

# HEVC DECISION OPTIMIZATION FOR LOW BANDWIDTH IN VIDEO CONFERENCING APPLICATIONS IN MOBILE ENVIRONMENTS

*Ray Garcia<sup>i</sup>, Member, IEEE, Damian Ruiz-Coll<sup>ii</sup>, Hari Kalva<sup>i</sup>, Senior Member, IEEE, and Gerardo Fernández-Escribano<sup>ii</sup>*

<sup>i</sup>Florida Atlantic University, Boca Raton, Florida, United States

<sup>ii</sup>Instituto de Investigación en Informática de Albacete, Universidad de Castilla-La Mancha, Albacete, Spain

## ABSTRACT

This paper presents a model and approach to select an efficient set of HEVC encoding options for mobile devices. The main goal is to reduce the encoding complexity without significantly affecting the quality of video conferencing applications. Video target bit rates from 100kbps to 600kbps were used within this study. Experimental results show that by carefully selecting the coding unit size, coding unit depth and to a lesser degree transform unit size, the encoder computational complexity can be reduced for the target bit rates while maintaining an allowable additional PSNR loss. Results show a 36.5% of complexity reduction on average with a negligible PSNR loss of less than 0.25 dB.

*Index Terms*— HEVC, Mobile, Bandwidth, Video Conference, Low Delay

## 1 INTRODUCTION

High Efficiency Video Coding (HEVC) is the latest video coding standard finalized in January 2013 by ISO and ITU's Joint Collaborative Team on Video Coding (JCT-VC). This paper's working environment is for HEVC video conferencing applications in the smart phone market segment with cellular bit rates, such as 200kbps and 400kbps. The primary focus is to reduce coding complexity, in terms of time savings, while maintaining high quality video.

The HEVC standard has adopted three profiles for wide services, such as broadcast, mobile communications and video streaming. Recent assessments show that HEVC can achieve equivalent subjective quality as H.264/MPEG-4 AVC with 50% less bit rate [1], by introducing new tools like the block partitioning structure [2], and variable block-size for prediction and transform coding [3] among others.

Undoubtedly, an effective tool included in HEVC is the new Coding Tree Block (CTB) partitioning structure. This can be thought of as a more generic version of Macro Block (MB) coding unit used in previous standards. The square CTB restricted to a maximum size of 64x64 pixels, which can be split into smaller quad-blocks called Coding Units (CU), with a minimum allowed size of 8x8. This means the maximum allowed CU Depth is 4.

The transform coding also uses a quadtree structure called Residual QuadTree (RQT), splitting a Transform Unit (TU) into smaller transform units, and limiting its size to a maximum size of 32x32 and minimum of 4x4. A 2Nx2N CU can use a maximum TU depth of 3, taking 2Nx2N, NxN and N/2xN/2 sizes.

HEVC encoder and decoder complexity assessment is a research topic for [4] – [7], where the different HEVC tools are analyzed in terms of performance and computational complexity.

As with H.264, HEVC uses a well-known Rate Distortion Optimization model (RDO) [8] to achieve the best coding efficiency. RDO reaches the optimal partitioning by evaluating all CU sizes, prediction unit (PU), and TU size for each of those CU-PU combinations. This approach greatly increases computational complexity of encoders. So, this makes it more difficult for real time encoding implementations, especially for portable and mobile devices where power consumption is one of the key factors. To reduce the RDO complexity some fast algorithms have been proposed, which focus on reducing the number of coding blocks (CB), prediction blocks (PB), and TU sizes to evaluate. [9] proposed a coding tree pruning based on SKIP mode detection. If the SKIP mode is selected as the best prediction mode then no further processing of smaller CB sizes is performed. This approach results in encoding time reduction of approximately 42% with a negligible loss of bit rate (< 0.6%) for equal PSNR. The proposed method by [10] is a fast RDO algorithm based on two techniques: (1) Stop Skip, which selects the initial CU size and (2) Early Termination, which limits the smaller CB sizes. This reduces the encoding time by 40% with a bit rate increase of 2%.

---

This work was supported in part by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under grants TIN2009-14475-C04 and TIN2012-38341-C04-04.

Schwartz et. al., show the effects of constraining the Largest Coding Unit (LCU) for Random Access (RA) and Low delay (LD) configurations of main profile [11]. Decreasing the LCU from 64x64 to 32x32, will realize a 18% and 17% time savings with a small bit rate penalty of 2.2% and 3.7% respectively. On the other hand, using a LCU of 16x16 instead 64x64, reduces encoding time by 42% but with a bit rate increase of 11% and 17.4% respectively. Also reducing the maximum RQT depth from 3 to 2 yields a time saving of 10% with a slight bit rate increase of 0.3% for RA and 0.4% for LD configurations.

Finally, a complexity control algorithm based on coding tree depth selection for CTB, is proposed in [12] which obtains a computational complexity reduction of 40% with a bit rate increase of 3.5%. For [13], a maximum CU-depth dynamic adjustment is proposed, which testing showed a 40% complexity reduction, 0.1dB PSNR, and a 3% bit rate increase.

## 2 METHOD

### 2.1 Setup

The mobile device working environment, within this research, was based on multimedia streaming guidelines and content providers. The content is defined as “higher quality” for 640x360 resolution at 400kbps and “medium quality” for 400x300 resolution at 200kbps [14] [15]. We chose to have a superset of target bit rates, from the recommendations, in order to observe the video quality over a range of bitrates. The bit rates within this study are 100, 200, 300, 400, and 600 kbps. Also, we decided to have a several video resolutions. The lower resolutions made available by JCT-VC and defined in [16] were used. HEVC video sequences chosen for the experiments were from Class C, D, and E as defined by JCT-VC. Video sequences used for the experiments are defined in Table 1.

Table 1 : Video sequence definition

Class	Video Sequence	Resolution	Fps	f
C	BasketballDrill	832x480	50	500
C	BQMall	832x480	60	600
C	PartyScene	832x480	50	500
C	RaceHorses	832x480	30	300
D	BasketballPass	416x240	50	500
D	BlowingBubbles	416x240	50	500
D	BQSquare	416x240	60	600
D	RaceHorses	416x240	30	300
E	FourPeople	1280x720	60	600
E	Johnny	1280x720	60	600
E	KristenAndSara	1280x720	60	600

The study’s experiments used HEVC encoder reference software version HM8.0 [17] with Main Profile and Low-delay configuration as the baseline configuration. This

configuration was meant for real-time personal video communications which uses the previous frame for motion estimation. The common JCT-VC test conditions described in [18] were used with the aim of analyzing the HEVC performance. The input variables “Bit Rate”, “Largest CU (LCU) size”, “CU-depth”, and “TU-inter-depth” were modified between tests.

The HM8.0 source code was modified to output additional CU and TU data that was used for analysis. The collected data included CU Mode (i.e. Intra or Inter), LCU, CU-depth, TU-inter-depth for each CU block within all frames for the video sequences. A total of 1110 video sequence experiments were run during the study. This data was used for analysis. Allowances were made to run experiments over multiple computers. However, each video sequence experiment occurred on same computer to eliminate any comparison disconnects.

Each sequence was encoded with a baseline configuration (all options on) and additional configurations where for a given bit rate, the LCU, CU-depth, and TU-inter-depth were varied. All video sequences and input variable combinations are as follows:

- LCU: 64x64, 32x32 and 16x16
- CU-depth: 4, 3, 2, 1
- TU-inter-depth: 3, 2, 1
- Bit Rate: 100, 200, 300, 400, and 600 kbps

All videos were 10 seconds long with frame rates varying among 30, 50, and 60 FPS. Each frame had 66 data elements tracked. This yielded a total amount of over 37 million data elements that required reviewing and analyzing. The key outputs observed for performance analysis were the PSNR difference, identified as  $\Delta$  PSNR, and encoding time difference, identified as Time-savings, between the baseline configuration and each of the experimental configurations.

### 2.2 Prediction Modeling

To assist with data analysis and create a prediction model, the Waikato Environment for Knowledge Analysis (WEKA) Version 3.6.4 [19] was used for data mining results from the experiments. WEKA is an effective software tool used for sifting through large data sets and determining relevant data for the chosen prediction model.

The datasets used for prediction modeling training were chosen from one dataset per class. All remaining video sequence datasets were used for testing. From these experiments, the Time-savings and  $\Delta$  PSNR were recorded and used for the desired output to determine equation. More details on Time-savings and  $\Delta$  PSNR later within this report.

The WEKA software tool’s analysis classifier was set to “linear regression” for prediction and attribute selection method selected to “No attribute selection”, which allows all inputs to be potentially used for the modeling equation. Therefore, no pruning attempts were made to simplify the equations. Attributes used for prediction model includes:

resolution, bit rate, LCU, CU-depth, and TU-inter-depth. Attributes, such as “number of frames” and “frame rate” were removed from the attribute pool. These attributes are irrelevant for the calculations. The remaining tool configurations were left at default settings.

### 3 RESULTS

Data was gathered for all video sequences. All the data gathered within the study cannot be addressed or presented in this report. However, representative examples were used to explain underlying concepts.

#### 3.1 Data Reading Primer

For the rest of this document, “LCU” / “CU-depth” / “TU-inter-depth” will be shown as numbers separated by the slash (i.e. /). Therefore, 64/4/3 is representation for LCU = 64x64, CU-depth = 4, TU-inter-depth = 3.

The graph shown in Figure 1 is the “Time-savings vs.  $\Delta$  PSNR” for “KristenAndSara” video sequence. The graph is for bit rate of 100kbps. Each point is defined by “LCU”/“CU-depth”/“TU-inter-depth”. The curve within the graph is the outline curve for all points. The line represents the effective trade-off between  $\Delta$  PSNR and Time-savings. The closer the points are to the curve the better trade-off decision.

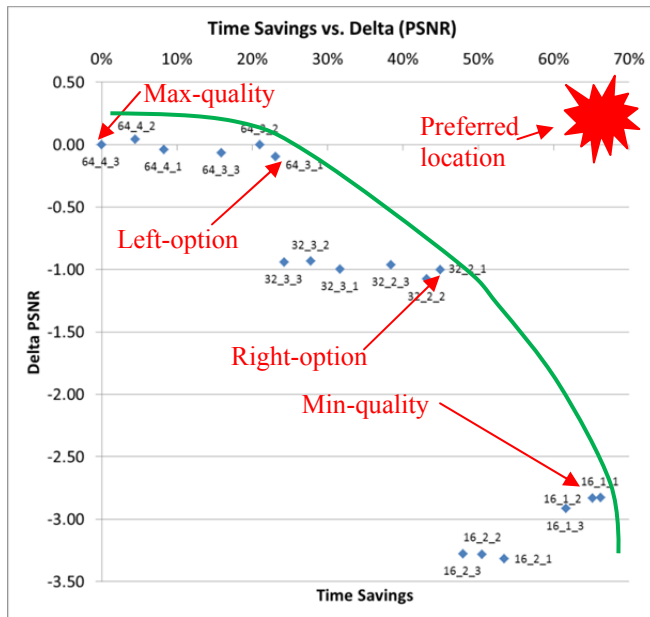


Figure 1: KristenAndSara 100kbps Time-savings vs.  $\Delta$  PSNR.

The most beneficial location is the upper right corner, which has maximum Time-savings and minimum  $\Delta$  PSNR loss. This area is defined in the graph as “Preferred location”. Within the data set, the best  $\Delta$  PSNR loss is 0 dB

and is held by the 64/4/3 data point. This data point is shown with arrow in upper left corner and is identified as Max-quality. As expected this encoded sequence has the longest encoding time, which leads to 0% Time-savings. Moving along the outline curve there are two points which are defined as “Left-option” point and “Right-option” point. These points are the optimal points as determined by us. Perceptual video quality assessment was carried out. The assessment consisted of subjectively reviewing each sequence to determine the preferred video sequence. In instances where there the subjective review returned the same result, the selection went to the sequence that yielded a higher  $\Delta$  PSNR x Time-savings value. From this effort, the optimal points mainly consisted of the minimum CU-depth and TU-inter-depth for LCU of 64x64 and 32x32. The majority of the optimal cases were clearly these points. We decided to simplify the option point’s selection and set it to the minimal configuration for LCU of 64x64 and 32x32. The data point on the lower right has the most restrictive settings. This point is identified as Min-quality. As expected, this point challenges for the best Time-savings and largest  $\Delta$  PSNR loss. There are frequent instances where this may not be the case as shown in Figure 1.

#### 3.2 Data Analysis

The graph (Figure 1) is a typical representation for the data gathered from the video sequences within this study. The Left-option point’s Time-savings is approximately 25% with  $\Delta$  PSNR of -0.07dB. The Right-option point has Time-savings of approximately 45% with  $\Delta$  PSNR of -1.03dB.

The Left-option and Right-option points vary depending on resolution. For 832 x 480 (Class C), the two points are 64/2/1 and 32/1/1. For 1280x720 (Class E) and 416x240 (Class D), the two points are 64/3/1 and 32/2/1. Table 2 shows the  $\Delta$  PSNR and Time-savings for Class C, D, and E. The C, D, and E average is shown as “All”. Further table explanations are shown later within this section.

Table 2: Class C, D, E, and All video sequence, average  $\Delta$  PSNR and Time-savings.

		Max.	Left-option	Right-option	Min.
C	$\Delta$ PSNR (dB)	0	-0.31	-0.59	-1.36
	Time-savings (%)	0.0%	41.8%	58.1%	55.6%
D	$\Delta$ PSNR (dB)	0	-0.21	-0.37	-0.7
	Time-savings (%)	0.0%	24.2%	36.4%	56.4%
E	$\Delta$ PSNR (dB)	0	-0.07	-1.03	-3.26
	Time-savings (%)	0.0%	23.6%	45.6%	67.2%
All	$\Delta$ PSNR (dB)	0	-0.25	-0.72	-1.82
	Time-savings (%)	0.0%	36.5%	55.4%	69.2%

Video sequences that match the points from Figure 1 were evaluated by us. A frame for each graph point is shown in Figure 2 and Figure 3. The example chosen is the KristenAndSara from Class E video sequence. The Max-quality video, which is the reference video sequence, is

impaired and loss of detail is observed in Figure 2 left side picture, which is 64/4/3. However, when compared to the Left-option point, which is 64/3/1 and on the right side, the degradation in the picture is minimal. The Right-option point to compare with is Figure 3 left side picture, which is 32/2/1. More degradation is noticed when compared to the reference (i.e. Max-quality) video sequence. Depending on user and environment needs, this may acceptable and will yield additional encoding time savings.



Figure 2: KristenAndSara sequence, 1280x720, 100kbps, Video frames for Max-quality and Left-option points.



Figure 3: KristenAndSara sequence, 1280x720, 100kbps. Video frames for Right-option and Min-quality points.

Table 2 shows the averages for  $\Delta$  PSNR and Time-savings. As noted earlier, the “64/4/3” data point has  $\Delta$  PSNR of 0 dB and Time-savings of 0%. This reference point represents the Max-quality option. All graphs have the normalization performed for consistency and clarity in data analysis.

As expected, the PSNR loss increased as the encoder options were reduced. Basically, encoder flexibility was limited by reducing LCU, CU-depth, or TU-inter-depth options.

Major contributors that affect Time-savings are either LCU or CU-depth depending on resolution. CU-depth change has more of an impact for smaller resolutions. LCU change has a bigger impact for larger resolutions. This observation is noticed when the test results for each sequence are ordered by Time-savings. Smaller resolutions order lined up according to CU-depth selections, and larger resolutions ordering lined up in accordance to LCU selections. This is an observation and detail is not shown within this report.

For a given video sequence, the effect of bitrate on Time-savings and  $\Delta$  PSNR is shown in Figure 4 and Table 3. As expected, the reference video sequence (i.e. Max-quality) absolute PSNR improves. Tool options changes have a greater impact on  $\Delta$  PSNR. All video sequences, in the study, had the similar shift.

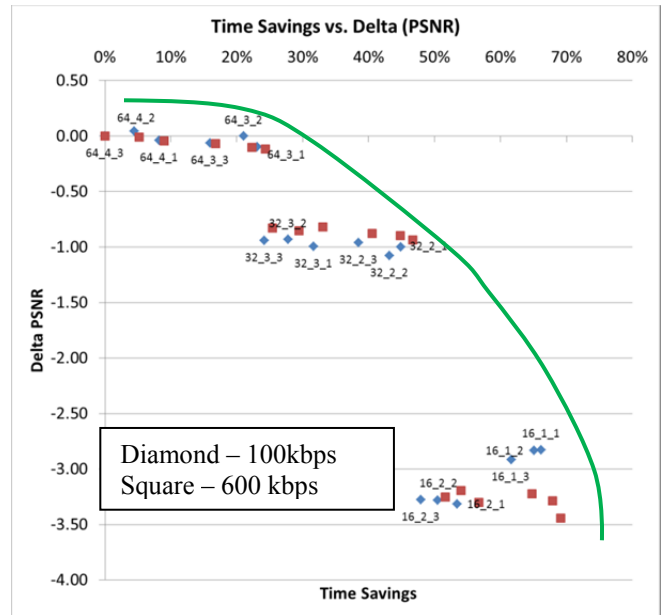


Figure 4: KristenAndSara 100kbps to 600kbps shift for Time-savings vs.  $\Delta$  PSNR.

Table 3: KristenAndSara 64/4/3 PSNR and  $\Delta$  PSNR for 100 and 600 kbps.

	64/4/3	$\Delta$ PSNR			
	PSNR	Max.	Left Opt.	Right Opt.	Min.
100 kbps	27.35	0	-0.10	-1.00	-2.83
600 kbps	33.63	0	-0.12	-0.93	-3.44

### 3.3 Prediction Model Analysis

The data analysis showed the video sequence data has a fairly good relationship to each other when the data is sorted by resolution size. However, all resolutions combined, as shown by Table 2 (rows “All”), will be the comparison point for the predictive model explored in this paper. There are content dependencies not evaluated within this paper and it is noted within this section.

From the data set gathered from the experiments, a data mining exercise took place using WEKA Version 3.6.4 in order to generate a prediction model. This could potentially assist with the selection of the optimal settings to maximize Time-savings and stay within the allowable  $\Delta$  PSNR for the given user defined environment. The prediction model equations are shown in Equation 1 and Equation 2. Equation discussion will be presented in the form of tables and graphs that correlate with the results.

Equation 1 :  $\Delta$  PSNR  
 $\Delta\text{PSNR} = (A * (\text{Height}) + B * (\text{bit rate}) + C * (\text{LCU}) + D * (\text{CU-depth}) + E * (\text{TU-inter-depth}) + F) / G$   
 Where:  $A = -2132.73, B = -316.28, C = 29834.18,$   
 $D = 156664.03, E = 3339.60, F = -1261163.72,$   
 $G = 1,000,000$

Equation 2 : Time-savings  
 $\text{Time-savings} = (H * (\text{Height}) + I * (\text{bit rate}) - J * (\text{LCU}) - K * (\text{CU-depth}) - L * (\text{TU-inter-depth}) + M) / G$   
 Where:  $H = -1.52, I = 41.48, J = -769.43, K = -203990.83,$   
 $L = -31218.52, M = 949433.69, G = 1,000,000$

Using the WEKA derived equations; the Left-option and Right-option points are valid as shown in Figure 5. The arrows identify the same points from Figure 1. The Time-savings from the WEKA graphs are accurate representation. However, the  $\Delta$ PSNR has more variability when compare to Figure 1, but the PSNR points relative to one another within the WEKA graph is representative. To get better correlation to actual data, the WEKA generated  $\Delta$ PSNR and Time-savings graph is normalized by shifting 64/4/3 data point to 0,0. All other points are shifted by same amount. Given this, the shape of the points within the WEKA model can be used to determine the optimal points.

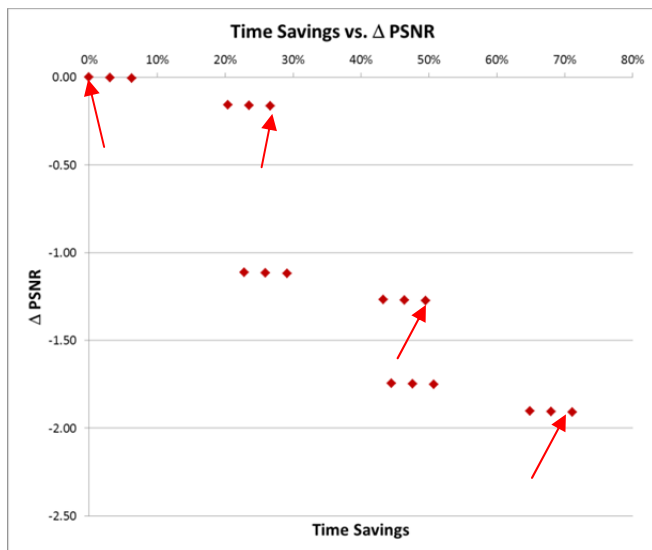


Figure 5: KristenAndSara 100kbps Time-savings vs  $\Delta$  PSNR WEKA derived. 64/4/3 is normalized to 0,0.

From the WEKA derived equations, video sequences  $\Delta$  PSNR and Time-savings were calculated and averaged for all classes. Results are shown in Table 4. The results in Table 4 have been normalized to Max-quality data point (i.e. 64/4/3).

Table 4: Video sequences WEKA estimate for  $\Delta$  PSNR and Time-savings. Max. (64/4/3 point) is normalized.

	Max.	Left-option	Right-option	Min.
$\Delta$ PSNR	0.00	-0.22	-1.33	-1.91
Time-savings (%)	0.0%	34.1%	56.9%	71.1%

Comparison between actual values and WEKA derived results are shown in Table 5. Results from Table 2 will be referred to as “Actual” and results from Table 4 will be referred to as “WEKA”.

An “Actual” versus “WEKA”  $\Delta$  PSNR differences comparison shows a difference no greater than 0.61 dB for the optimal points, as shown in Table 5. This shows there are very good estimate results by using WEKA. Also, the Time-savings comparison between “Actual” and “WEKA” shows better results as shown in Table 5. The worst case for the optimal points for any Class is a 2.4% difference.

The correlation between “Actual” and “WEKA” values were derived and the results are shown in Table 6.

This data shows that the WEKA derived data has an excellent correlation for encoding time savings. Therefore, by using the prediction model the developer can confidently estimate the encoding time savings changes by tweaking LCU, CU-depth, and TU-inter-depth to suit the user and environment needs.

$\Delta$  PSNR correlation is good with a correlation of 0.77 or higher. This indicates the WEKA model is a fairly good indicator for  $\Delta$  PSNR. The  $\Delta$  PSNR variation is mainly due to bit rate changes and video content dependencies.

By using the derived models, the developer can determine the impact of the video encoding changes could cause to the video quality via PSNR and the encoding time savings. In addition, the developer can change the model during the video encoding to use additional parameters derived from the previous frames. By using this additional real-time data, the model’s quality will improve and yield more accurate results. The correlation improves to 0.95 for  $\Delta$  PSNR and 0.99 for encoding time savings, when block types of previous frames are used within the prediction model.

Table 5: Video sequences Actual vs. WEKA difference.

	Max.	Left-option	Right-option	Min.
$\Delta$ PSNR diff.	0.00	-0.03	0.61	0.09
Time-savings diff.	0.0%	-2.4%	1.5%	1.9%

Table 6: Correlation between actual and WEKA results.

	Correlation
$\Delta$ PSNR	0.77
Time-savings	0.98

#### 4 APPLICATION

With the concepts within this study, the developer can estimate the PSNR with option changes. A no reference PSNR estimator, as presented by [20], can be used as a “coarse” PSNR estimate. The  $\Delta$  PSNR, calculated by the prediction model, will “fine tune” the absolute PSNR value for the projected video sequence. The developer can make informed decisions for video quality for transmitted video sequence. Also, using the prediction model, the developer can estimate how much the encoders will be taxed in terms of encoding time. Please note that each encoder implementation will derive different formulas and the method defined within this study will need to be re-established to determine the systems performance. This will only need to be characterized once for the H/W and S/W setup of the system.

#### 5 CONCLUSION

Using predictive modeling software, such as WEKA, can be effectively used to estimate  $\Delta$  PSNR and encoding time savings (i.e Time-savings) changes from the base, which is encoded video that has all the desired tool options enabled. With this data, the developer can effectively manage video encoding for the user and environment, such as transmission bandwidth limitations. Video conferencing environments will benefit greatly from the modeling by giving the developer a reliable method to estimate PSNR loss changes and encoding time savings gains. This will allow the developer to maximize the video quality (based on PSNR loss estimate) and finely tune to the available bandwidth (by using the encoding time savings estimate). Also, the model derived within the study has applicability for content independent use, especially useful when the prediction model uses real time data, such as block types used in previous frames, as part of the model.

#### 6 REFERENCES

- [1] Ohm, J.; Sullivan, G.J.; Schwarz, H.; Thiw Keng Tan; Wiegand, T., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol.22, no.12, pp.1669,1684, Dec. 2012
- [2] Il-Koo Kim; Junghye Min; Lee, T.; Woo-Jin Han; JeongHoon Park, "Block Partitioning Structure in the HEVC Standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol.22, no.12, pp.1697,1706, Dec. 2012
- [3] G. J. Sullivan and J.-R. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)", *SPIE Applications of Digital Image Processing XXXIII, Vol:7798*, pp:1-7, August2010
- [4] G. Correa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consumer Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.
- [5] On Software Complexity, *JCTVC-G757, ISO/IEC-JTC1/SC29/WG11*, Geneva, Switzerland, 2011.
- [6] M. Viitanen, J. Vanne, T. D. Hämäläinen, M. Gabbouj, and J. Lainema, "Complexity analysis of next-generation HEVC decoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 882–885.
- [7] T. Anselmo and D. Alfonso, HM Decoder Complexity Assessment on ARM, document *JCTVC-G262, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC)*, Geneva, Switzerland, Nov.2011.
- [8] L. Xiang, M. Wien, and J. R. Ohm, "Rate-complexity-distortion optimization for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 7, pp. 957–970, Jul. 2011.
- [9] JCT-VC, "Coding tree pruning based CU early termination", *JCTVC-F092*, July 2011
- [10] Cassa, M.B.; Naccari, M.; Pereira, F.; , "Fast rate distortion optimization for the emerging HEVC standard," *Picture Coding Symposium (PCS)*, 2012, vol., no., pp.493-496, 7-9 May 2012
- [11] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiw Keng Tan, and Thomas Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards Including High Efficiency Video Coding (HEVC)", *Circuits and Systems for Video Technology, IEEE Transactions on*, Issue:99, pp:1-15, October2012
- [12] Correa, G.; Assuncao, P.; da Silva Cruz, L.A.; Agostini, L.; "Adaptive coding tree for complexity control of high efficiency video encoders," *Picture Coding Symposium (PCS)*, 2012, vol., no., pp.425-428, 7-9 May 2012
- [13] Correa, Guilherme; Assuncao, Pedro; Agostini, Luciano; da Silva Cruz, Luis A., "Motion compensated tree depth limitation for complexity control of HEVC encoding," *Image Processing (ICIP)*, 2012 19th IEEE International Conference on, vol., no., pp.217,220, Sept. 30 2012-Oct. 3 2012
- [14] Brightcove, "Encoding for Mobile Delivery", Retrieved 2012July08 from <http://support.brightcove.com/en/docs/encoding-mobile-delivery>, July2012
- [15] Apple, "iOS Developer Library - Preparing Media for Delivery to iOS-Based Devices", Retrieved 2012July08 from [https://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html#apple\\_ref/doc/uid/TP40008332-CH102-SW8](https://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html#apple_ref/doc/uid/TP40008332-CH102-SW8), July2012
- [16] Frank Bossen, "Common test conditions and software reference configurations", *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-I1100*, pp:1-3, May2012
- [17] High Efficiency Video Coding (HEVC) version 8.0. Fraunhofer Institut Nachrichtentechnik Heinrich-Hertz Institut. Retrieved 2012August15 from <http://hevc.hhi.fraunhofer.de/>
- [18] Frank Bossen, "Common test conditions and software reference configurations", *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-I1100*, pp:1-3, July2012
- [19] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); *The WEKA Data Mining Software: An Update*; SIGKDD Explorations, Volume 11, Issue 1
- [20] Bumshik Lee and Munchurl Kim, "No-Reference PSNR Estimation for HEVC Encoded Video", *IEEE Transactions on Broadcasting, Vol:59, Issue:1*, pp:20-27, March2013