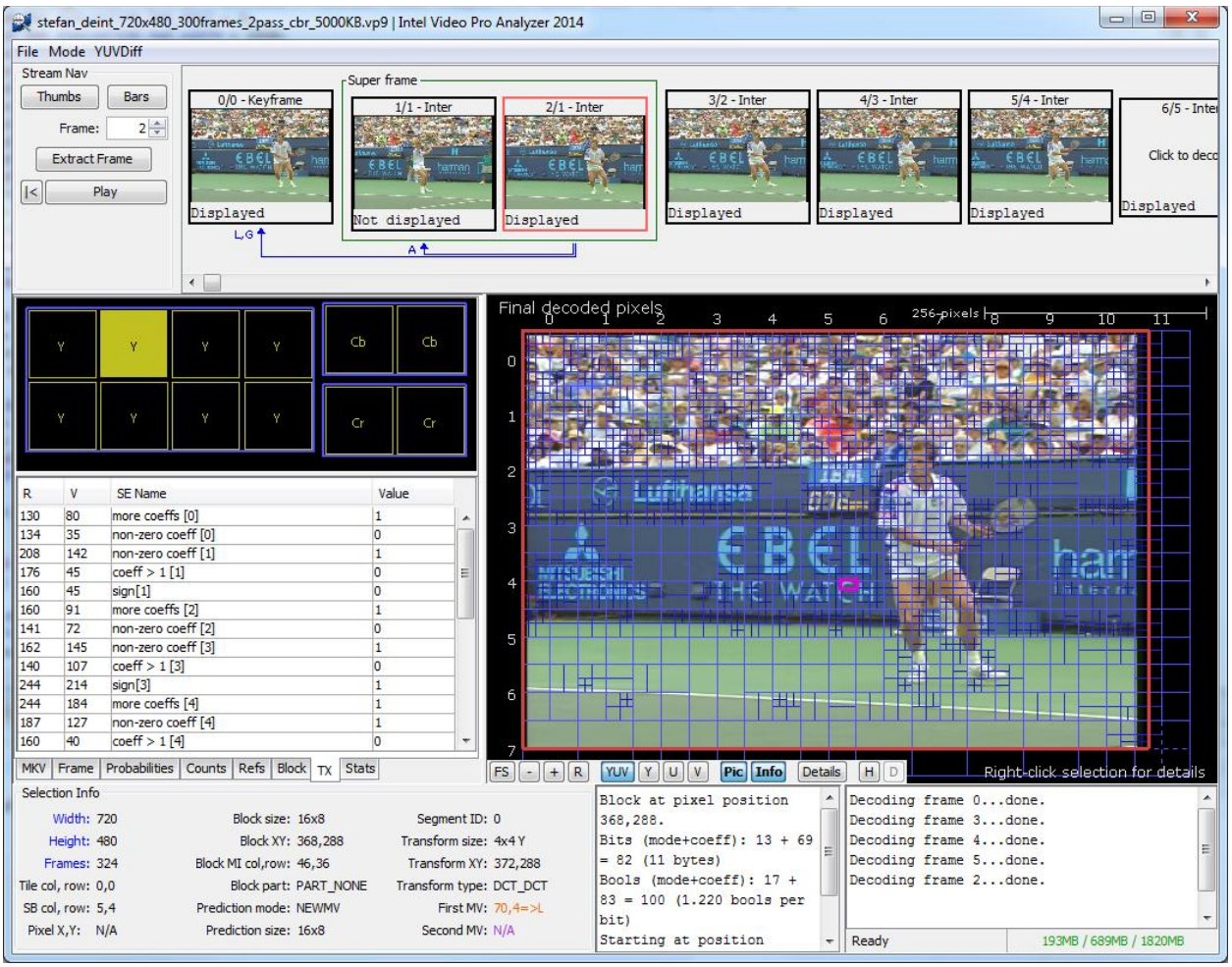
3.2.9 Simple Motion

Simple Motion mode offers a way to quickly view the modes and motion of a picture at a glance. No overlays are present except a colored dot showing information about the PU's mode. Green dots indicate intra PUs, and purple/orange L0/L1 motion vectors for inter PUs are drawn as simple lines. When the motion vector is very small, a dot is drawn instead so that the mode is still easy to identify. When a motion vector does not point to the 0-th index of its respective list, the vector is drawn using a dashed line.

4 VP9*

The following sections describe all available features when loading a VP9* bitstream. The bitstream must be containerized with either the WebM or IVF container format. The publicly available VP9 reference software can output both formats. Raw uncontainerized bitstreams are presently not supported due to the nature of the VP9 standard; there is no way to determine where each frame starts without doing a full decode of the sequence.

Only YUV color space and 4:2:0 chroma subsampling are supported.



4.1 UI Components

The following sections describe the various components in the UI when an VP9* bitstream is loaded.

4.1.1 Top Filmstrip

The top filmstrip is a horizontally scrolling overview of the pictures in the bitstream. The current picture is highlighted with a red border. On top is the frame number in both decode and display order respectively and the picture type as indicated by the frame header. On bottom is indicated if the picture should be displayed or not. Arrows indicate the reference frames.

Clicking on a picture will cause Intel® Video Pro Analyzer to decode it and mark it as the current picture, updating the rest of the UI correspondingly. If the new picture requires other pictures to be available as reference that have not yet been decoded, those will automatically be decoded.

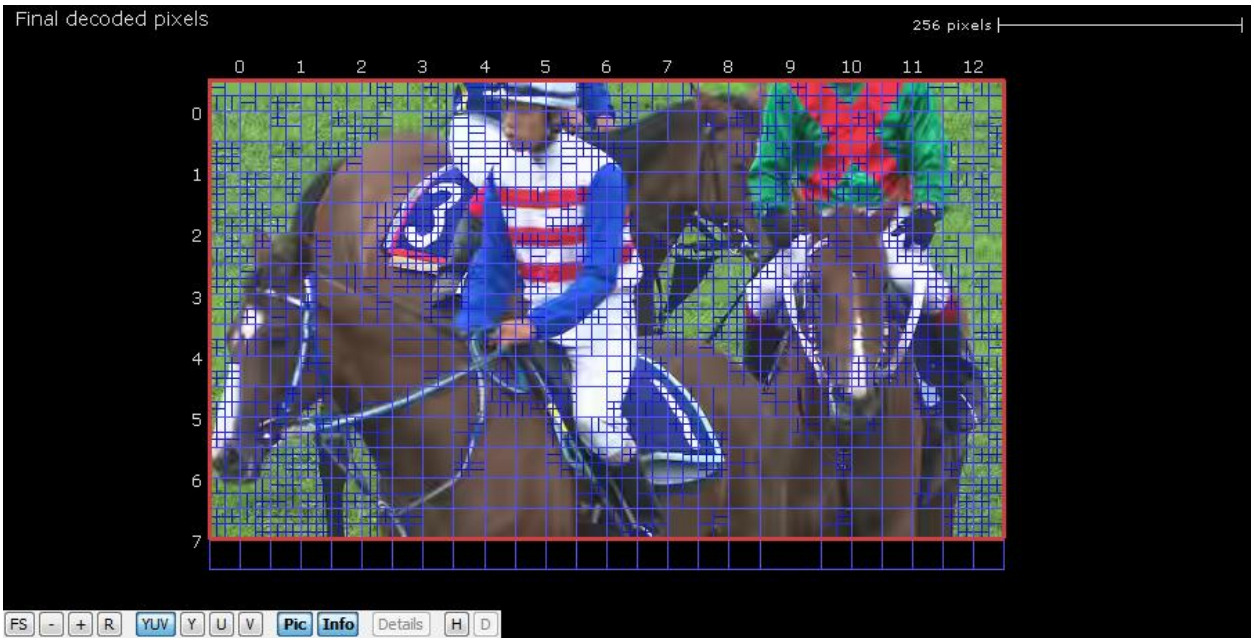
When a frame from the IVF or WebM/MKV container is a super frame (meaning 1 or more non-displayed picture and one displayable), this collection of pictures is surrounded by a box indicating which frames belong to the super frame.

Pictures are decoded on demand since it is impractical to store all details of all pictures in memory, especially for HD sequences. The current picture can also be selected by typing in a picture number in the box on the left of the panel. When using this box, the filmstrip is scrolled such that the newly selected current picture is visible.

On the left of the filmstrip are two buttons that allow toggling the filmstrip between thumbnails and bar graphs.

The button labeled "|<" will rewind, causing the current frame to be set to the first frame in the loaded bitstream. The "Play" button will start playback of loaded bitstream at a reasonable rate, limited by CPU performance.

4.1.2 Main Panel



The main panel displays the selected picture with visual annotations. The type of annotations and associated interactive behavior depends on the current mode, see 4.2 VP9 Modes. Click-dragging moves the picture around, and the mouse wheel zooms in or out about the mouse cursor.

At the top left of the panel, the current mode is displayed. Along the top right the current scale is shown. In all modes, the super-block row and column values are displayed along the left and top border of the picture and tile boundaries are thick green lines. The screenshot above shows an example of a stream that is 13x8 SBs.

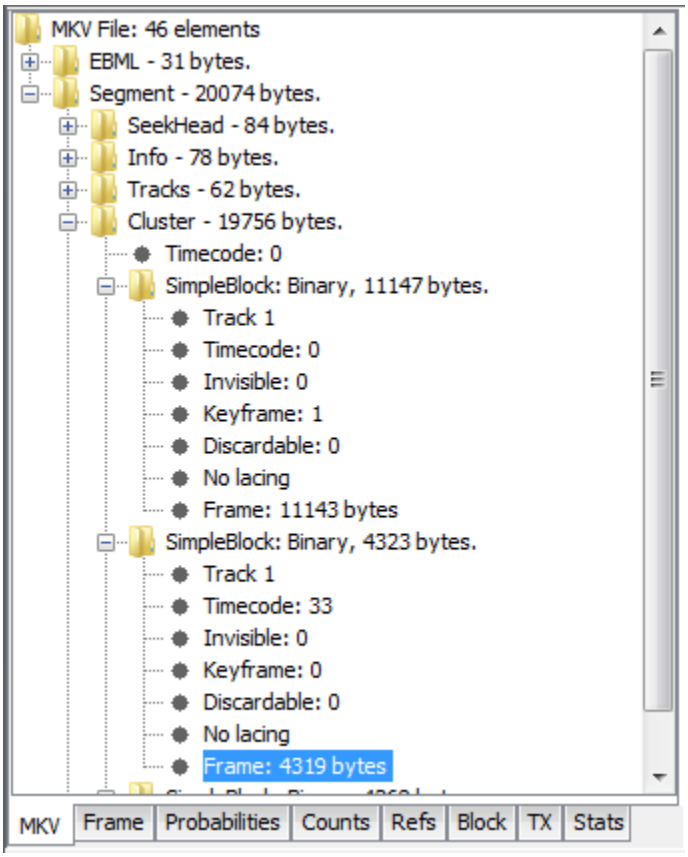
The bottom left has a cluster of buttons:

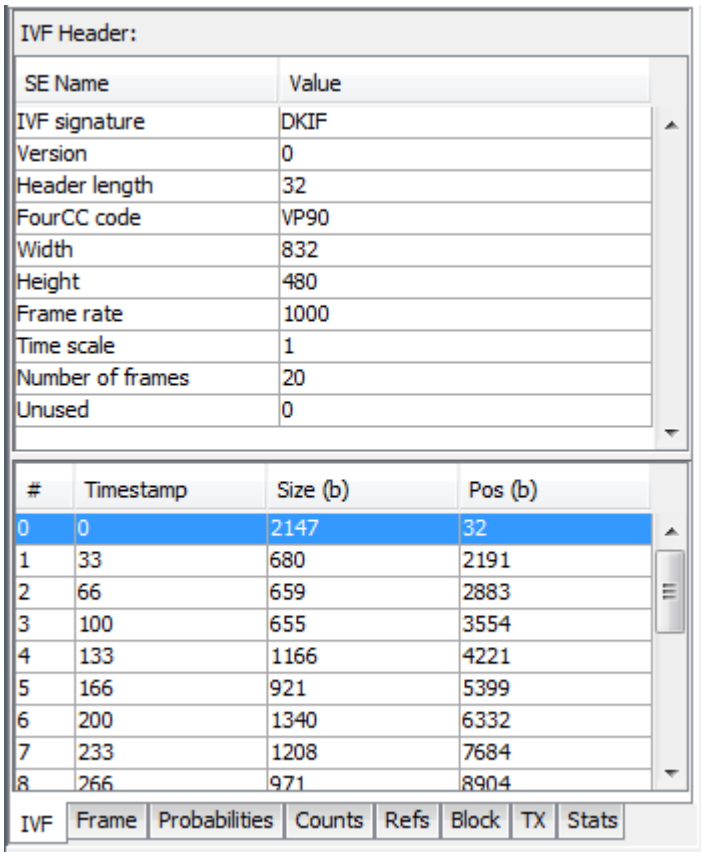
- FS: Toggles full-screen mode, which hides all UI elements except the main panel. Keyboard shortcut is F.
- +/-: Zooms in or out, centered on the center of the panel. Note that zooming with the mousewheel is much easier than using these buttons. Keyboard shortcuts are + and -.
- R: Resets the current zoom to fit the entire picture in the main panel. Keyboard shortcut is R.
- YUV, Y, U, V: Toggles between per-component image, or full-color YUV.
- Pic: Toggles the actual picture underneath the annotations. The picture is mode-dependent. For example in prediction mode, the picture is comprised of the prediciton samples before the residual signal has been added in. It is often helpful to turn off the picture in order to make the annotation easier to read. Keyboard shortcut is P.
- Info: Toggles the annotations on or off. Keyboard shortcut is I.
- H/D: Toggles between hexadecimal and decimal display of the annotations. This applies to the values in the Left Tabs as well. Keyboard shortcuts are H and D.

4.1.3 Left Tabs

The left side of the UI is a resizable panel with several tabs that display information about the current selection in one way or another. The following subsections describe each tab.

4.1.3.1 Container (IVF or MKV)

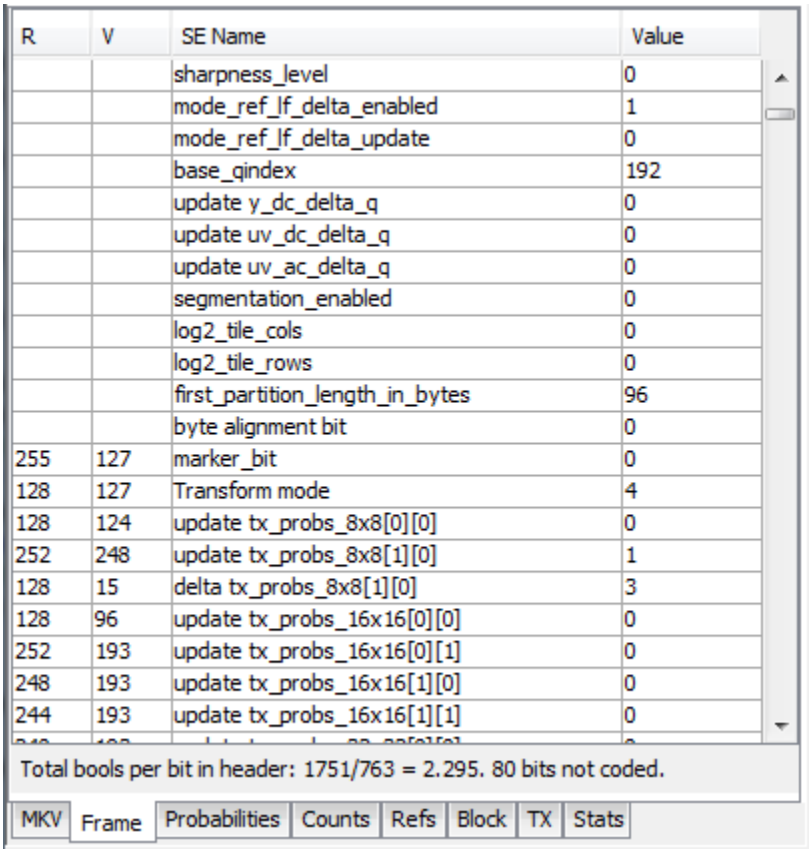




Depending on the container type of the loaded bitstream, either the MKV or IVF panel will be present in the UI.

- VP9 streams using the WebM or Matroska container will cause the UI to display the container's EBML document in tree form. Clicking an element causes the raw bytes coding that element to be displayed in the Raw Bytes panel (bottom center of the UI). When decoding a particular frame, the corresponding element will be highlighted in this panel. Double-clicking a frame in this panel causes that frame to be decoded. Other tracks may be present in the MKV file (such as audio), but they are not processed. The raw bytes can still be viewed though.
- Streams containerized with the IVF format causes the UI to display the IVF information in this left-most tab. Selecting a frame causes the Raw Bytes panel to show the raw bytes of that frame, including the IVF frame header. Double clicking a frame will causes that frame to be decoded.

4.1.3.2 Frame



R	V	SE Name	Value
		sharpness_level	0
		mode_ref_lf_delta_enabled	1
		mode_ref_lf_delta_update	0
		base_qindex	192
		update y_dc_delta_q	0
		update uv_dc_delta_q	0
		update uv_ac_delta_q	0
		segmentation_enabled	0
		log2_tile_cols	0
		log2_tile_rows	0
		first_partition_length_in_bytes	96
		byte alignment bit	0
255	127	marker_bit	0
128	127	Transform mode	4
128	124	update tx_probs_8x8[0][0]	0
252	248	update tx_probs_8x8[1][0]	1
128	15	delta tx_probs_8x8[1][0]	3
128	96	update tx_probs_16x16[0][0]	0
252	193	update tx_probs_16x16[0][1]	0
248	193	update tx_probs_16x16[1][0]	0
244	193	update tx_probs_16x16[1][1]	0

Total bools per bit in header: 1751/763 = 2.295. 80 bits not coded.

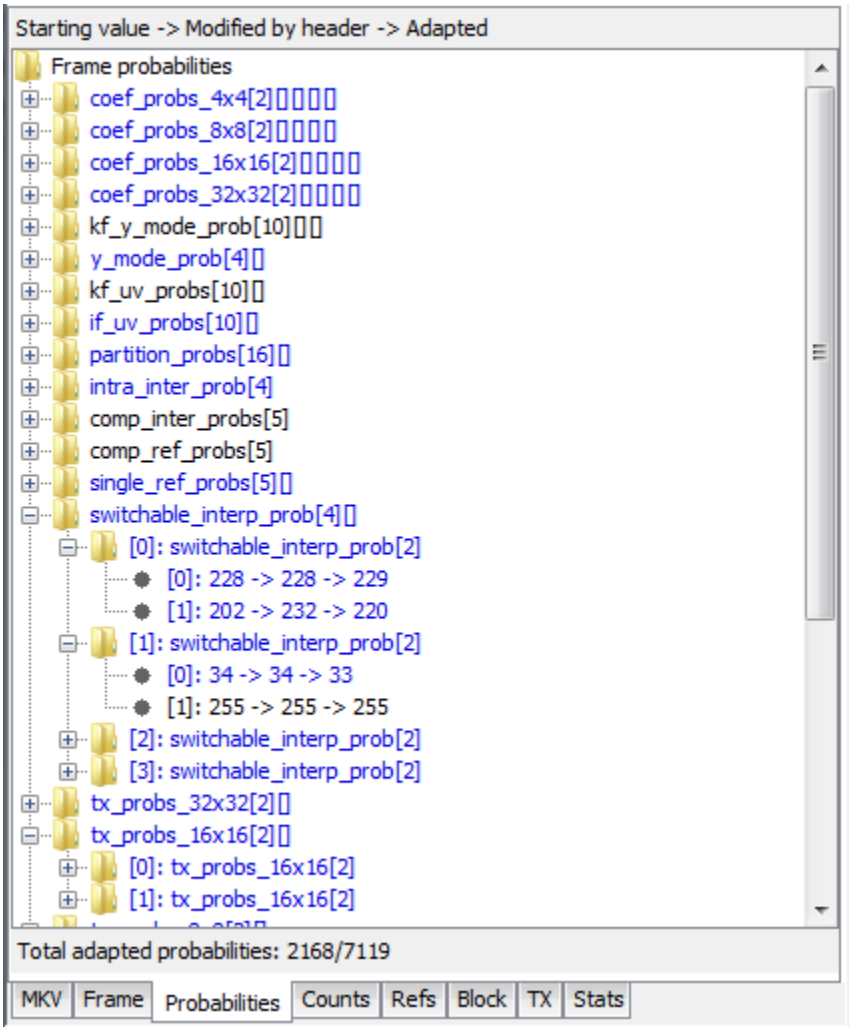
MKV | Frame | Probabilities | Counts | Refs | Block | TX | Stats

The Frame tab shows all syntax elements decoded in the frame header, both the uncompressed and bool-coded partitions. A few stats about the size of the header are shown below the list.

Since the VP9 reference software does not always assign a variable name to each decoded token, a descriptive name has been assigned which conveys the meaning or usage of the token.

The two left columns R and V represent the value of the bool coder's range and value prior to decoding the token. For the uncompressed part of the frame header, these values are not present.

4.1.3.3 Probabilities

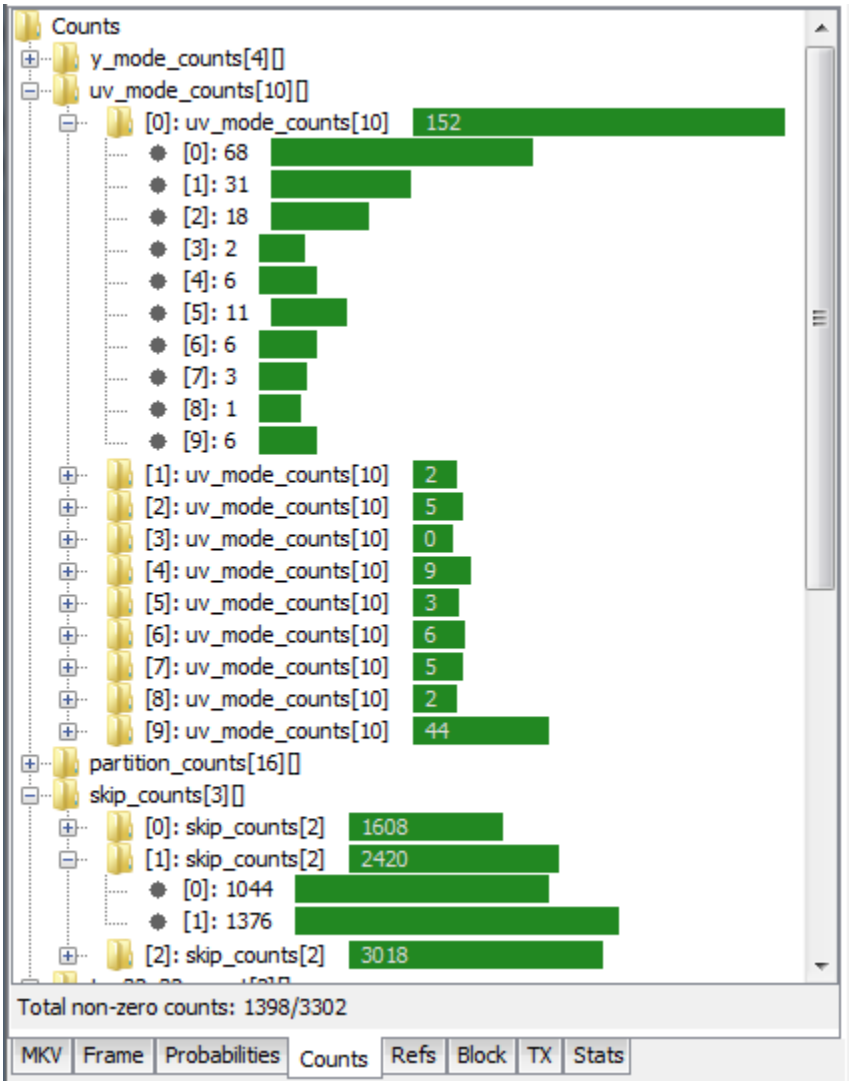


The probabilities used to decode the current frame are visible in this panel. They are organized hierarchically in the same way as the VP9 reference software. Each probability is given as three numbers. From left to right:

1. Probability as it was at the start of the frame decode process. For keyframes or when the reset_frame_context flag is equal to 1, this is the VP9 default value.
2. Probability value after modification by the bool-coded section of the frame header.
3. Probability value after modification by the adaptation process after frame decode.

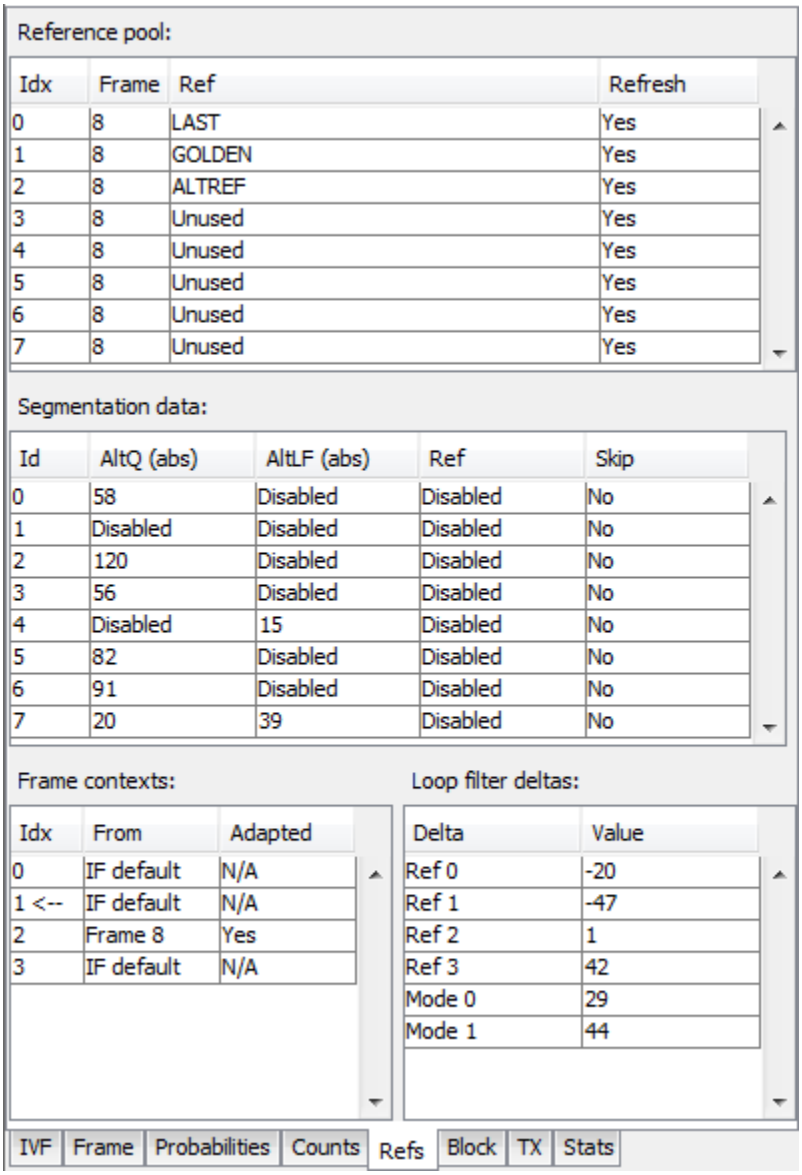
When the probability adaptation process causes the value to actually change, the probability is shown in blue. The total number of probability values that were actually changed is given below the tree.

4.1.3.4 Counts



This tab shows the values of all the counters used for probability adaptation at the end of the frame decode. The counters are organized in hierarchical fashion in the same way as the VP9 reference decoder. Green bars give an indication of relative size of each counter in its group, and the parent icon contains the sum of the counts.

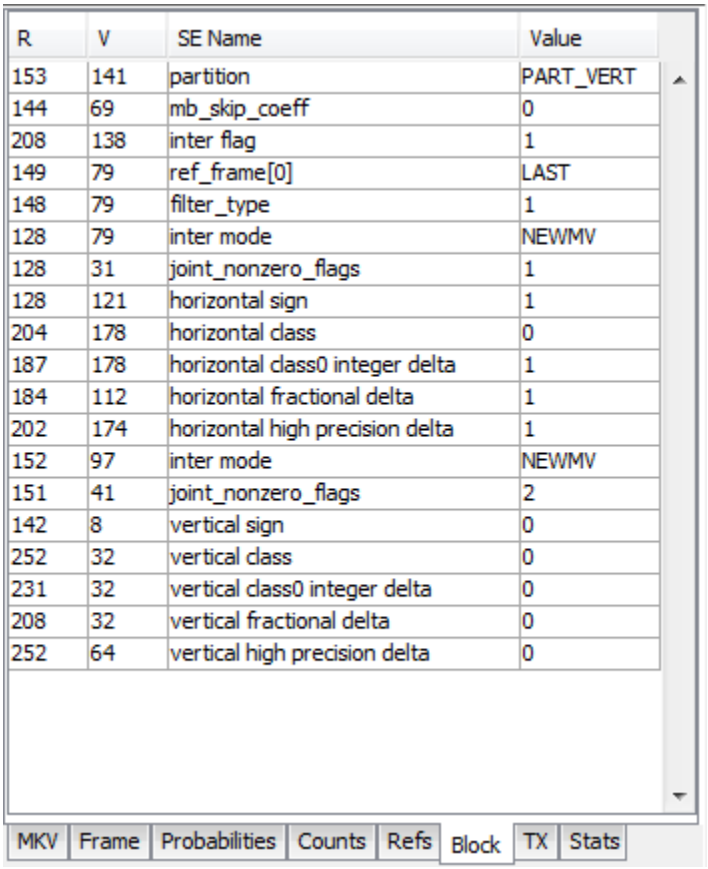
4.1.3.5 Refs



This panel displays the data that persists in the decoder between frames:

- The 8-entry pool of reference frames and which of each of these is considered the Last, Golden and AltRef frame. If any of the frames should be replaced with the current frame after decode (as specified by refresh_frame_flags), the right most column will indicate that.
- Segmentation data for each segment.
- The 4-entry frame context buffer as it applies to the current frame. The center column indicates the frame that generated that context, and the right column indicates whether or not it was the result of adaptation at that time.
- Loop filter deltas for all four references and both modes.

4.1.3.6 Block



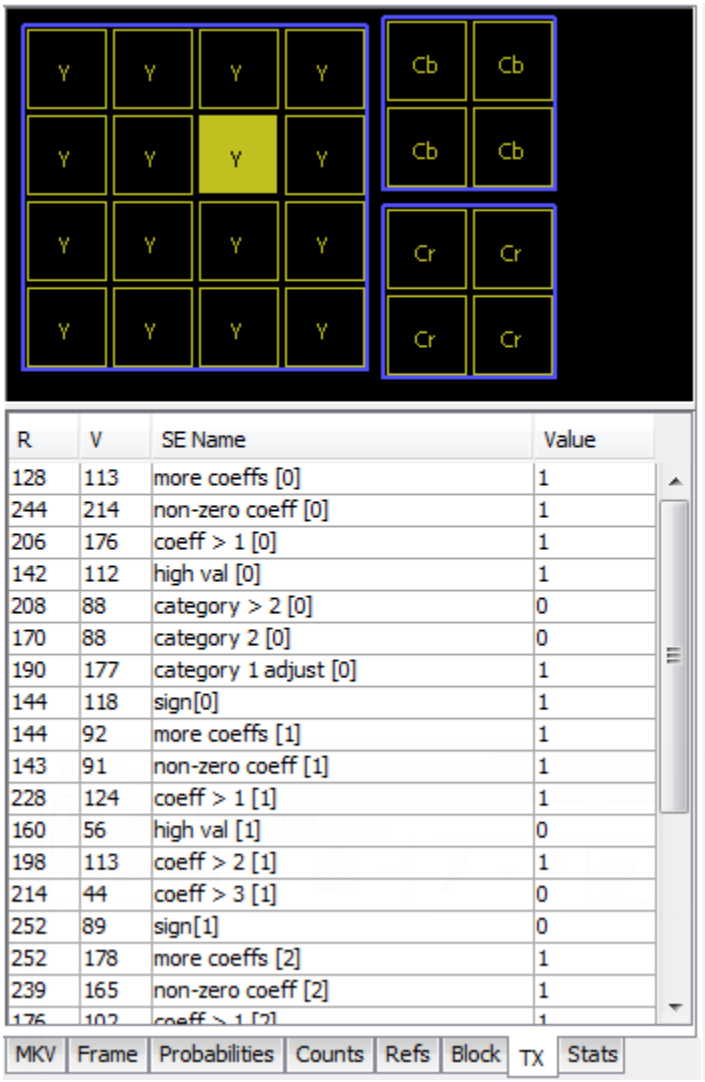
R	V	SE Name	Value
153	141	partition	PART_VERT
144	69	mb_skip_coeff	0
208	138	inter flag	1
149	79	ref_frame[0]	LAST
148	79	filter_type	1
128	79	inter mode	NEWMV
128	31	joint_nonzero_flags	1
128	121	horizontal sign	1
204	178	horizontal class	0
187	178	horizontal class0 integer delta	1
184	112	horizontal fractional delta	1
202	174	horizontal high precision delta	1
152	97	inter mode	NEWMV
151	41	joint_nonzero_flags	2
142	8	vertical sign	0
252	32	vertical class	0
231	32	vertical class0 integer delta	0
208	32	vertical fractional delta	0
252	64	vertical high precision delta	0

This panel displays the syntax elements associated with the currently selected block. Since the VP9 reference software does not always assign a variable name to each decoded token, a descriptive name has been assigned which conveys the meaning or usage of the token.

Note that only the syntax elements at the currently selected block hierarchy are shown. To see the PART_SPLIT tokens at higher levels in the recursive subdivision process, repeatedly click the same block in the main panel to navigate the hierarchy.

The two left columns R and V represent the value of the bool coder's range and value prior to decoding the token.

4.1.3.7 TX

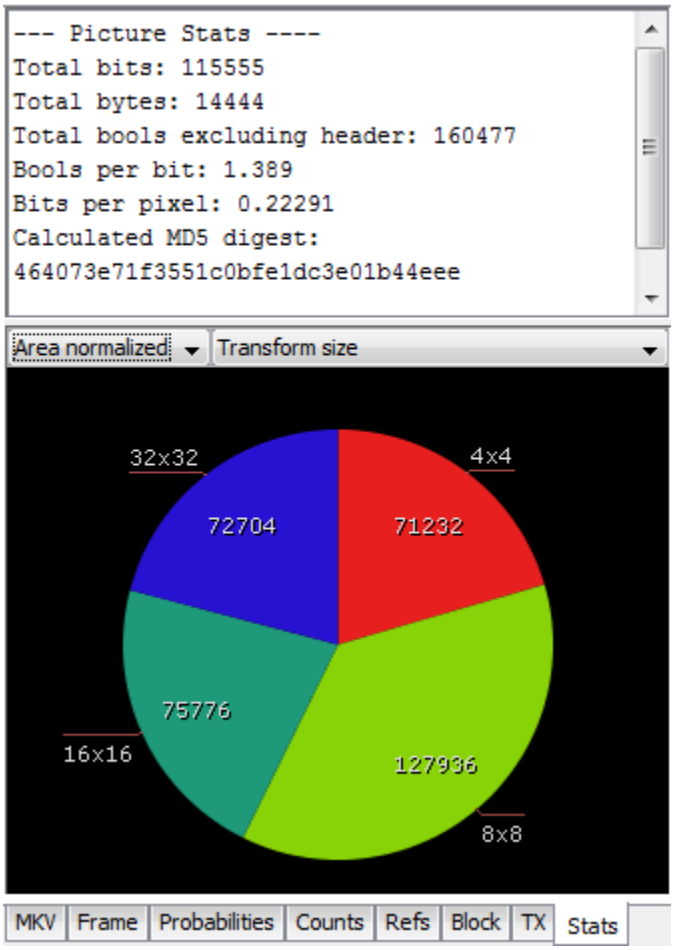


This panel displays the syntax elements associated with the currently selected transform block. Since the VP9 reference software does not always assign a variable name to each decoded token, a descriptive name has been assigned which conveys the meaning or usage of the token.

A particular transform block may be selected in the Main Panel (when in Residual mode), or by clicking in the diagram above the syntax list. Note that when not in detail mode, only the luma transform blocks are selectable from the Main Panel.

The two left columns R and V represent the value of the bool coder's range and value prior to decoding the token.

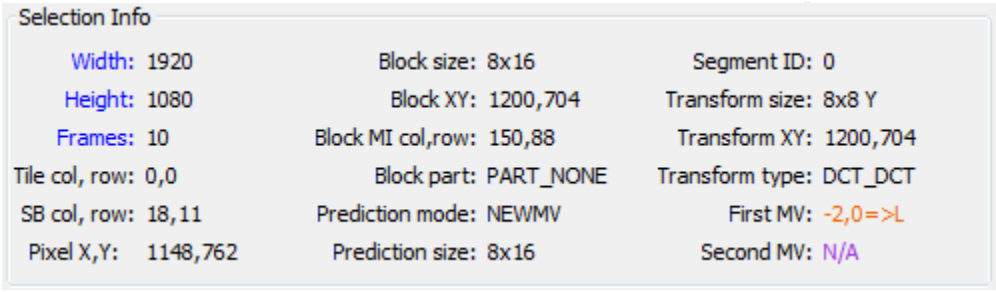
4.1.3.8 Stats



This tab displays various statistics extracted from the current picture. The top half shows some picture size and compression stats as well as the image digest information. The bottom half displays pie charts for a number of metrics. Each pie chart can be drawn normalized or un-normalized. Normalized data is weighted by area or bits. For example there may be a much lower number of 64x64 blocks than 8x8 blocks in the frame (smaller un-normalized pie wedge), but they could still make up the majority of the picture area, making the 64x64 pie wedge large. Normalized numbers in the pie chart are in units of pixels or bits, and un-normalized numbers are raw counts.

The pie chart can be moved by dragging the mouse, and zoomed with the mouse scroll wheel.

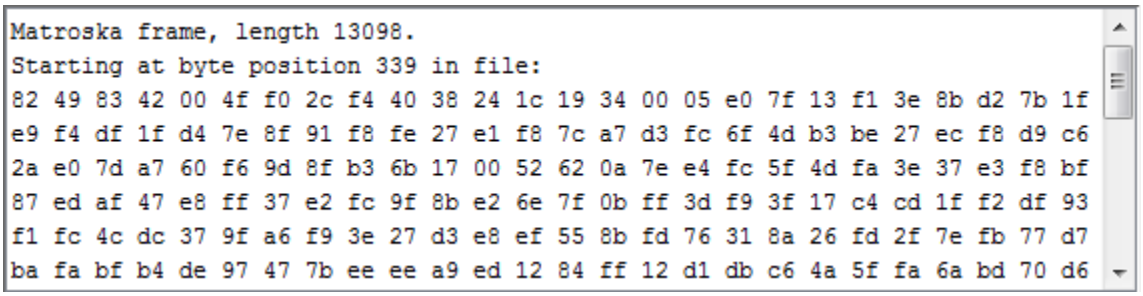
4.1.4 Selection Info

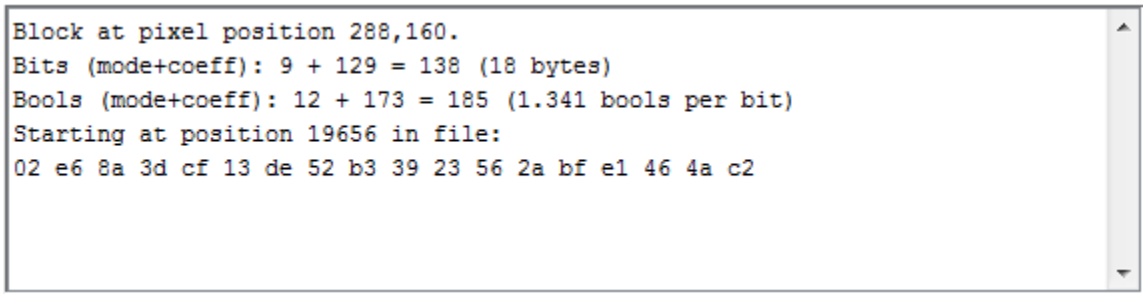


This panel shows a few details about the current selection at a glance:

- Frame height & width
- Number of frames in the bitstream
- Tile row and column number
- Current super-block row and column number
- X/Y position of the mouse cursor in the frame
- Size of the selected block
- X/Y position in pixels of the selected block's upper left corner
- MI (8x8 granularity) column and row number of the upper left of the selected block
- Selected block's partitioning mode
- Selected block's prediction mode
- Size of the selected prediction block
- Segment ID of the selected block
- Transform size used in the selected block (luma)
- X/Y position in pixels of the selected transform block (Residual Mode)
- Type of transform used on the selected transform block (Residual Mode)
- Value of the motion vector(s) used by the current prediction block in $1/8^{th}$ pel. An arrow and letter indicate the reference frame: L for Last, G for Golden and A for AltRef.

4.1.5 Raw Bytes





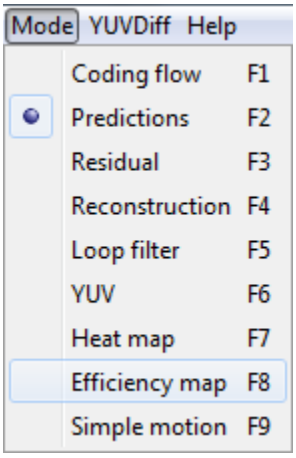
This resizable panel displays raw bytes from the bitstream as it applies the current selection, as well as few extra details. The current selection can be a particular block, a Matroska element/frame or an IVF frame (depending on the current bitstream's container).

4.1.6 Messages

This resizable panel in the lower right of the UI displays messages about the decoding process, and shows progress of any actions that may take a while to complete.
The 3 numbers in the lower right of this panel represent the memory state of JVM. From left to right the 3 numbers are:

- Amount of memory used
- Total memory claimed by JVM
- Maximum amount the JVM might claim

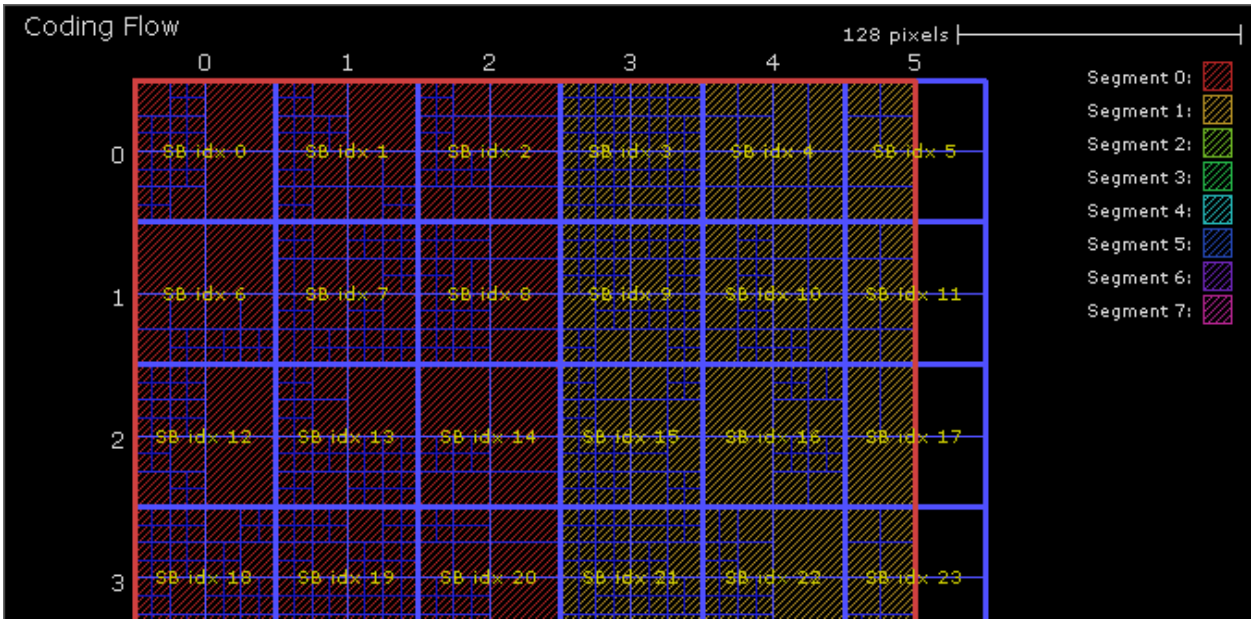
4.2 Modes

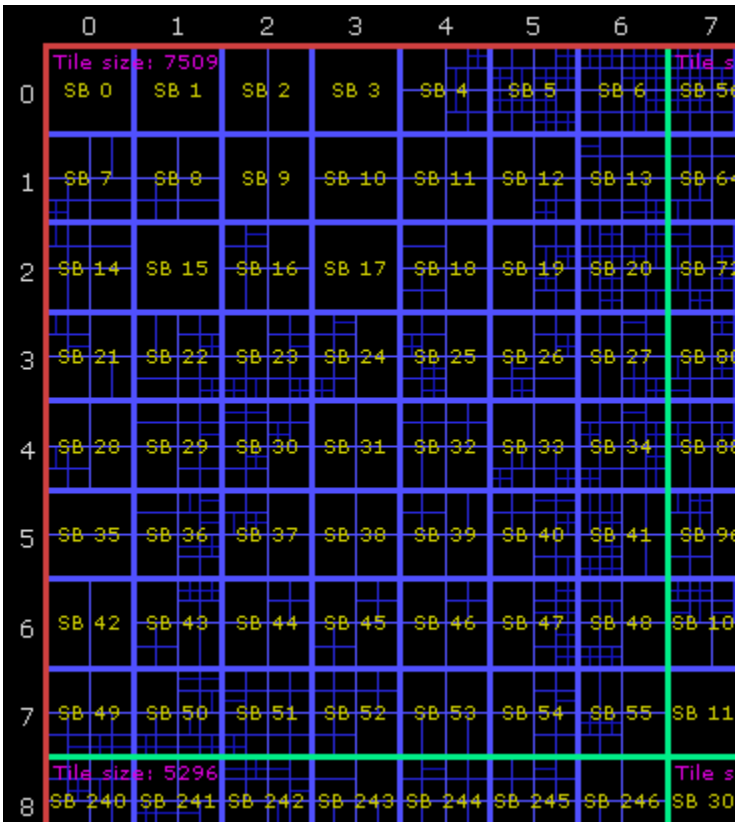


With a VP9* bitstream loaded, Intel® Video Pro Analyzer can be put into one of 9 modes using either the F1-F9 keys, or using the Mode menu. The mode selection affects only what is displayed in the Main Panel.

Most modes can also show details of the current selection (block, prediction/transform block). This can be toggled with the right mouse button or the main panel's button strip.

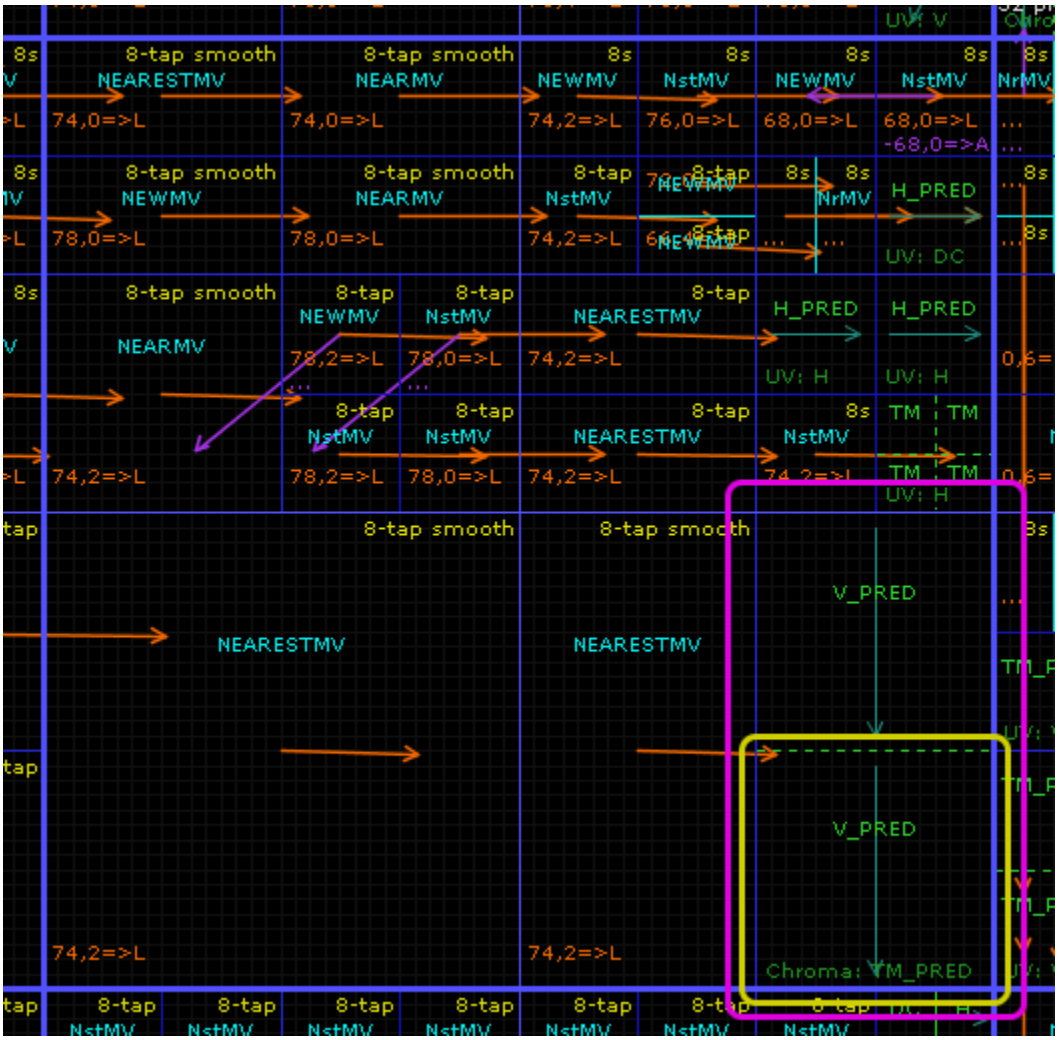
4.2.1 Coding Flow





Coding flow mode gives an overview of the way the frame is constructed. Each superblock's decode index is shown in yellow and block-partioning is indicated with blue boundaries. Tile boundaries, when present, are shown with thick green lines and the tile size as coded in the bitstream is shown at the top left of each tile that has a size coded. When segmentation is enabled, each segment is shaded with a unique color. No detail mode is available.

4.2.2 Predictions



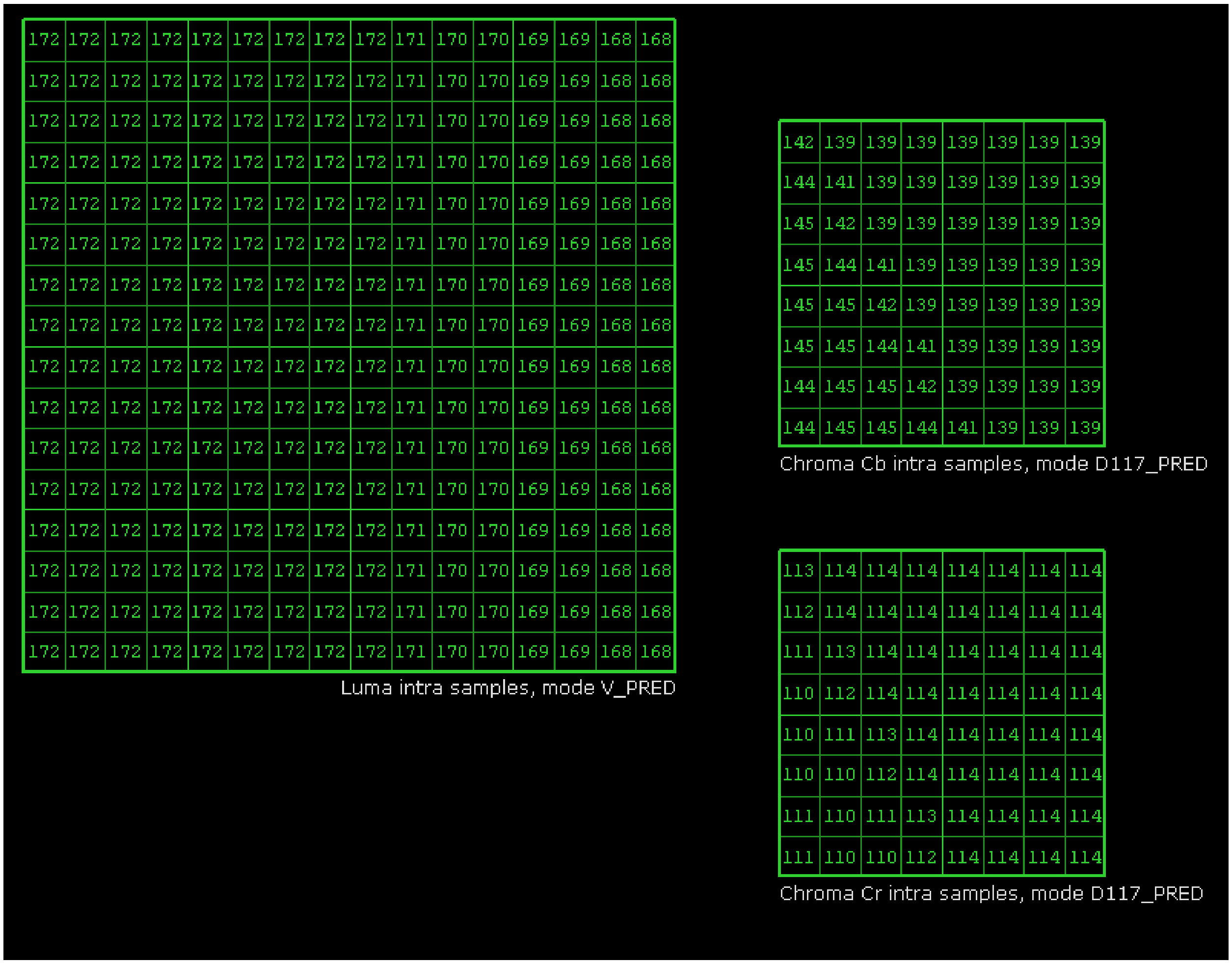
This mode shows the details of the prediction blocks in the frame. Block splits are indicated with blue lines, the selected block is surrounded with a pink box and the selected prediction block is surrounded with a yellow box. When a prediction block is selected, the syntax used to code it is displayed in the Block tab of the Left Panel. Clicking a block repeatedly will navigate the recursive block hierarchy.

Intra prediction block contain the intra luma mode in bright green, and the chroma intra mode in dark green near the bottom of the block. When the intra prediction operation is split due the size of the transform blocks, the intra split boundaries are indicated with dashed green lines. Directional intra luma modes also contain a turquoise arrow indicating the prediction direction.

Inter prediction blocks show the inter mode in cyan, the first motion vector in orange and the second motion vector in purple. The motion vector value is in units of $1/8^{th}$ pixels. After the motion vector value the reference frame it points to is indicated with a single letter: L for Last, G for Golden and A for AltRef. The motion compensation filter type that applies to the prediction block is indicated in yellow near the top right corner of the block.

4.2.2.1 Prediction Detail Mode

Entering detail mode on a selected prediction block allows the sample values of the prediction process to be viewed directly.



172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168
172	172	172	172	172	172	172	172	172	171	170	170	169	169	168	168

Luma intra samples, mode V_PRED

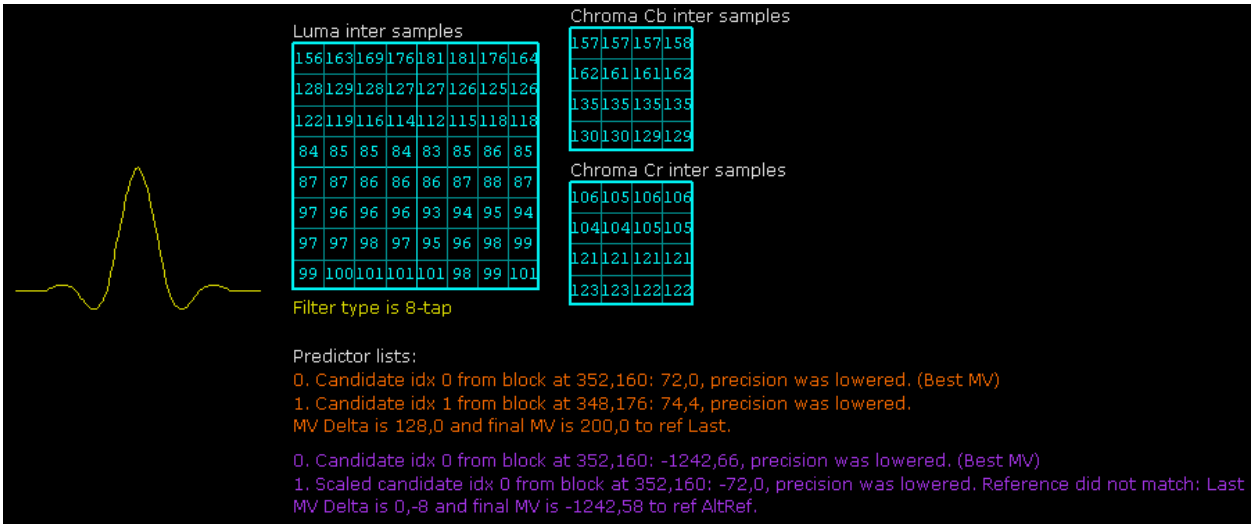
142	139	139	139	139	139	139	139
144	141	139	139	139	139	139	139
145	142	139	139	139	139	139	139
145	144	141	139	139	139	139	139
145	145	142	139	139	139	139	139
145	145	144	141	139	139	139	139
144	145	145	142	139	139	139	139
144	145	145	144	141	139	139	139

Chroma Cb intra samples, mode D117_PRED

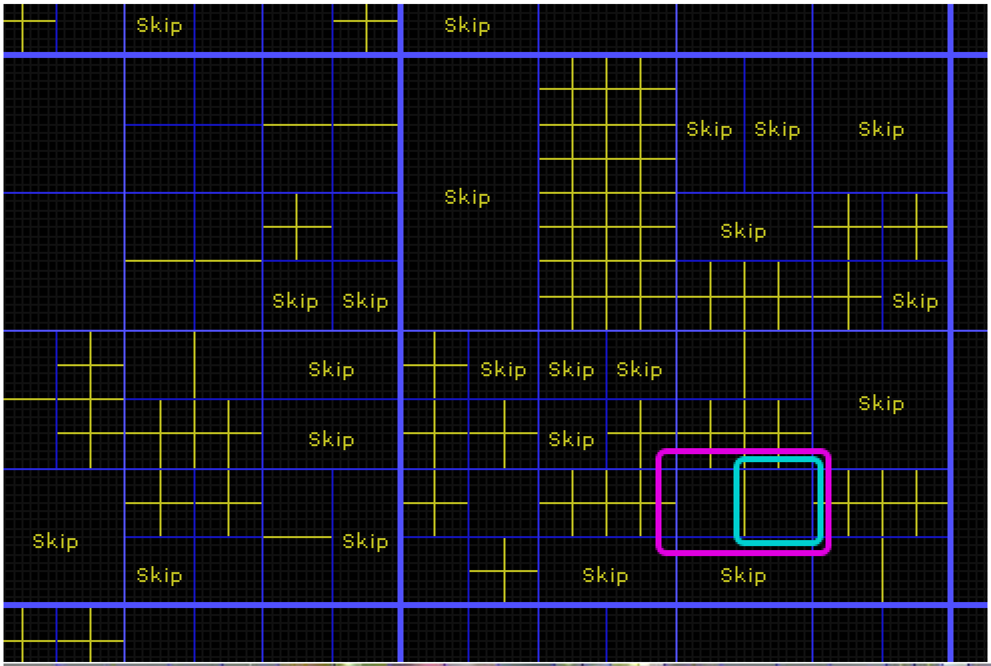
113	114	114	114	114	114	114	114
112	114	114	114	114	114	114	114
111	113	114	114	114	114	114	114
110	112	114	114	114	114	114	114
110	111	113	114	114	114	114	114
110	110	112	114	114	114	114	114
111	110	111	113	114	114	114	114
111	110	110	112	114	114	114	114

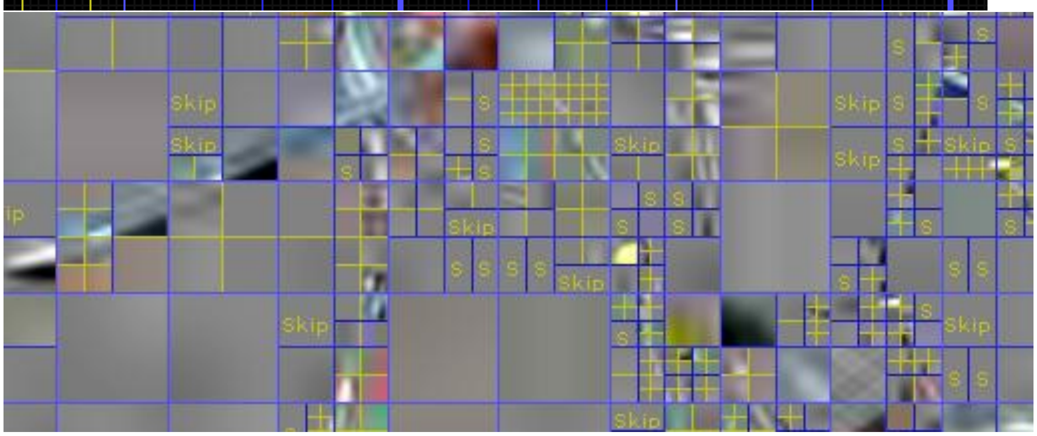
Chroma Cr intra samples, mode D117_PRED

Inter-predicted blocks show the filtered prediction samples arrays of each component, the predictor list used for motion vector prediction (for each motion vector) and the filter type applied. To the left of the prediction sample arrays the impulse response of the applied filter is drawn (mostly decoratively).



4.2.3 Residuals





Residual mode allows inspection of the transform structure of each block. Block boundaries are displayed in blue, the currently selected block is outlined with a pink box and the selected transform block is outlined with a blue box. Blocks that code no residual signal are indicated with “Skip” in yellow. Blocks that use a transform size that is smaller than the block’s own size show the transform boundaries within the block in yellow.

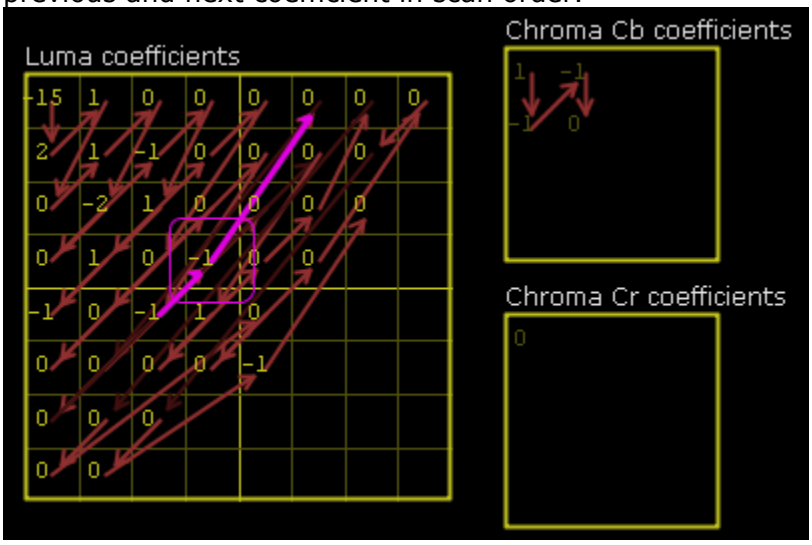
Clicking a particular transform block causes the syntax elements used to decode it to be visible in the “TX” tab of the Left Panel.

The residual image itself can be shown using the “Pic” button on the bottom button strip. Residual values of 0 are flat gray, negative values are darker and positive values are brighter.

4.2.3.1 Residual Detail Mode

To view the full details of the transform blocks belonging to a particular block, selected a transform block and enter detail mode by right clicking or using the detail mode button at the bottom of the main panel. The currently selected transform block is highlighted with the sample grid. The transform structure of the selected block is drawn three times, arranged in a column from top to bottom, showing the 3 major steps in recovering the residual signal:

Decode and inverse scan process. In this diagram the scan pattern is drawn for each transform block. In an attempted to reduce clutter, larger scan pattern jumps are drawn with a darker color. The currently selected transform block is highlighted with the sample grid. Moving the mouse over a particular coefficient will highlight the previous and next coefficient in scan order.

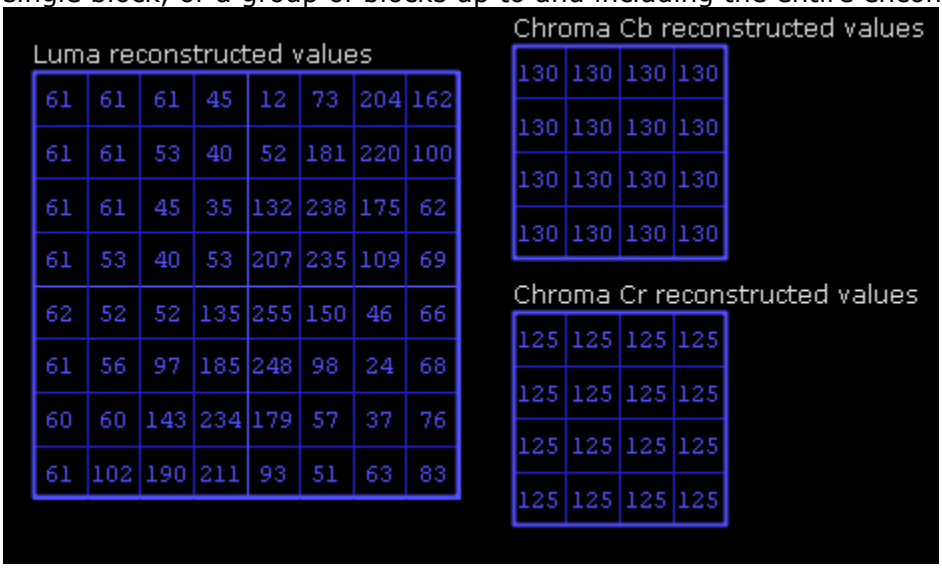


4.2.4 Reconstruction

In Reconstruction mode the reconstructed samples prior to loop filtering can be inspected. As with Prediction/Residual mode, block boundaries are shown with blue lines. Blocks with no coefficients are marked with "Skip" in white.

4.2.4.1.1 Reconstruction Detail Mode

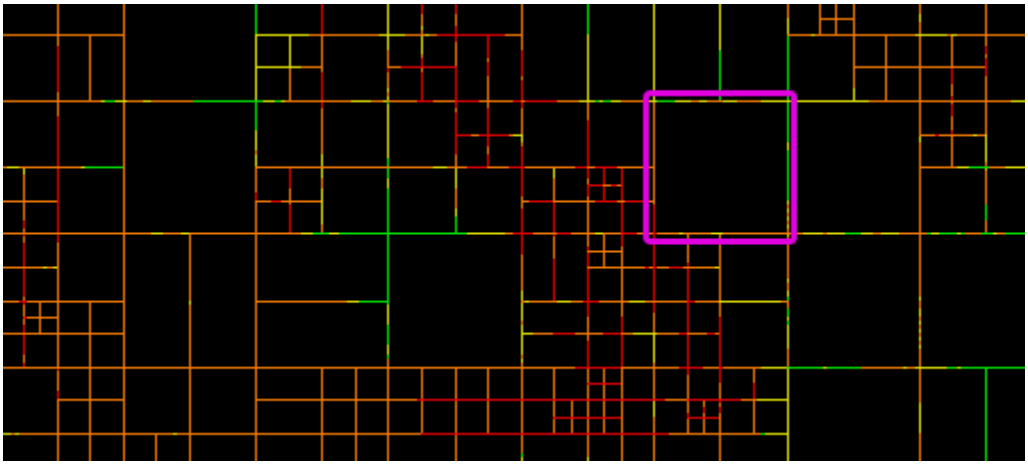
In detail mode, the reconstructed sample values can be inspected. The selection may be a single block, or a group of blocks up to and including the entire encompassing super-block.



4.2.5 Loop Filter

Loop filter mode shows all processed edges in the frame. Edges are 1 sample high/tall, and as color coded as follows:

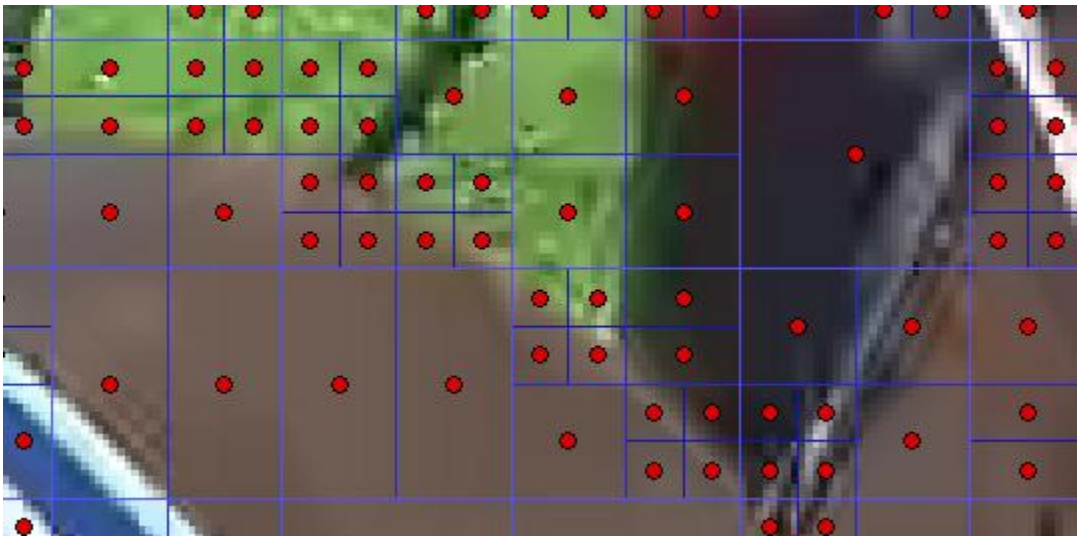
- Green – 16-tap filter
- Yellow – 8-tap filter
- Orange – 4-tap filter
- Red – Not filtered, threshold not met (but processed).



4.2.5.1 Detail Loop Filter Mode

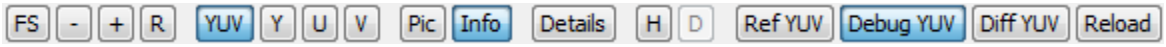
The filtered sample values can be inspected directly in detail mode. As with Reconstruction Details, the inspected area can consist of one or more blocks. Sample values are indicated in yellow if they were modified by the deblocking process. Samples to the left and above the current block need for the filter are shown as well. Filtered edges belonging to the selected blocks are shown in thick dashed lines, color coded the same way as in non-detail mode.

4.2.6 YUV



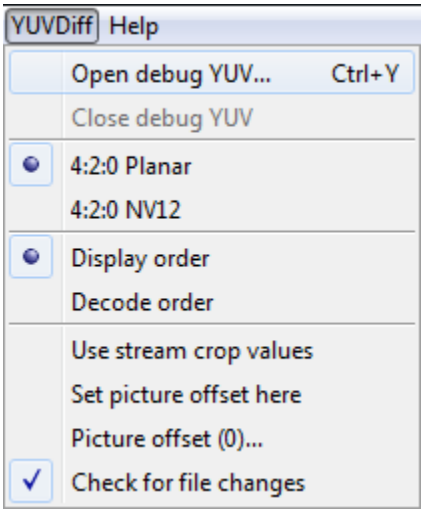
YUV mode allows inspection of the final decoded sample values, without additional overlay data. In this mode, the YUVDiff feature is exposed. This feature allows for comparison of an external decoded YUV file with the decoded bitstream. Supported formats are 4:2:0 planar and NV12. Also, YUV files that are zipped or gzipped may be opened directly without the need to decompress separately. If a zip file contains more than one YUV file, only the first one is used. When a YUV file is loaded, any mismatches will be indicated with a red dot in the CU containing the mismatch. This allows the user to quickly identify the nature of the mismatch which can assist with debug.

When a file is opened, four additional buttons become visible on the lower left of the main window:



- Ref YUV: The original, expected YUV image.
- Debug YUV: The loaded debug YUV file.
- Diff YUV: The delta image. Like in Residual mode, areas with zero delta (i.e. Original and Debug image are identical) are flat gray. Areas where the debug YUV has a lower value are darker, and areas where the debug YUV has a higher value are brighter.
- Reload: Reloads the YUV file, which can be useful as a shortcut to loading via the YUVDiff menu.

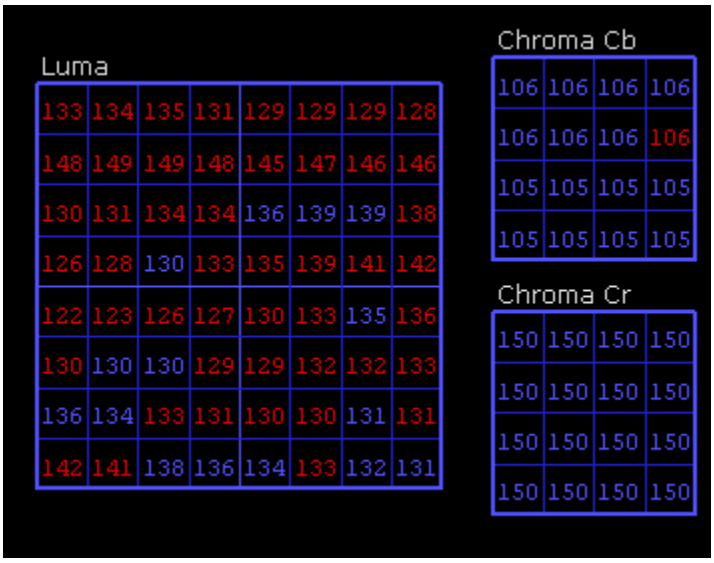
4.2.6.1 YUVDiff Menu

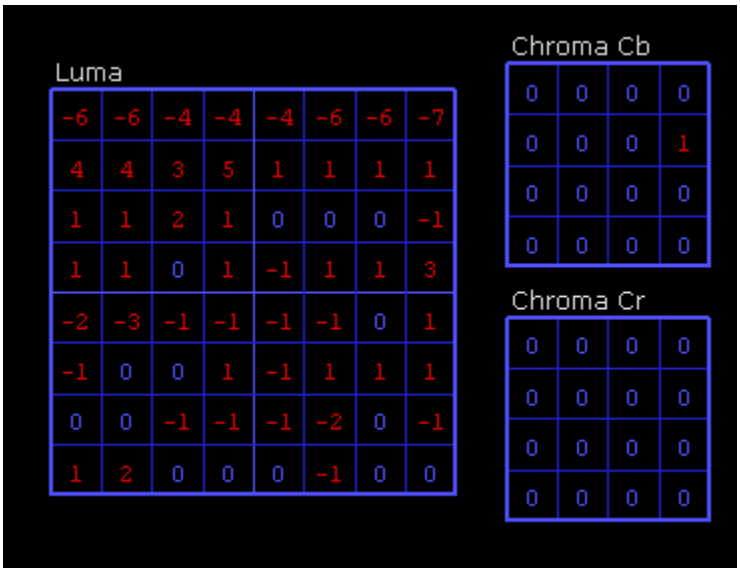


- Open: Brings up a file chooser to open a YUV file
- Close: Closes the currently opened YUV file. Any mismatch indicators will disappear.
- 4:2:0 Planar: Causes Intel Video Pro Analyzer to interpret the loaded YUV file as 4:2:0 Planar.
- 4:2:0 NV12: Causes Intel Video Pro Analyzer to interpret the loaded YUV file as 4:2:0 NV12. This option is chosen automatically when the loaded YUV file has the file extension ".nv12".
- Display order: Causes Intel Video Pro Analyzer to use display order picture numbers when determining how far to seek into the YUV file to extract the image to compare with the current picture.
- Decode order: Causes Intel Video Pro Analyzer to use decode order picture numbers when determining how far to seek into the YUV file to extract the image to compare with the current picture.
- Use stream crop values: When checked, YUV files are assumed to contain samples only within the cropping window as defined by the loaded bitstream. Samples outside this window are not compared, and are assumed 0 when viewing the loaded YUV image directly.
- Set picture offset here: Shortcut for setting the picture offset to the current picture number. See bullet below.
- Picture offset: This brings up a dialog allowing the user to enter the picture number of the first picture in the YUV file. For example if a 100-picture bitstream is loaded but the YUV file only contains pictures 80-99, the user would enter 80 to properly line up the YUV file with the decoded bitstream.
- Check for file changes: When checked, this option causes Intel Video Pro Analyzer to periodically check if the loaded YUV file has changed on disk since it was last loaded. If the file has indeed changed, a dialog pops up offering a chance to reload the YUV file.

4.2.6.2 YUV Detail Mode

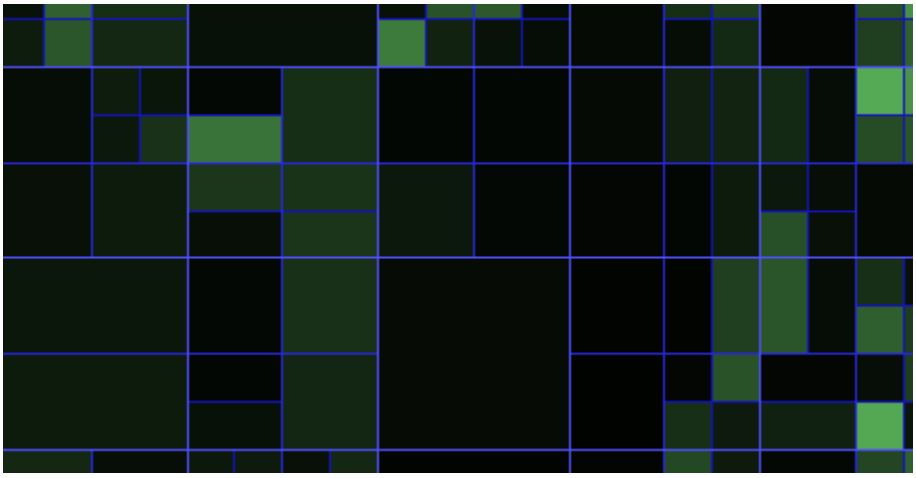
In detail mode the expected YUV values can be inspected. Note that the displayed sample values are the same as those in Loop Filter mode since loop filtering is the last step in the decode process. When a YUV file is loaded, the debug and delta YUV values can be inspected as well. Mismatching values are shown in red.

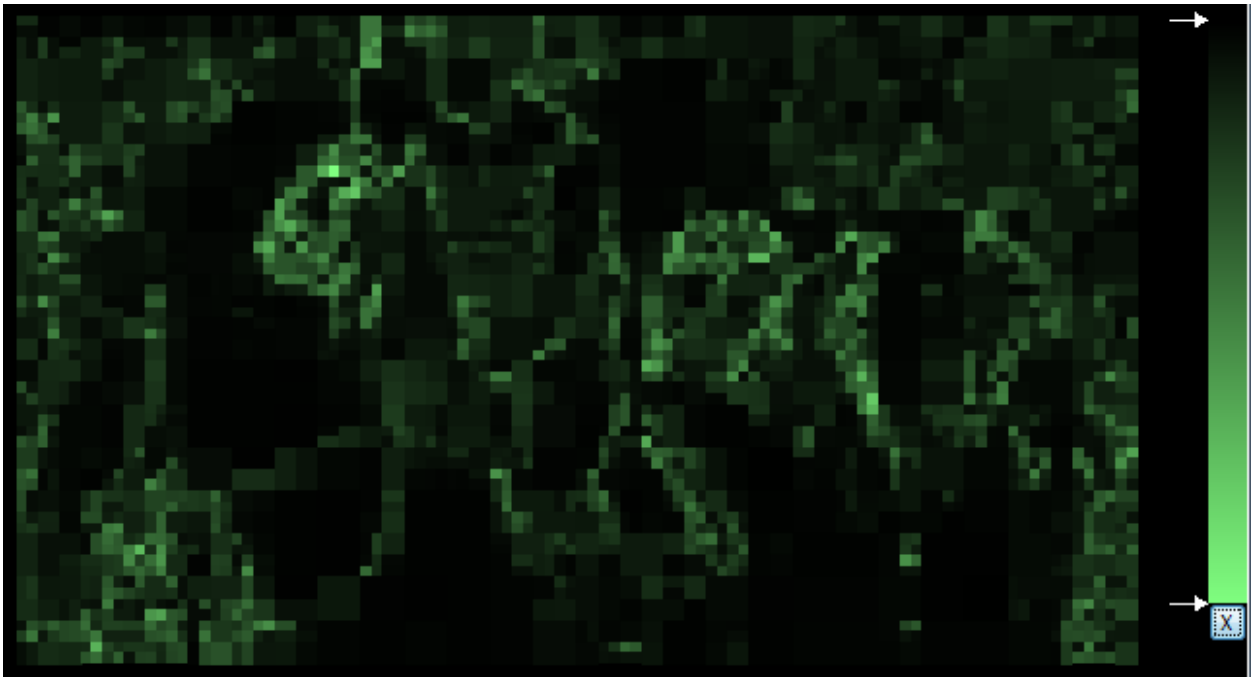




4.2.7 Heat Map

Heat map mode shows visually how the compressed bits of the picture are distributed spatially. Blocks with more bits per pixel are brighter than blocks with less. By default this mode also shows the block boundaries in blue. It may be useful to turn off this overlay using the "Info" button on the lower left of the main panel. Heat Map does not offer a detail mode.



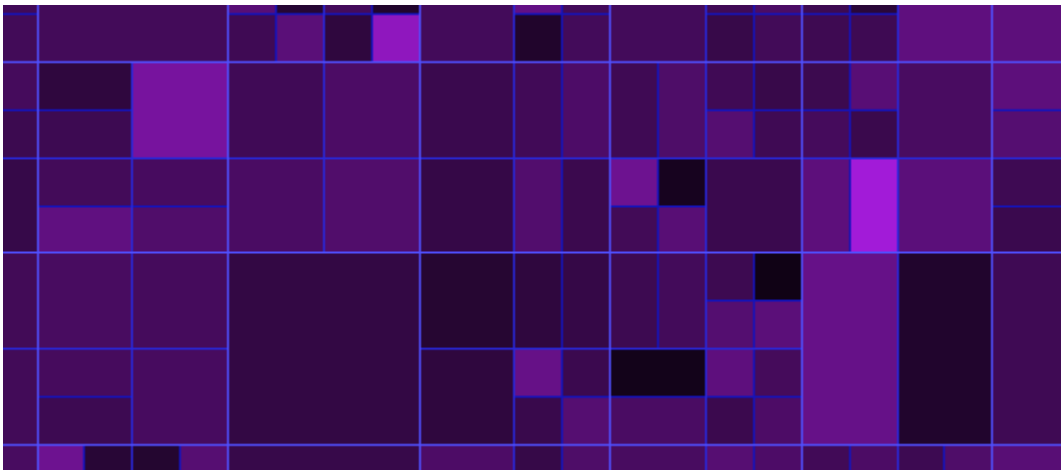


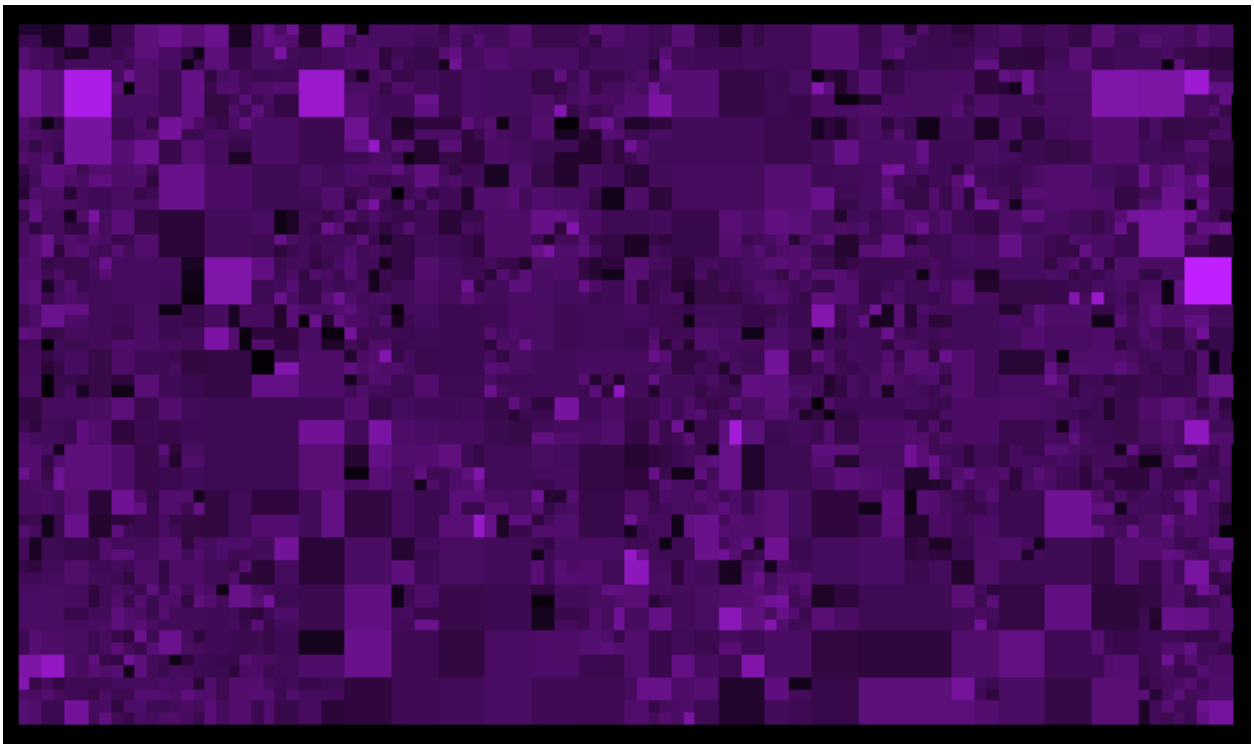
The gradient used to display the heat map may be edited:

- Drag the gradient markers up and down to reposition them.
- Drag a marker away from the gradient to remove it. A red X indicates removal will take place.
- Double-click a marker to change the color.
- Double click the gradient bar to add a new marker.

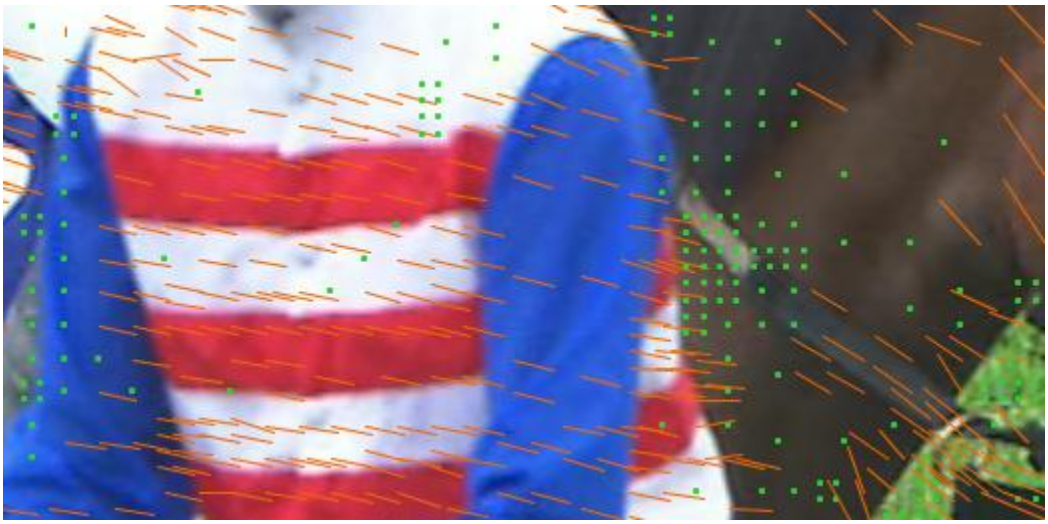
4.2.8 Efficiency Map

Efficiency map mode shows visually how efficient the arithmetic coding of each block is. Blocks with more bools per bits are brighter than blocks with less. By default this mode also shows the block boundaries in blue. It may be useful to turn off this overlay using the "Info" button on the lower left of the main panel. Efficiency Map does not offer a detail mode.





4.2.9 Simple Motion



Simple Motion mode offers a way to quickly view the modes and motion of a picture at a glance. No overlays are present except a colored dot showing information about the prediction mode. Green dots indicate intra blocks, and purple/orange motion vectors for inter blocks are drawn as simple lines. When the motion vector is very small, a dot is drawn instead so that the mode is still easy to identify.

5 Command Line Features

```
Intel Video Pro Analyzer 2014 - version 1.1.0
<file name>: Open the indicated file immediatly after loading the UI.
-regress <file name>: Don't bring up the GUI, just decode the stream.
-dump: write the decoded sequence to output.yuv (default is decode order). Must be used with -regress
-crop: crop the output according to stream parameters. Must be used with -regress and -dump
-stats: write some stats to stats.csv. Must be used with -regress
-norm_pix: Normalize stats by pixel area
-norm_bits: Normaluze stats by compressed bits
-fast: Skip reconstruction. Must be used with -regress and -stats, and cannot be used with -dump.
-display_order: if used with -dump, the output yuv is written in display order.
```

Intel® Video Pro Analyzer has a few command line options:

- Adding -help prints the available command line options.
- Adding a file name to the command line will cause Intel Video Pro Analyzer to open that file in the UI right after startup, bypassing the file menu. This can be helpful as part of a debug script.
- -regress: This causes Intel Video Pro Analyzer to run without the UI. A file must be specified. This is mainly useful only for HEVC streams with digest SEI messages, as it will display an error message and abort if the digest mismatches. For VP9* and other HEVC streams, the decoder will simply run and exit with no output.
- -dump: Must be used with -regress and a file name. Causes Intel Video Pro Analyzer to decode the specified file and write the decoded output sequence to a file called output.yuv in decode order. The format written is planar YUV. To change the output order to display order, add the command line switch -display_order. Adding -crop will crop the output to the display size, as specified by the bitstream.

6 Miscellaneous

A few other features to keep in mind:

- Intel® Video Pro Analyzer saves various settings upon exit. Settings include window size, recent files etc., and are restored on the next launch.
- If the loaded bitstream is modified or deleted while loaded, a dialog will pop-up asking to reload or close. This protects against unintended behavior.
- Because Intel Video Pro Analyzer runs in the Java* Virtual Machine, it starts up with a defined maximum memory allocation (max heap). The Windows executable launches the JVM requesting a max memory allocation of 2GB. This means that Intel Video Pro Analyzer may use up to 2GB if a large sequence is loaded. If more memory is needed, the .jar file may be launched manually, specifying the desired heap max using the –Xmx switch as follows, replacing N with a number in gigabytes:
    - `java -jar -XmxNg -jar IntelVideoProAnalyzer.jar <optional args>`

  It may be useful to place the above command in a batch file or script of some kind if you are frequently loading large sequences. In general, 2Gb should be sufficient for most sequences 1080p or smaller.