White Paper

Addressing Challenges in Developing HEVC based Codecs with Good Analysis Tools

VEGA Analyzer Team
Digital Media Group, Interra Systems, Inc.

Interra Systems, Inc.
1601 S. De Anza Boulevard, Suite 212
Cupertino, CA 95014
USA

High Efficiency Video Coding (HEVC) is an emerging video coding standard introduced by ITU-T VCEG and ISO/ IEC MPEG. HEVC is designed for encoding and decoding video streams that can be stored and delivered more efficiently and economically compared to its predecessors such as the H.264 or MPEG-2 standards. The HEVC standard aims to deliver an average bit rate reduction of 50% for the same video quality as compared to H264 and also delivers higher quality at the same bit rate. Demand for high quality video for a multitude of consumer-driven applications has driven this standard into prominence in the recent years.

HEVC introduces several additional coding tools to increase the compression performance. This performance gain comes at the cost of increased complexity, which is incorporated by adding tools like variable block size, additional prediction modes, SAO filtering and other highly computational modules. For achieving the goal of real-time processing, HEVC provides several parallel processing tools over previous standards. The processing time can be reduced significantly by effectively utilizing components like tiles, slice and WPP.

A complex video compression standard like the HEVC can pose tremendous challenges for creating robust, high quality products including encoders, decoders, multiplexers, transcoders etc. 4K and 8K videos further add to the complexity as they require faster processing power as well. In-depth analyzers that can provide deep insight into the compressed stream are indispensible tools for professionals and engineers involved with research, development and testing of HEVC video.

In this paper we will explore the features of HEVC video and the challenges posed by this standard as far as analysis and debug are concerned. Further, we will discuss the powerful mechanism of debugging these complicated modules through Interra Systems' VEGA Media Analyzer.

1. BACKGROUND

H.264/AVC compression standard provides a satisfactory combination of compression efficiency and quality. This standard was developed in 1999-2003, and offered consistent improvements until 2009. Factors such as exploding video content, increased importance of video quality, emergence of new super-HD formats like 4K and 8K, and services like VOD, and streaming have created a very strong requirement for a codec that can offer superior coding efficiency and quality. With this goal in mind, JCT-VC proposed a new, enhanced codec, which could reach the same video quality as AVC at nearly half the bitrate. This project was called as High Efficiency Video Coding (HEVC).

HEVC is an open standard, defined by standardization organizations in the telecommunications (ITU-Ts VCEG) and technology industries (ISO/IEC MPEG). The emerging High Efficiency Video Coding (HEVC) standard has improved the coding efficiency drastically, and can provide equivalent subjective quality with more than 50% bit rate reduction compared to its predecessor H.264. As expected, the improvement on coding efficiency is obtained at the expense of more intensive computation complexity.

The techniques and algorithms used in HEVC are significantly more complex than those of H.264 and MPEG-2. While HEVC tools are designed to improve parallel processing capabilities, the sheer number of tools with increased complexity is very large. it is estimated that HEVC encoding will require up to ten times more processing power than H.264 encoding. There are more decisions to make when encoding a given video stream and as a result, more calculations need to be made in compressing video assets.

In the next section, we will examine different HEVC structures, which add significant complexity but make it twice as efficient as H.264.

2. CHALLENGES IN HEVC CODEC DEVELOPMENT

Complexity Analysis

HEVC offers more possibilities to split a frame into multiple units and more ways of combining different coding tools and parameters. Though this doesn't have a significant impact on the decoder from the complexity aspect, it imposes a heavy computation burden to the encoder to fully leverage these capabilities.

HEVC has many more mode combinations as a result of the added flexibility from the quadtree structures and the increase of intra picture prediction modes. For all-intra configuration, the coding complexity mainly comes from the mode decision of all available candidate modes. The most time-consuming part is the motion estimation as a result of the multiple reference motion compensation and sub-pixel interpolation etc.

Complexity of some key modules such as transforms, intra picture prediction, and motion compensation is higher in HEVC when compared with H.264. An encoder fully exploiting the capabilities of HEVC is thus expected to be more complex than an H.264 encoder. Computation complexity in HEVC encoder can be managed by optimizing the process of intra mode decision, reference frame selection and inter CU splitting decision.

Compression Analysis

HEVC is a block-based hybrid-coding scheme. For achieving the higher compression performance, the major contributor in HEVC is the introduction of larger block structures with flexible sub-partitioning mechanisms. HEVC supports large block sizes for encoding large smooth regions more effectively while it also has a more flexible partitioning structure to allow smaller blocks to be used for more textured and uneven regions.

Encoders make intelligent decisions to achieve most efficient bit-rate reduction while maintaining a certain picture quality level. By varying the quantization parameter values and implementing different block-based partitioning, one can expect increase in compression efficiency.

When testing the compression efficiency of different test streams generated from different encoders, additional tools are required to provide comparison between various parameters like bit-rate, frame sizes, compression ratio, QP, Buffer occupancy etc. to better understand the impact of changing values on the test streams.

Quality Analysis

Encoder developers strive to build an encoder that can produce highly compressed stream, and yet, maintain the same quality as that of the original video. The deviation from the original bit stream is seen as distortion and therefore, minimizing the distortion level by improving the encoder becomes an important proposition.

The various mathematical models used are Signal-to-Noise-Ratio (SNR), Mean-Square-Error (MSE) and Structural-Similarity-Index (SSIM). Apart from these, the encoding process also introduces visual quality artifacts. Blockiness, blurriness, loss of contrast, pixelation, ringing, contouring, posterizing and mosquito noise are some of the artifacts introduced in video by encoder, and pose great challenges for the encoder developer.

A good analysis tool should provide as much detail about the artifacts affecting the video quality as possible.

Bit Stream Violations

A bit stream generated by any encoder shall fulfill all requirements specified in conformance clauses of the specification. With HEVC, analyzing conformance checks become a challenging task due to large array of coding tools and syntax elements provided in it. Any encoder developer while experimenting with the coding tools provided, may violate bit stream conformance accidentally.

So with better compression and quality, conformance checks also become an integral part of the encoder development process. Any tool that facilitates encoder performance analysis can be of great help to the developers, if conformance violation analysis is also provided alongside.

Buffer Analysis

Video coding standards use intra- and inter-prediction techniques to compress the video frames which results in variations in the coding bits required to compress each frame. The compressed video might be transmitted over channels at approximately constant bit-rate. To handle fluctuation in the bit rate of the video transmitted at constant or near to constant bit-rate, the hypothetical reference decoder (HRD) model is used at both encoder as well decoder side.

Video coding standards do not focus on specific encoder or decoder buffering mechanisms, but they expect encoders to control bit-rate fluctuations so that a hypothetical reference decoder (HRD) of a given buffer size would decode the video bit stream without suffering from buffer overflow or underflow. An underflow causes delay noise and overflow causes packet loss.

This is an ideal decoder model that decoder manufacturers can use as a reference for their implementations, but its main goal is to impose basic buffering constraints on the bit-rate variations of compliant bit streams.

It's really helpful for the encoder or decoder developer, if a tool can provide quick and helpful information to debug the overflow and underflow in the video stream.

3. DEBUGGING HEVC COMPONENTS

Various new components are introduced in HEVC. The table-1 shows some of the critical components of an HEVC video and where they stand in terms of challenges. Let's see the "Coding Blocks" components in table-1, the challenge here is that it's a complex structure, codec developers need to optimize the coding blocks for achieving compression, maintaining the quality, identifying the violations and comparing coding blocks of different streams generated from different encoders.

Components/ Challenges	*Coding Blocks*	*Intra Prediction*	*Motion Compensation*	*Residual*	*SAO*	*Quality*	*HRD*
Compression Analysis	√	√	√	√			
Complexity Analysis	√	√	√		√		
Quality Analysis	√	√	√	√	√	√	
Violations	√		√	√			
Buffer Analysis							√
Comparative Study	√	√	√	√			

These challenges are discussed in detail below and also provide a mechanism to debug and overcome them.

<u>Coding Blocks</u>

All the codecs break down a picture into small square blocks and then encode the blocks. HEVC supports a flexible coding structure as compared to an H.264 video, that has fixed size macro blocks of 16x16 pixels. The analogous structure in HEVC is coding tree block (CTB) that can go from 16x16 pixels to 64x64 pixels. The CTB's can be partitioned further into Coding Units (CUs) using a quad-tree structure.

The smaller size partitions are required when more detailed predictions are needed, and larger size partitions provide higher coding efficiency. The Coding Unit contains one or several Prediction Units (PUs) and Transform Units (TUs).

Debugging is always a challenge for an encoder/decoder developer due to complexity of coding blocks structure. The HEVC video supports 4k and 8k resolution streams and it's almost impossible to debug the blocks when picture is portioned into numerous blocks. The partitions are the building blocks of any stream and the primary challenge for developers is to debug and optimize the block structure. A software tool that can allow quick browsing to any block in the picture and can provide both high-level and low-level details of the blocks, is a strong necessity for codec developers.

As shown below (Figure 1), the HEVC blocks are shown clearly and you can quickly navigate to any block for debugging all the encoded and decoded parameters. The parameters like bits used, QP, Motion Vectors, Prediction Modes, Reference Indexes and Interpolation Types are overlaid on the picture for quick debugging. The tabular form tooltip is designed to display all the critical parameters of a block at a single place, further reducing the debugging effort.

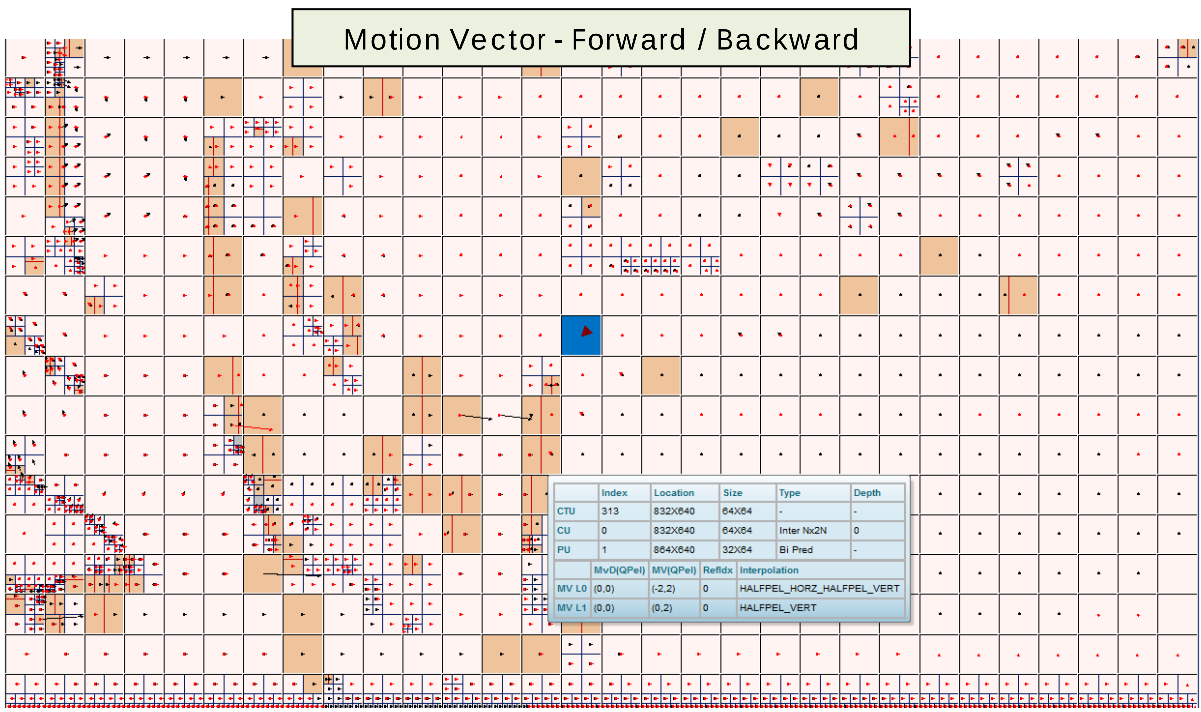


Figure - 1, Courtesy, Interra Systems

Intra-Prediction

Intra-prediction approach in HEVC is on the similar lines with H.264/AVC but with certain enhancements. In HEVC, it operates in accordance with CU size & samples are predicted from reconstructed samples of neighboring blocks. A significant change comes from the introduction of larger block sizes, where intra-prediction using one of 35 prediction modes (33 directional, a DC and a Planar mode) may be performed for blocks of size up to 32×32 samples as opposed to 9 prediction modes in H.264.

For Intra DC mode, the predicted block is filled with pixels obtained by averaging pixels as shown in Figure 2 below. The Intra DC mode is the least computationally-expensive mode. When using the Intra Planer mode, which is the most computationally-expensive mode, the encoder calculates a bidirectional interpolation function, which is used to fill the predicted block. The angular modes are shown in Figure 3.

With the increased number in prediction modes, analyzing and debugging intra prediction becomes a challenging task for codec developers. The challenge is to find out the left and top pixel values, how these pixels are getting used in intra prediction, what the modes present are and what are the final calculated pixels. All these challenges can be easily overcome by using the in-depth video analyzer as shown in Figure 2-3 below.

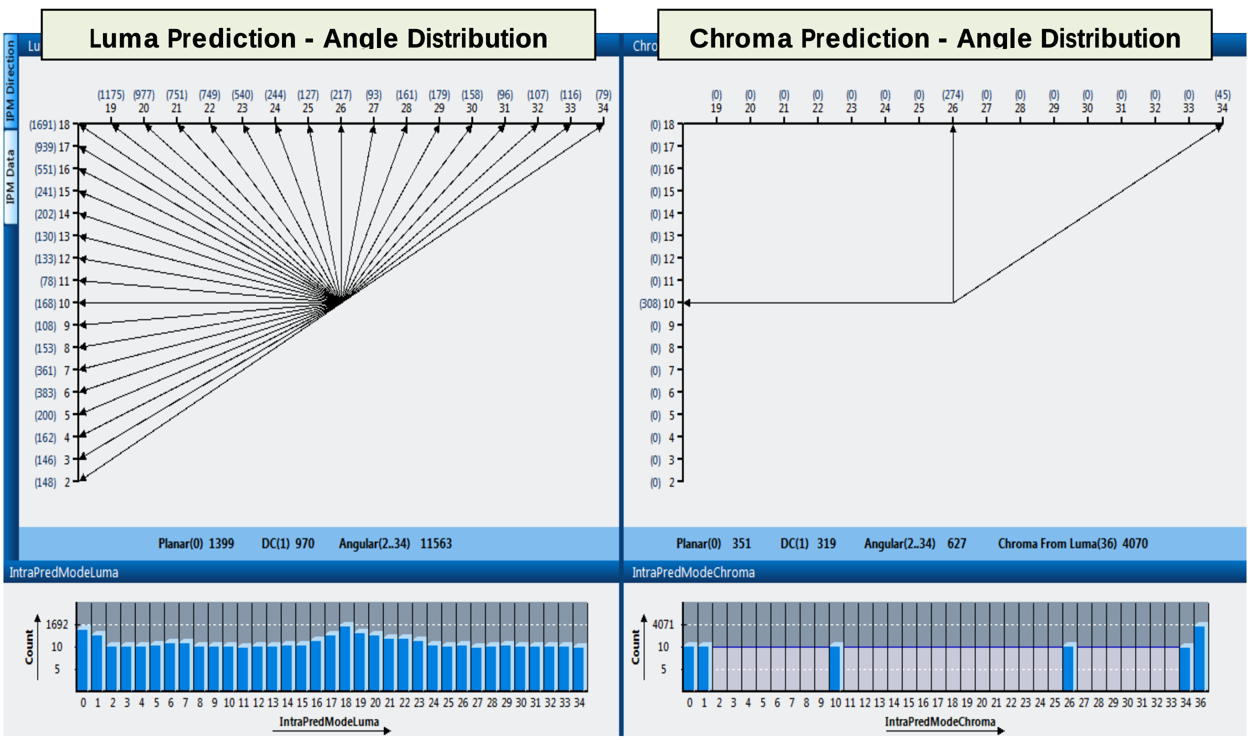


Figure - 2, IPM Direction View , Courtesy, Interra Systems

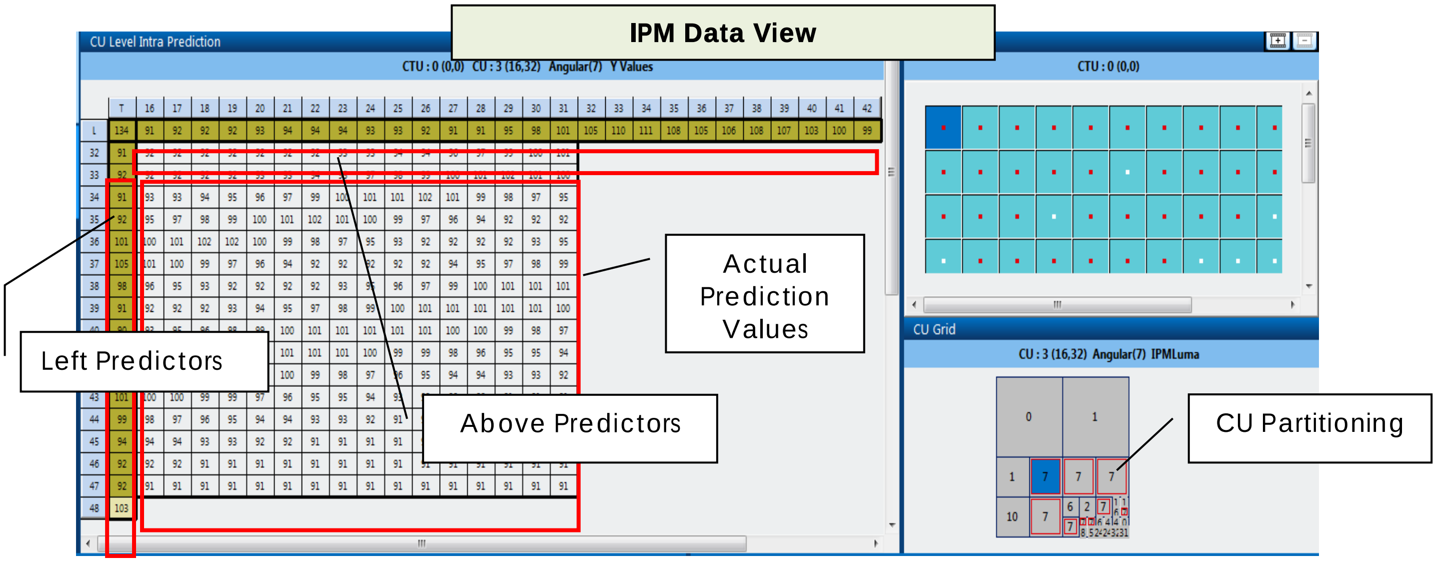


Figure - 3, Courtesy, Interra Systems

Motion Compensation

For inter prediction, similar to H.264, HEVC employs the block-based motion compensation (MC) with multiple reference pictures, but due to the advanced coding tools adopted, it is more complex compared to H.264. For example, asymmetric block partitioning adds more complexity to motion estimation search algorithms in HEVC which require more computational power.

Also, for sub-pixel interpolation, 8-tap DCT based interpolation filter is employed as compared to 6 tap in H.264 and for chroma component, 4-tap DCT-IF is applied as compared to bilinear filter in H.264 which leads to an increase in memory bandwidth and in the number of multiply-accumulate operations required for motion compensation.

Moreover, multiple motion vector predictors derived by the advanced motion vector prediction (AMVP) in HEVC increase the motion search candidates by many times. And new coding modes, such as **merge mode** (*which sets all motion parameters of an inter picture predicted block equal to the parameters of the merge candidate)* also increase the complexity of motion estimation and compensation greatly due to the cross reference of the motion information of the spatial and temporal neighboring PUs.

This creates a challenge for debugging since it is difficult to find out what motion vector is used, what are the neighboring blocks available and from where it is derived. Also, it's critical to find out the reference picture details and the displacement of the coded block from the previous position. All these challenges are very well addressed below in Figure - 4 and 5.

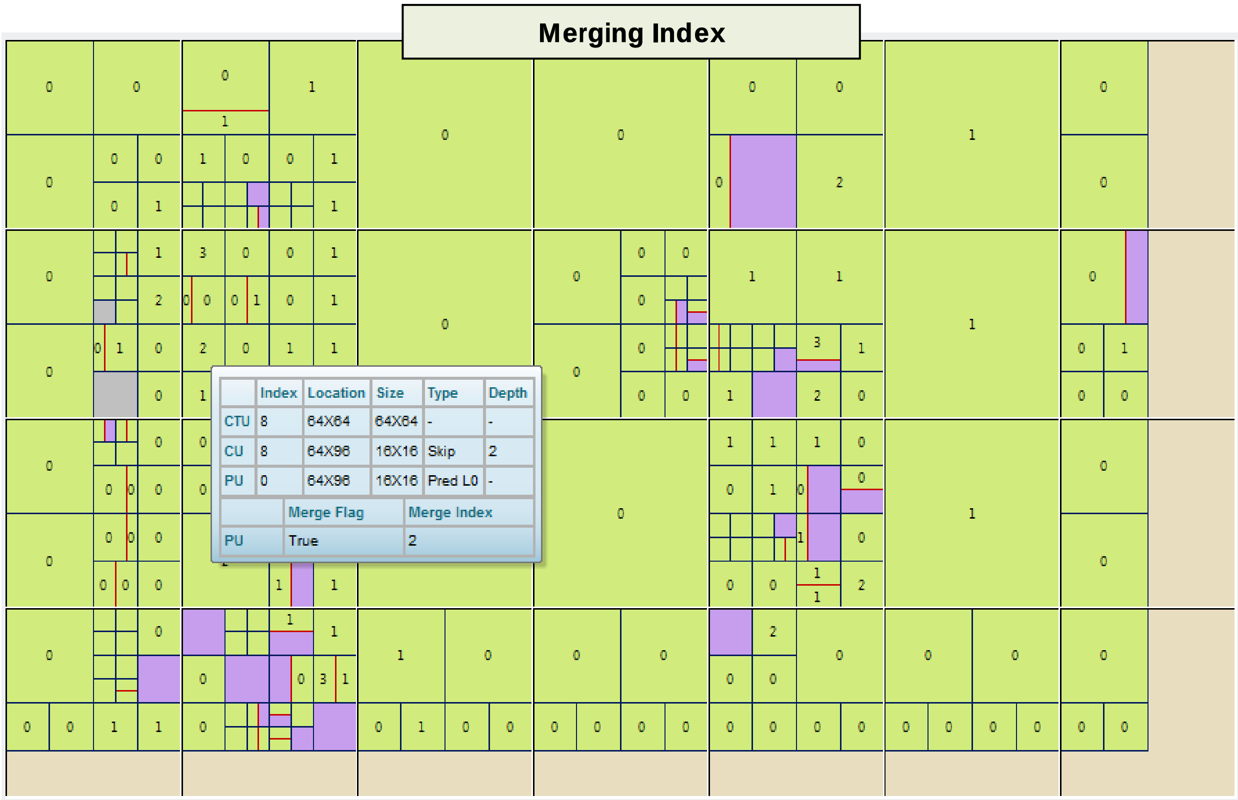


Figure - 4, Courtesy, Interra Systems

In Figure - 4, you can see the Prediction units (PUs) for which the Merge Mode is enabled and the Merge Index that is coded in the bit stream also overlaid on the picture.

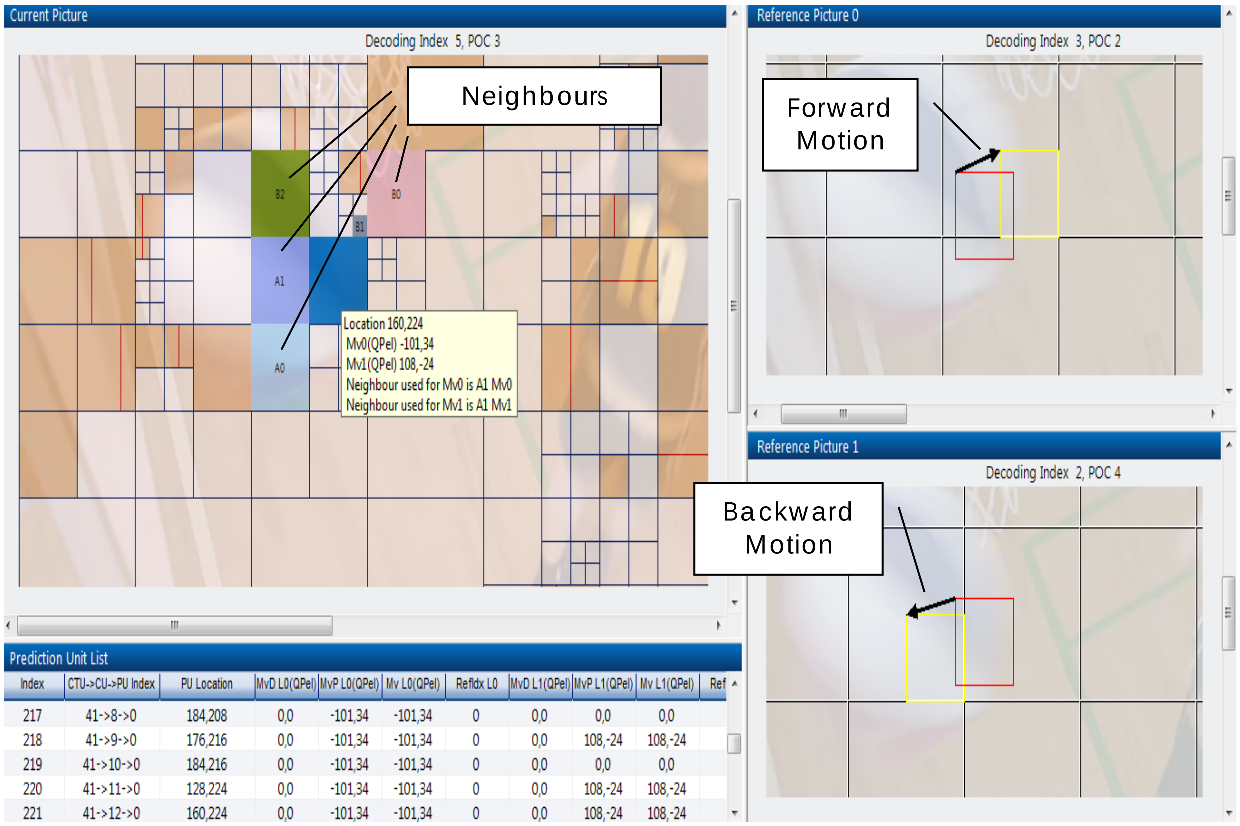


Figure - 5, Courtesy, Interra Systems

In Figure 5, you can easily debug all the PUs which form the merging list and also see which PU from the merging list is actually used for generating the Motion Vector.

Sample Adaptive Offset

HEVC adds a new filtering process over its predecessor H.264, i.e., Sample Adaptive Offset (SAO) filtering. The key idea of SAO is to reduce sample distortion by first classifying reconstructed samples into different categories, obtaining an offset for each category, and then adding the offset to each sample of the category. The offset of each category is properly calculated by the encoder and explicitly signaled to the decoder for reducing sample distortion effectively. To achieve low latency of only one coding tree unit (CTU), a CTU based syntax design is specified to adapt SAO parameters for each CTU.

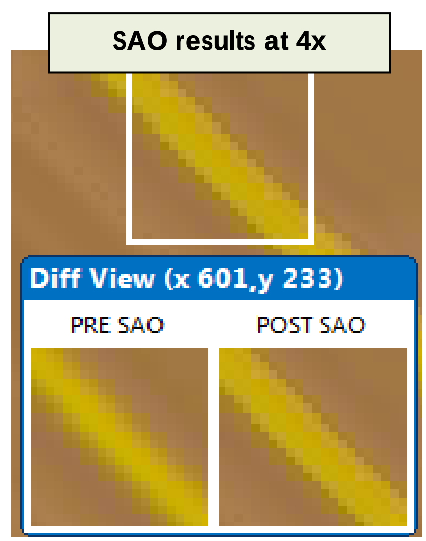


Figure - 6, Courtesy, Interra Systems

A good in-depth analyzer enables users to effectively analyze and debug the filtering process in the stream. For debugging developers need both AU level and pixel level details at a single place. HEVC reduces the redundancy in Offset information by sharing it within neighboring CTU. This is done with the help of two merge flags, top and left. This merging forms a long chain of CTU that are dependent upon each other for correct output of the picture, thus increasing the difficulty for debugging the output of this process. Because the offsets used in a particular CTU may not be encoded within the same and to debug the results, the user have to find the CTU from where the erroneous offset were decoded. This dependency information is shown in an easy to navigate form in the Figure 7 below.

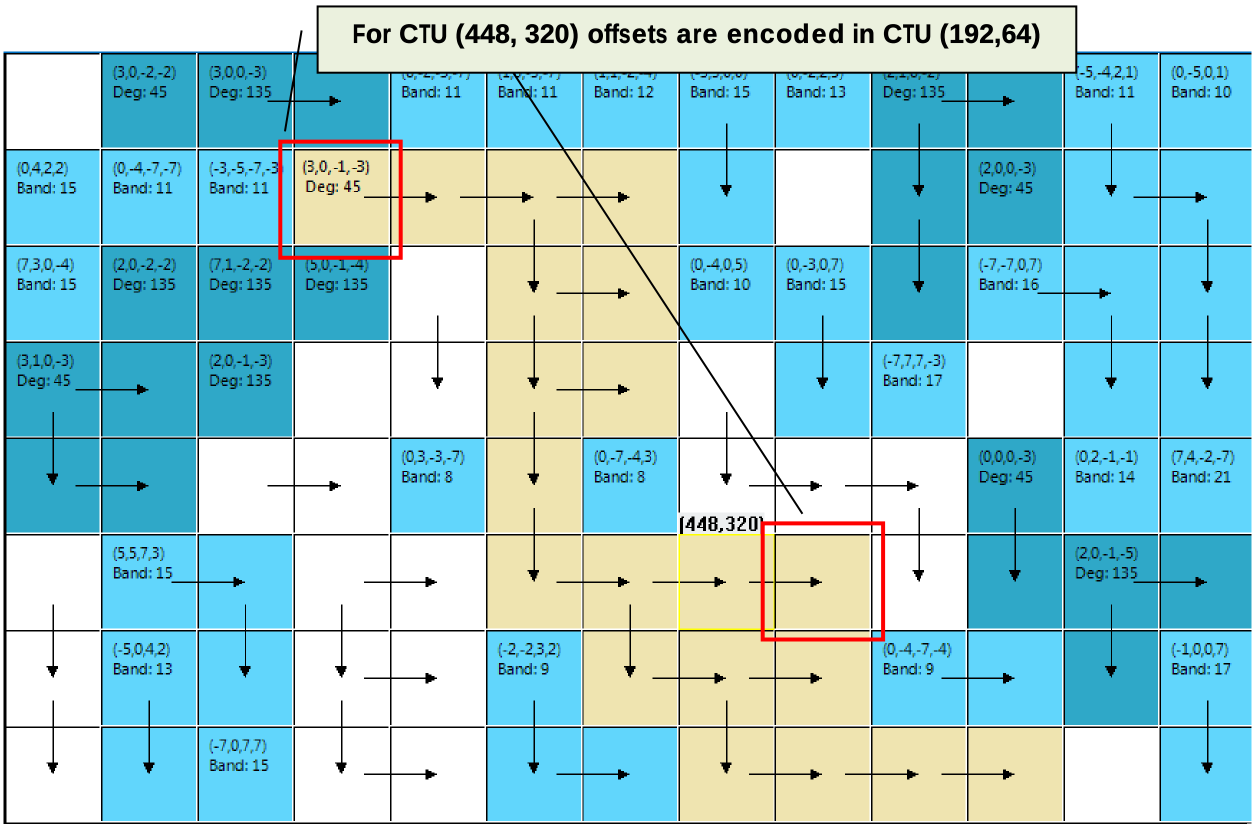


Figure - 7, Courtesy, Interra Systems

<u>Residual Parsing</u>

For achieving compression, codec developers always try to reduce the bits consumed by syntax elements. As Residual Data is the largest part in terms of bandwidth consumption in any video bit stream, the HEVC standard adds some methods for reducing the data that is encoded in the residues. One of the methods employed is to increase the Residual Block size which can be as large as 32X32. Another method is using the Discrete Sine Transform instead of Discrete Cosine Transform. Further, it allows various scanning modes for residues like horizontal, vertical and diagonal scanning.

In general, analyzing the Residual Data is quite difficult since it is very large and not clearly visible. Due to the above features introduced by HEVC, it becomes even tougher to analyze Residual Data. The Figure - 8 shows the Residual Data for each block, and makes it easy to quickly locate the decoded Residual Data and results in saving the development time.

In Figure - 8, you can see the different types of scanning orders with different colors on the left side grid of CUs and also on the right side, you can see the scanning order and values for the selected TUs.

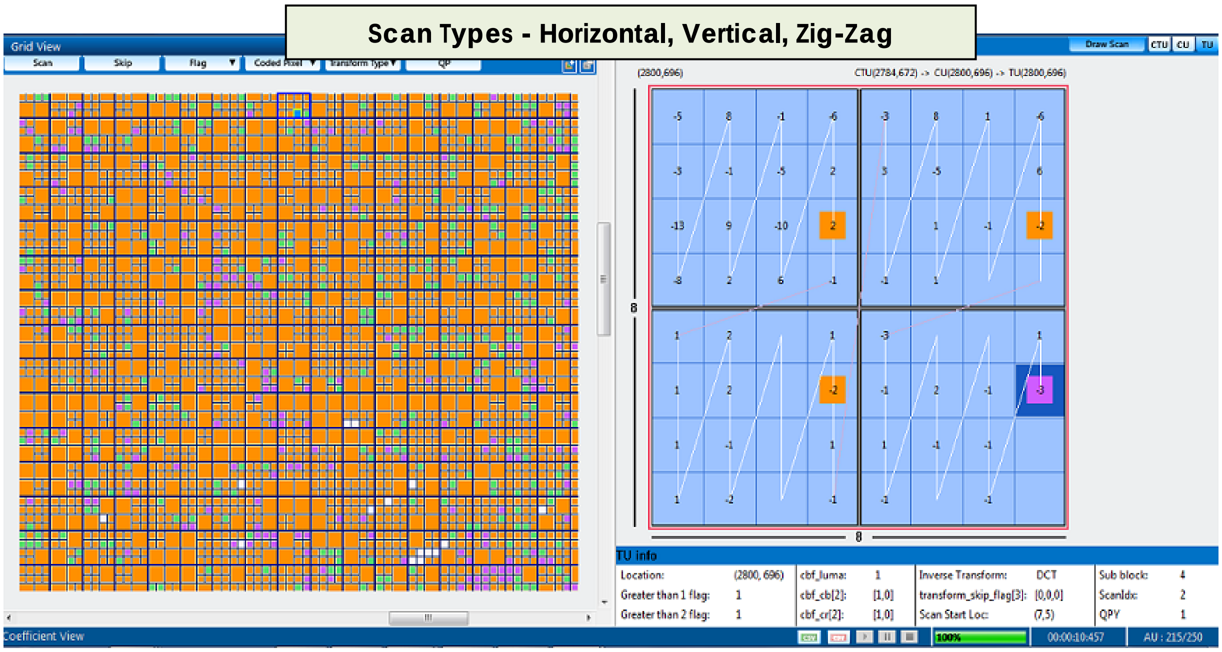


Figure - 8, Courtesy, Interra Systems

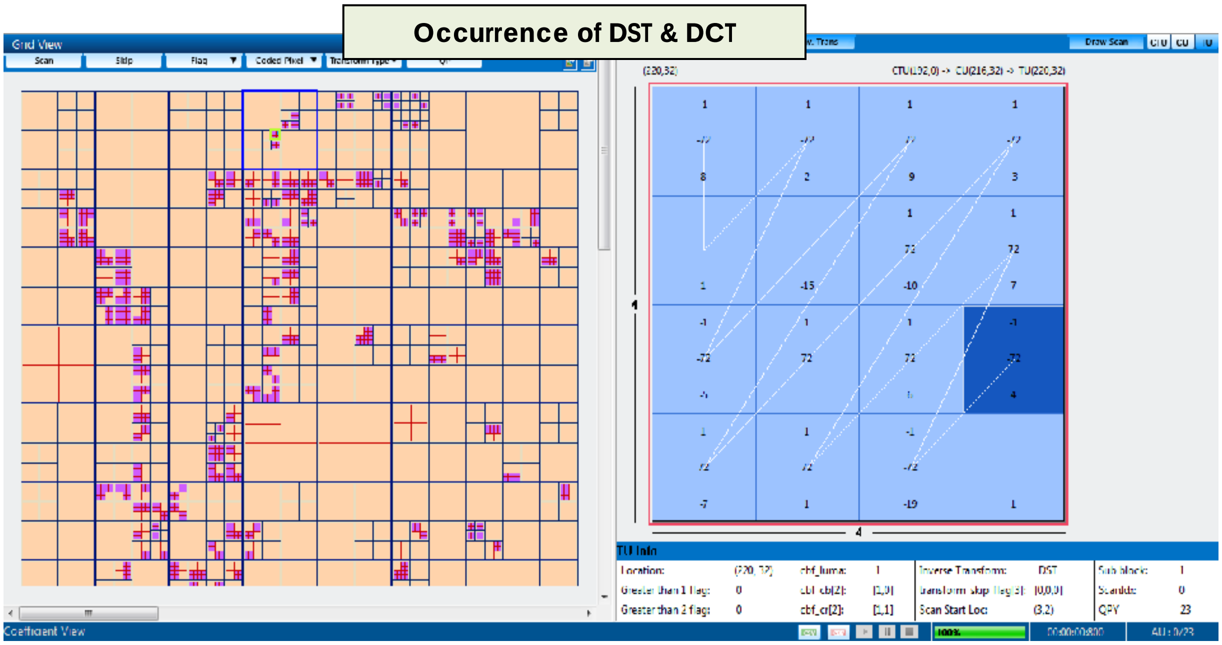


Figure - 9, Courtesy, Interra Systems

In Figure 9, you can see the DST Transform Blocks on the left side and the Residual, Inverse Transform and Scaled values on the right side for the selected TUs. So, the compression challenge can be easily understood and resolved by using the features mentioned above.

Hypothetical Reference Decoder

While applying the HRD model on the HEVC video, overflow or underflow could occur. If the buffer overflow happens during analysis, there could be loss of video frames and the buffer underflow may cause noisy video. This problem must be resolved at the development stage otherwise it can lead to critical issues in the video. The Figure - 10 shows how one can easily locate the overflows and underflows in the video stream. Also, the developer can tweek the HRD parameters to get the optimal parameters for a specific stream. The other critical thing is to know the state of all parameters used during buffer analysis. By knowing these states, the developer can debug and fix the issues at any particular point in the bit-stream. This detailed information is also provided in the Figure - 11. The features below are very useful and are the quickest in resolving the Buffer Analysis challenges.

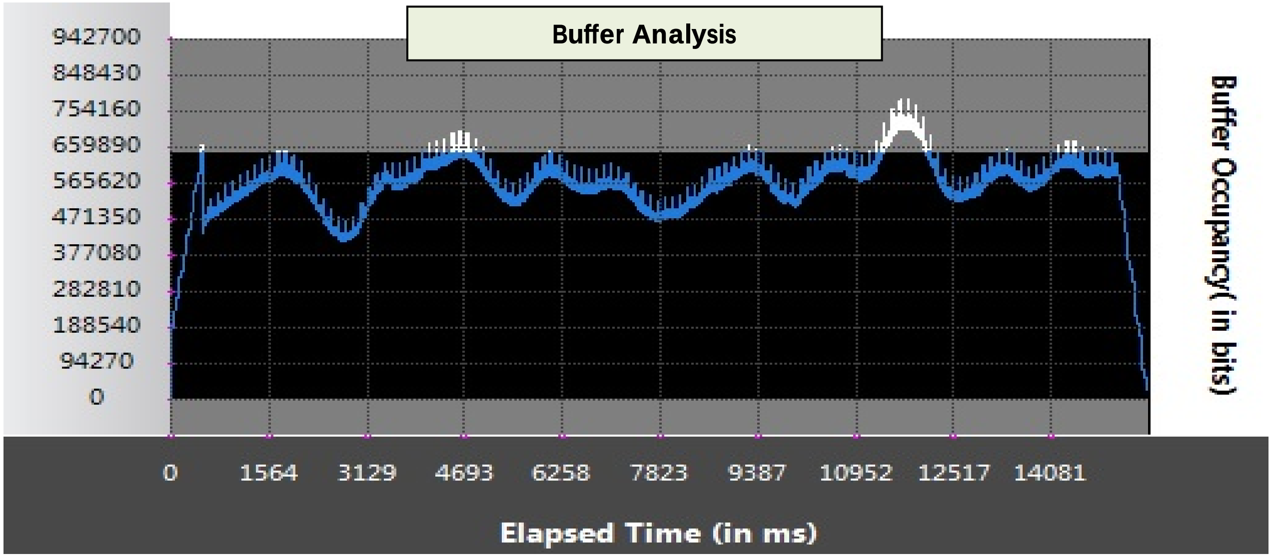
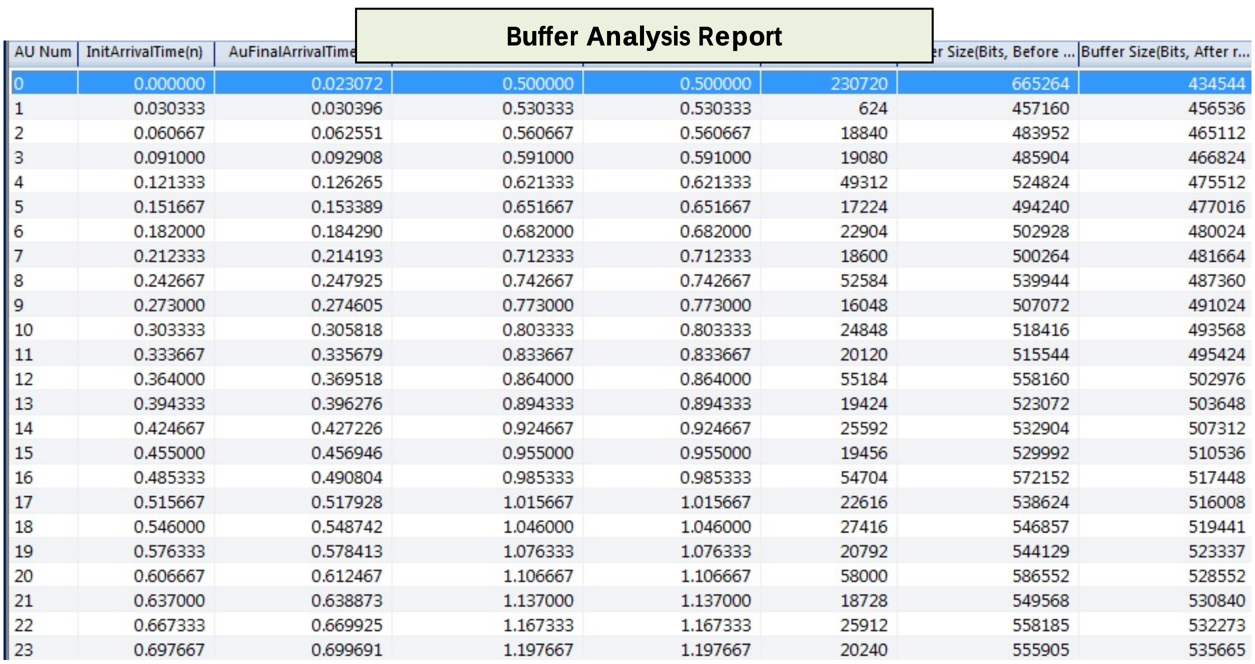


Figure - 10, Courtesy Interra Systems



Buffer Analysis Report

AU Num	InitArrivalTime(n)	AuFinalArrivalTime				r Size(Bits, Before ...	Buffer Size(Bits, After r...
0	0.000000	0.023072	0.500000	0.500000	230720	665264	434544
1	0.030333	0.030396	0.530333	0.530333	624	457160	456536
2	0.060667	0.062551	0.560667	0.560667	18840	483952	465112
3	0.091000	0.092908	0.591000	0.591000	19080	485904	466824
4	0.121333	0.126265	0.621333	0.621333	49312	524824	475512
5	0.151667	0.153389	0.651667	0.651667	17224	494240	477016
6	0.182000	0.184290	0.682000	0.682000	22904	502928	480024
7	0.212333	0.214193	0.712333	0.712333	18600	500264	481664
8	0.242667	0.247925	0.742667	0.742667	52584	539944	487360
9	0.273000	0.274605	0.773000	0.773000	16048	507072	491024
10	0.303333	0.305818	0.803333	0.803333	24848	518416	493568
11	0.333667	0.335679	0.833667	0.833667	20120	515544	495424
12	0.364000	0.369518	0.864000	0.864000	55184	558160	502976
13	0.394333	0.396276	0.894333	0.894333	19424	523072	503648
14	0.424667	0.427226	0.924667	0.924667	25592	532904	507312
15	0.455000	0.456946	0.955000	0.955000	19456	529992	510536
16	0.485333	0.490804	0.985333	0.985333	54704	572152	517448
17	0.515667	0.517928	1.015667	1.015667	22616	538624	516008
18	0.546000	0.548742	1.046000	1.046000	27416	546857	519441
19	0.576333	0.578413	1.076333	1.076333	20792	544129	523337
20	0.606667	0.612467	1.106667	1.106667	58000	586552	528552
21	0.637000	0.638873	1.137000	1.137000	18728	549568	530840
22	0.667333	0.669925	1.167333	1.167333	25912	558185	532273
23	0.697667	0.699691	1.197667	1.197667	20240	555905	535665

Figure - 11, Courtesy, Interra Systems

Quality Comparison

While optimizing encoder, the developer tries different coding tools and analyzes the rate distortions of each output bit-stream and chooses the best output according to the requirements. But to reach a conclusion of the acceptable algorithms and parameters, the developer needs to do an in-depth debugging of the stream and impact of each coding tool used on the quality of the bit-stream.

In general, analyzing the stream quality parameters is quite difficult since the output YUV data is very large and quality differences are not clearly visible. Due to this reason, one needs to identify the frames and blocks of the frame for quality changes and compare the same with the coding tools used in that area. For such extensive debugging, the developer needs a tool that can perform stream, frame and in-depth block level analysis. The Figure - 12 below shows the PSNR, SSIM and MSE comparison between two streams. The developer can visualize what frames have high and low distortion levels. Figure-13 shows similar comparison of the stream for visual quality index and Figure-14 shows contrast changes for each block. This makes it quite easy to overcome the Quality Analysis challenges in codec development as explained above.

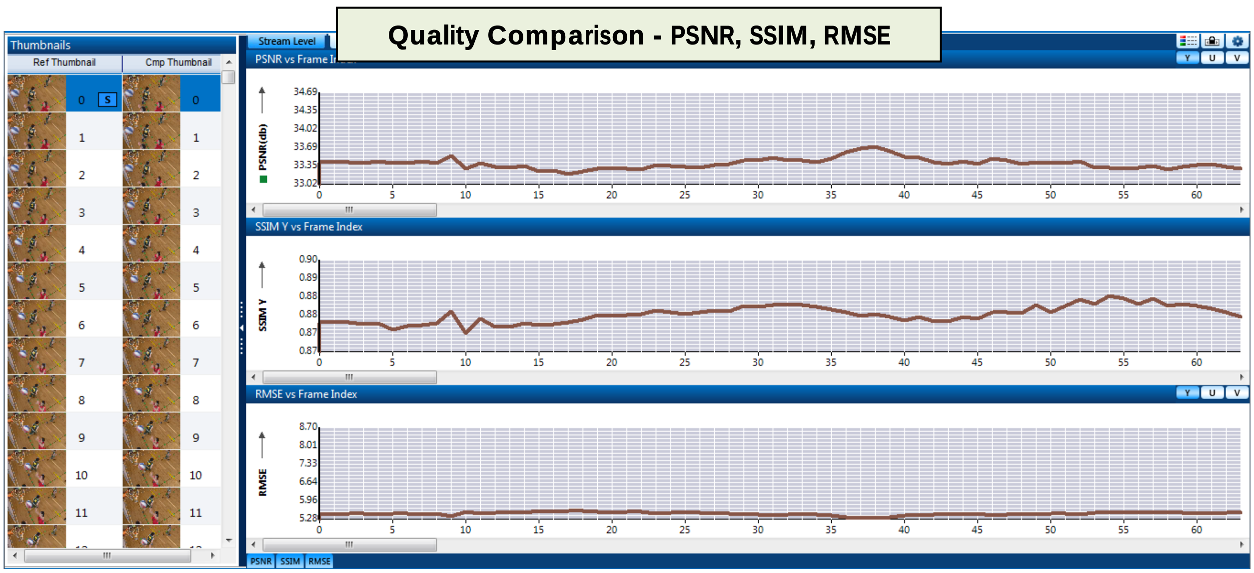


Figure - 12, Courtesy, Interra Systems

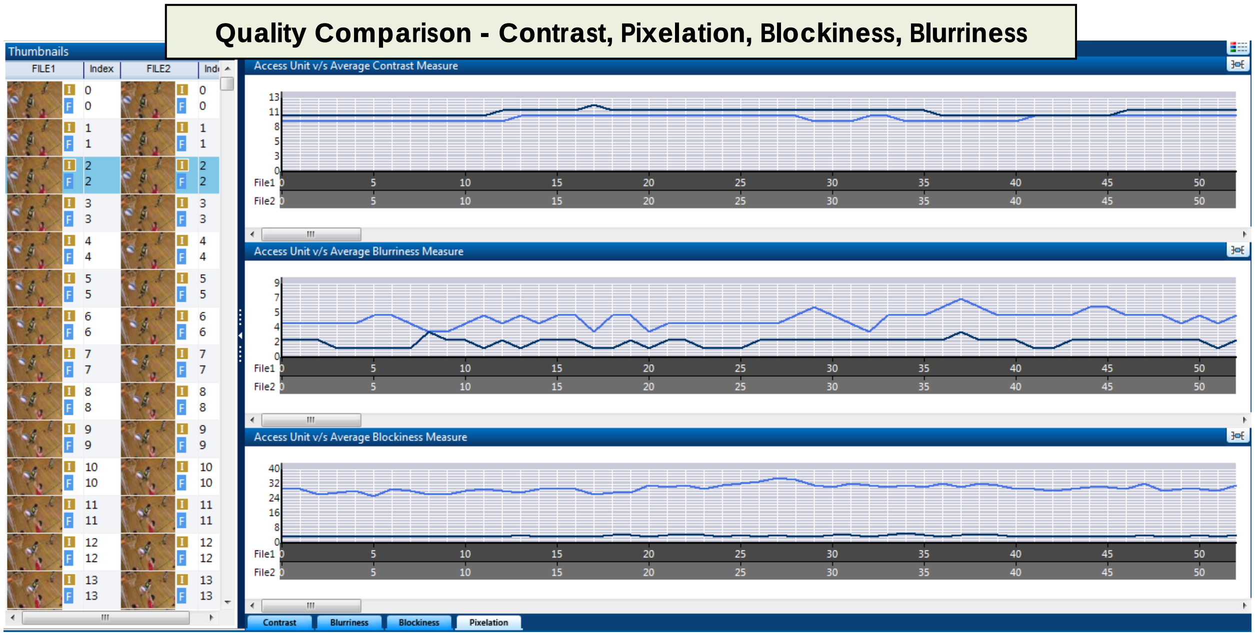


Figure -13, Courtesy, Interra Systems

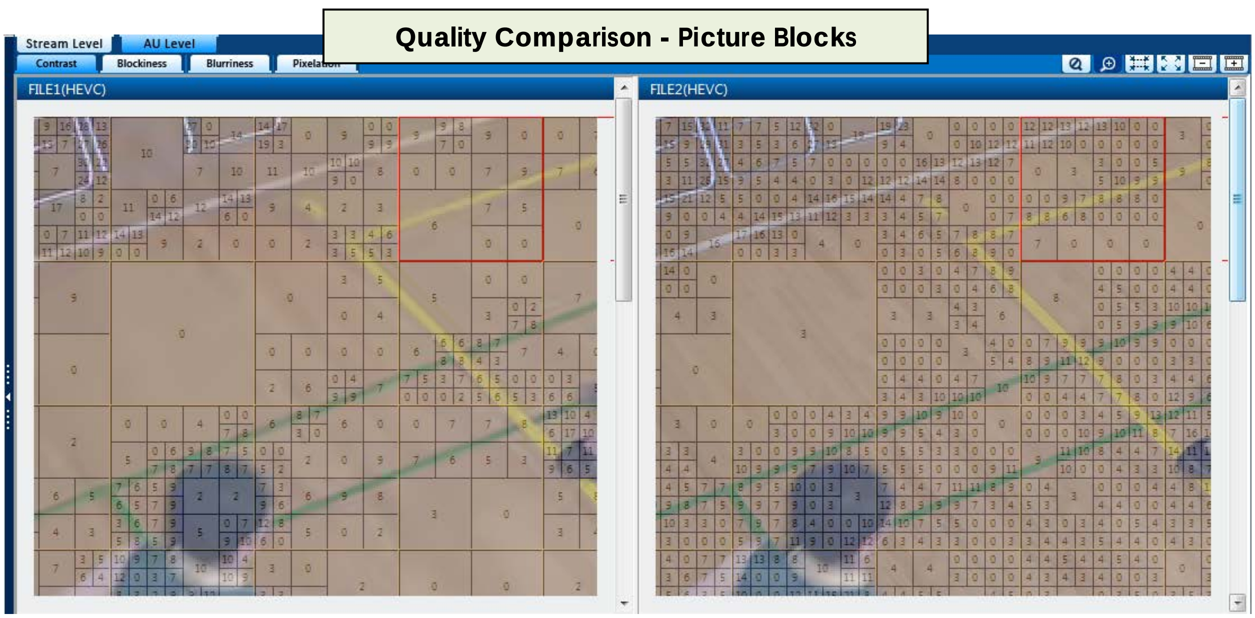


Figure -14, Courtesy, Interra Systems

4. SUMMARY

To summarize, HEVC is a complex video compression standard, requiring advanced analysis tools for development purposes. The right correlation of different picture elements coupled with good visualization and intuitive, easy-to-use presentation of information is critical for development efficiency. In addition, good analyzers must provide increased productivity tools like regression mode, batch analysis, data dumping, buffer analysis, quality metrics, stream comparison etc. to aid in the development of superior quality HEVC product.

5. ABOUT INTERRA SYSTEMS

Interra Systems provides software and services for the digital media industry. The company's solutions include Baton, an automated verification system that ensures media content readiness, VEGA, a family of audio/video analyzers that accelerate media product development and Orion, a real-time content monitoring solution. Interra Systems is headquartered in Cupertino, CA. For more information, please visit http://www.interrasystems.com.