IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 9, Issue 3, Ser. I (May. - June. 2019), PP 11-17
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A Survey of Fast Inter Mode Decision Algorithms for HEVC

Ramazan Duvar
(Department of Computer Engineering, Dogus University, Istanbul,Turkey)

Abstract: *The use of coding tree units instead of macroblocks in HEVC increases the coding efficiency while increasing computational complexity. Early termination for coding units at depth level and skipping computations at prediction units called fast mode decision provides a significant contribution to reducing computational complexity in HEVC. In this work, a detailed comparison process is performed by examining the fast inter-mode decision algorithms developed for HEVC.*

Keywords: *Coding tree units, computational complexity, HEVC, inter-mode decision.*

Date of Submission: 08-06-2019 Date of acceptance: 25-06-2019

I. Introduction

When digital video signals are stored in raw form there is a need for high amount of memory. In the case of real-time video transmission very high bandwidth is required when recently introduced 4k and 8k resolutions taken into account. Because of this reason, raw video has to be compressed by making use of spatial, temporal and statistical redundancy in this data.

For years, various studies have been carried out to set standards for video compression. In 2013, JCEG-VC (Joint Collaborative Team on Video Coding), which was formed by the combination of VCEG and MPEG, was declared the HEVC (High Efficiency Video Coding) [1] standard. The general structure of HEVC is similar to previous standards, but also includes many new features and complex algorithms. Thanks to these new features, HEVC can achieve twice the compression ratio of the previous H.264/AVC standard at the same bit rate. But due to new features added, computational complexity of HEVC is significantly more than H.264[2]. A general block diagram of HEVC is given in Fig 1.

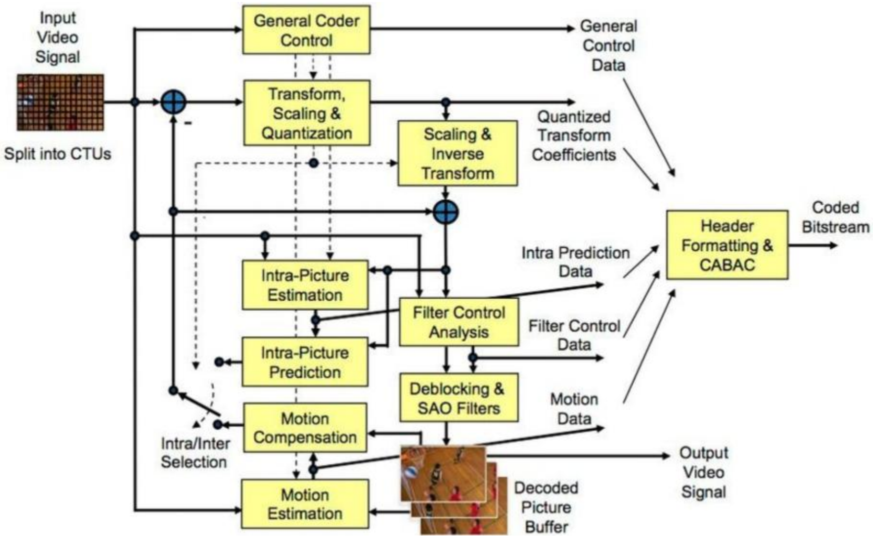

Figure 1: Block diagram of HEVC[1]

A detailed analysis of time consumption of different stage in HEVC is provided in [3]. The most time-consuming part of HEVC is motion estimation and mode decision process where temporal redundancy is exploited. For example, in HM, the reference software of the HEVC standard, an average of 70% of the coding time is spent on processes used to remove temporal redundancy.

One of the most important features added to HEVC is considered as Coding Tree Units(CTUs). Unlike previous coding standards, HEVC uses 64×64 sized and more flexible CTUs instead of 16x16 macroblock blocks. CTUs can consist of 8×8, 16×16, 32×32 or 64×64 sized coding units (CU). In order to determine the optimal CTU structure, motion estimation and ratedistortion (RD) costcomputations are performed on the prediction units(PU), which increases the coding performance and the computational complexity of the HEVC. There are many studies presented in the literature under the name of fast mode decision in order to reduce the computationalcomplexity originating from these operations.

In the second part of this study, the procedures performed during the formation of the CTU in HEVC are described. In the third chapter, fast inter-mode selection algorithms in the literature are examined. In the fourth section, the performance comparison of these methods are given and evaluated.

II. A Detailed Overview of the Quadtree Structure in HEVC

In many video standards, group of pictures (GOP) are constructed by taking the original frames. The next step is to perform slice coding. In HEVC, CTUs and CUs are created in slice coding stage and motion estimation (ME) operations are performed using PU units of CUs. TUs (Transform Units) are used for residual coding. During the encoding process, the RD cost is computed and the structure of CUs in CTUs are determined. In addition to that, during the encoding of GOP structures, the reconstructed frames are also included in the ME process as a reference frame. A general flow chart of the procedures performed to determine the optimal tree structure in HEVC is given in Fig 2.

CU sizes can vary from 8×8 pixels to 64×64 pixels. The CU size is 64×64, when the depth is 0, and the CU size is 8×8 when the depth is 3. The depth is initialized to 0 at the first step in mode decision. The skip mode is checked at the first stage of each depth, and if the best mode is skip mode, other operations are skipped. In the other case, inter-prediction and intra-prediction operations are performed using PUs. PU modes in HEVC are given in Fig. 3. After finding the mode with the lowest RD cost at the relevant depth, the CU is divided into 4 parts symmetrically to form 4 CUs.Same PU dividing operations are performed for these CUs as well. The structure of the block to be coded is created by performing RD cost calculations for all PU modes up to the maximum depth in a recursive way.

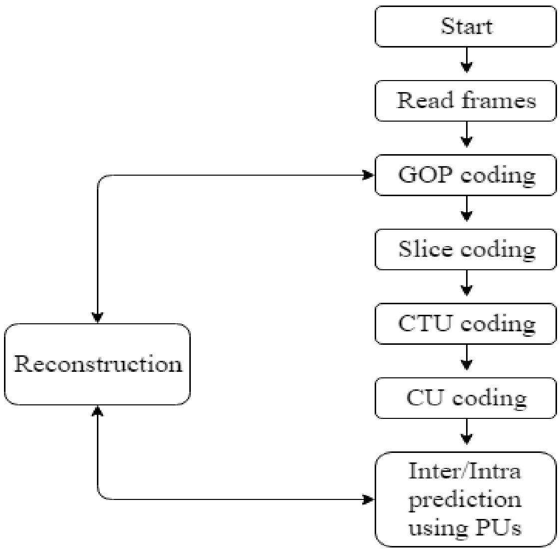

Figure 2: General flowchart

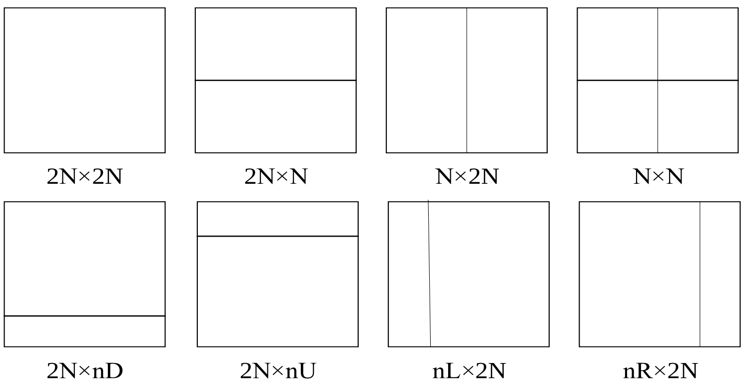

Figure 3:PU modes

Mode Decision: CTU, the coding unit in HEVC, consists of CUs. The CTU size is 64×64. According to the cost calculations, the CTU may consist of one or more CUs. Mode selection operations in HEVC is performed to the algorithm given in Fig.4.

CTUs can contain CUs with different depths according to RD cost computations at each level. Therefore, CTUs in each video frame to be encoded may have a different tree structure. Fig.5 shows a sample CU distribution of CTU and its tree structure.

In addition to these, CUs can also be composed of different PU modes according to the RD cost computation. Fig. 6 shows the CUs selected during encoding of 2^{nd} frame of the "Basketball Pass" sequence and the PU modes selected for these CUs.

Motion Estimation: In HEVC, similar to other encoders, full and sparse search algorithms are used for ME process. In HEVC, SAD approach is used as the block matching criterion during ME process. Test Zone Search [4] algorithm used in HEVC for fast ME computation can be considered as a combined version of diamond and grid search. After the motion estimation is performed, the residual coding and the number of bit required to encode motion vectors are used to decide CTU.

RD Cost Computations: Lagrangian coder control is generally used in video encoders to handle rate-distortion optimization. In HEVC, the Lagrangian multiplier is determined using the QP parameter. This procedure is given in (1). The value calculated according to (1) is used in quantization operations performed during mode selection, motion estimation and coding of residual data.

$$\lambda = \alpha \times Q^2 \tag{1}$$

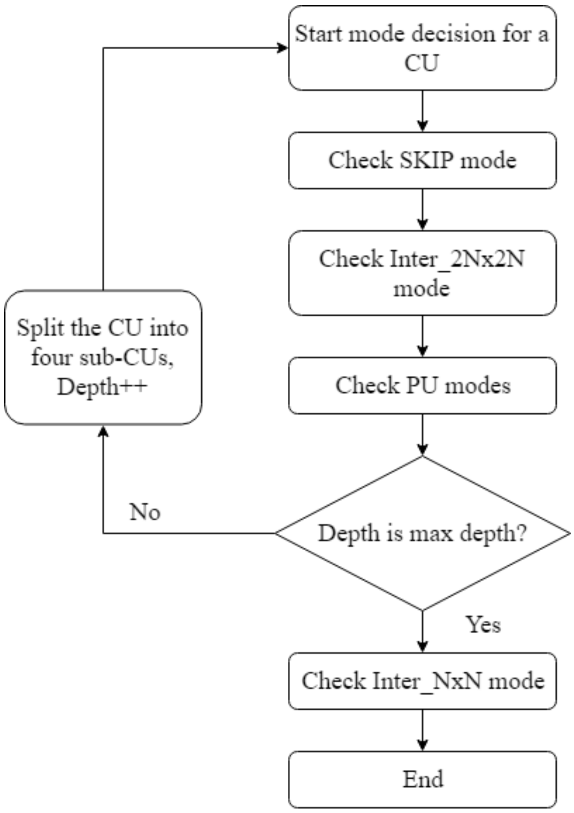

Figure 4: Illustration of CU splitting and mode decision

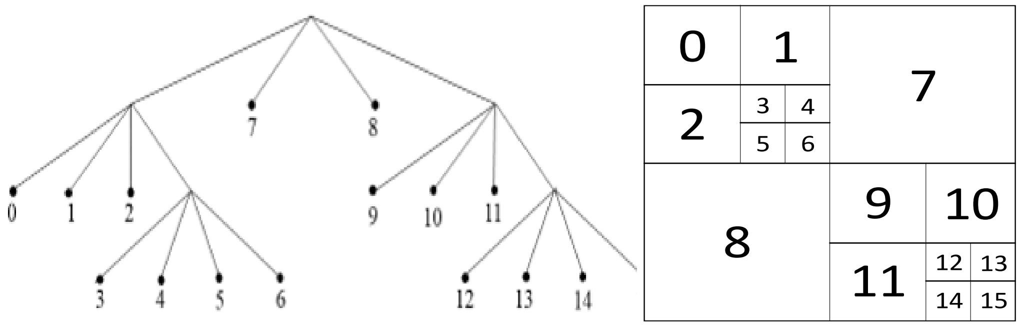

Figure 5:A sample quadtree structure

In this equation,the λ symbol represents the Lagrangian multiplier while Q symbol represents the quantization step interval controlled by the QP parameter.

During mode decisions operations in HEVC, the encoder performs decision-making according to (2).

$$c^* = \arg\min (D_k(c)+\lambda\times R_k(c)),\ c\in C_k \qquad (2)$$

In this equation,C_k represents the set of modes that can be selected, $D_k(c)$represents the SSD (Sum of Squared Difference) value between the original block and the reconstructed block, and $R_k(c)$denotes the amount of bit needed to encode motion vectors and residual data when using mode c.

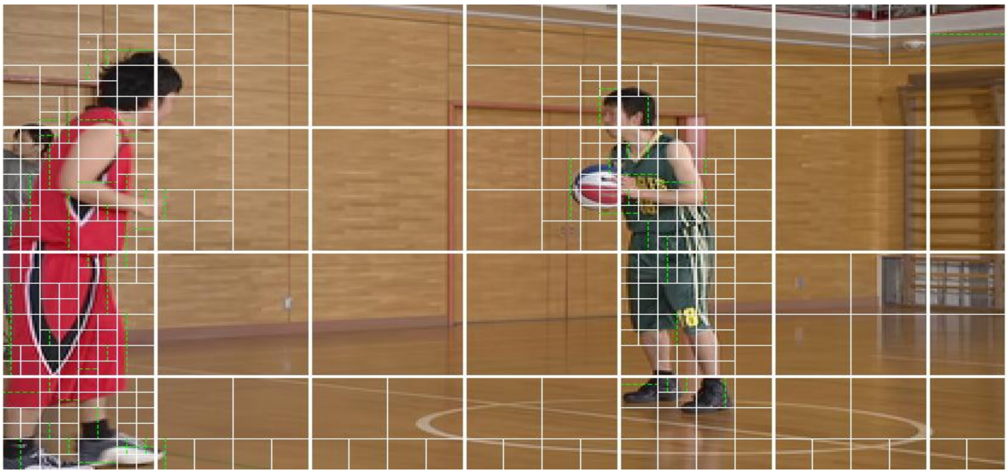

Figure 6: Distribution of the CUs and PUs within CTUs for a frame of "BasketballPass" sequence

In Fig. 6, the regions within the white color represent the CUs, while the regions with green lines indicate the PUs selected for the CUs. The absence of a green line indicates that the CU size is selected as 2N × 2N.

III. Fast Inter Mode Decision Algorithms in the Literature

While determining the CTU structure in HEVC, inter- and intra-predictions are performed for PU modes at various depths and appropriate CTU structure is decided. It is possible to reduce computational complexity by performing operations such as skip, early termination, and prediction in mode decision. There are methods implemented in HEVC reference software for speeding-up the mode decision.There are many studies on this subject in the literature as well. The mode decision methods presented in this study are divided into three sections as CU level, PUlevel and CU+PU level modes. Predicting the depth and early termination at a depth and skip calculations at other depths is considered as CU level and predicting the PU modes considered as PU level methods. In addition to them, there are methods that use both approaches.

Starting from HM version 3.2, various fast mode decision methods have been incorporated to the reference software. In Early Skip Detection (ESD), presented in [5], firstly motion estimation and RD cost calculation is performed. If the residual equal to 0 and the estimated motion vector is equal to motion vector of neighboring block, computations for PU modeare skipped. In the Coded Block Flag (CBF)[6] approach, if residual data is found to be 0 in any PU mode, operations are terminated for other PU modes at that depth. In the

CU Early Termination ECU[7] approach, after all PU modes have been tested, if the least cost is obtained in skip mode, early termination at depth level is carried out. These methods are deactivated in the reference software and can be enabledwhen required. By using these methods together, the total coding time can be reduced by up to 50% with an average loss of 2% on performance.

In [8], it is aimed to determine the estimated CU depth by using the CU depth calculated for neighboring blocks. In the meantime, the correlation between neighboring block and candidate block is utilized. A large correlation value means that a similar depth can be used for the candidate block. This method can reduce the total encoding time by up to 42% with a loss of 1.3%.

A Bayesian theorem-based algorithm is proposed to determine CU dimensions in [9]. In this study, the statistical analysis of coding costs isconsidered and the decision of early termination and skip mode for CUs is performed.

In the method proposed in [10], it is considered that homogeneous regions are encoded with larger and complex regions with smaller size CUs, and in this direction, a method which uses spatiotemporal relationships of pixels is developed for early termination at CU level.The proposed method can reduce the total encoding time by 49.6% with an average loss of 1.4%.

A machine learning-based method to determine CU depths is proposed in [11]. The coding cost and motion vectors were used as input data for the SVM (Support Vector Machine) based classifier. The proposed method can reduce computational complexity by an average of 51% with a loss of 1.98%.

In [12], the motion vectors and the residual data obtained by motion estimation for 2N × 2N blocks, depth information of neighboring blocks and the results of CBF approach were evaluated together in order to determine the CU depth level early. The proposed method can reduce computational complexity by an average of 51% with a loss of 1.1%.

The approach presented in [13], a depth range is determined for the CTU to be encoded using the standard deviation of depth levels of the CUs encoded previously. The proposed method resulted in a 42% reduction in total coding time with an average bit loss of 1.04%.

In [14], it is proposed to use the edge information in the blocks to estimate the appropriate PU modes. For this purpose, the edges of the CUs are determined using the Sobel operator and the appropriate PU mode is decided according to the orientation of these edges. This approach can reduce the coding time by up to 39% with a loss of 1.89%.

The presented method in [15], asymmetric PU modes are not used in depth 3 and depth 0, the RD cost estimation is performed for symmetric PU modes in other depths. Next, the estimation process is performed for asymmetric modes related to PU mode which gives the best results. The proposed method can reduce the total encoding time by up to 50% with a loss up to 1.3%.

The method presented in [16] is intended to find the appropriate CU depth as well as to find the appropriate PU mode. For this purpose, after motion estimation for 2N×2N mode, the distribution of residual data according to regions related to PU modes is evaluated and appropriate modes are decided according to this distribution. The proposed method can reduce the total coding time by an average of 80% with a loss of 4.5%.

It is proposed to decide the appropriate modes for CUs of size 2N×2N after estimation of N×N size PUs in [17]. In this approach, it is decided which modes to use according to the motion vector results obtained from N×N sized PUs.

The proposed algorithm in [18] consists of two stages. In the first stage, homogeneity and redundancy are checked and the appropriate depth is determined (Method 1), in the second stage, the PU modes are determined according to the homogeneity metric (Method 2).In order to determine homogeneity, the ratio of the average of the pixels in PUs to each other was used as a metric.

IV. Experimental Results

The performance of the methods in HEVC is generally evaluated by using the metric proposed in [19] and the ΔT metric is computed as in (3). ΔB metric is computed from bitrate and PSNR values obtained by running the reference software for various QP values. In addition, the ratio of ΔB to ΔTvalue is used to make a fair assessment. This metric is called ΔX.

$$\Delta T=\frac{Time_{referance}-Time_{proposed}}{Time_{referance}}\times 100 \qquad (3)$$

Table 1, Table 2 and Table 3 show the performance of the CU-level methods, PU-level methods and the CU+PU-level methods in the LDB configuration, respectively. “Traffic”, “Kimono”, “ParkScene”, “FourPeople”, “Johnny” and “KrsitenAndSara” sequences were used for comparison.

Table 1: Performance Comparisons Between CU level Methods In Low Delay B Configuration

Methods	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[8]	-	-	-	1.56	29.1	5.36	1.75	33.49	5.23	1.02	49.9	2.04	0.51	50.8	1.00	0.49	45.5	1,07
[9]	2.00	41.3	4.85	0.30	27.8	1.08	0.90	28.9	3.11	2.40	47.8	5.02	2.00	56.0	3.53	2.80	48.5	5.77
[10]	-	-	-	1.18	58	2.03	0.80	60	1.33	1.30	82.0	1.58	1.45	86.0	1.68	1.73	81.0	2.14
[11]	2.0	41.7		0.8	47.9		1.1	48.3		1.6	65.6		-0.3	73.9		0.5	69.6	
[12]	2.00	56.1	3.57	2.0	56.1	3.57	1.50	47.8	3.13	2.80	66.4	4.21	2.50	70.9	3.52	2.30	68.2	3.37
[13]	1.30	59.9	2.18	1.00	44.3	2.26	1.30	56.5	2.30	1.40	66.3	2.11	1.10	70.3	1.56	0.90	65.6	1.37
[14]	1.10	40.7	2.70	0.6	42.2	1.42	1.04	39.6	2.62	0.83	45.5	1.82	0.58	46.0	1.26	0.84	45.6	1.84
[18] (Method1)	0.41	51.6	0.79	0.71	44.9	1.58	0.68	49.0	1.38	0.48	66.6	0.72	1.09	75.2	1.45	0.48	68.6	0.70
[5]+[6] +[7]	1.50	56.5	2.65	1.20	43.4	2.76	1.50	51.2	2.92	1.30	70.3	1.85	1.30	77.2	1.68	1.10	70.8	1.55

Table 2: Performance Comparisons Between PU level Methods In Low Delay B Configuration

Methods	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[14]	2.10	39.2	5.36	-	-	-	-	-	-	2.03	34.6	5.86	1.87	33.9	5.51	-	-	-
[18] (Method2)	2.30	41.0	5.60	1.58	47.9	3.30	2.30	49.9	4.60	2.34	46.0	4.87	2.77	54.4	5.09	1.50	50.0	3.00

Table 3: Performance Comparisons Between CU+PU level Methods In Low Delay B Configuration

Methods	Traffic			Kimono			ParkScene			FourPeople			Johnny			KristenAndSara		
	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX	ΔB	ΔT	ΔX
[16]	-	-	-	3.30	75.6	4.36	4.40	77.2	5.70	4.80	85.1	5.64	5.90	88.2	6.69	5.00	85.8	5.83
[17]	3.87	60.0	6.45	2.81	59.0	4.76	2.48	55.0	4.51	-	-	-	-	-	-	-	-	-
[18] (Method1+ Method2)	3.50	65.6	5.33	3.16	67.7	4.66	3.79	68.6	5.52	3.30	79.3	4.16	4.21	84.7	4.97	2.35	81.5	2.88

Experimental results show that although all methods provide better results in some sequences compared to each other, CU and PU level methods reduce computational complexity by around 50% with acceptable bit losses. It is seen from the Table 3,computational complexity can be reduced to 80% in CU + PUlevel methods. In addition to these, the proposed method in [18], provides a good balance between coding efficiency and time savings compared to the state-of-the-art approaches considering all test sequences.

V. Conclusion

In this work, fast inter-mode decision approaches proposed for HEVC are compared. In HEVC, it is found that the coding units called CTU showed a tendency to increase the computational complexityto the older video coding standards according to the dimensions of the units depending on the frame sizes, and this trend makes a significant contribution to the reduction of the computational complexity of the HEVC reference software.

References

[1] G. J.Sullivan, J. R.Ohm, W.-J.Han, T.Wiegand, Overview of the High Efficiency Video Coding (HEVC) standard, *IEEE Transactions on Circuits and Systems for Video Technology, 22(12)*, 2012, 1648–1667.

[2] J. R.Ohm, G. J.Sullivan, H.Schwarz, T. K.Tan, Wiegand T., Comparison of The Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC), *IEEE Transactions on Circuits and Systems for Video Technology, 22(12)*, 2012, 1669-1684.

[3] F.Bossen, B.Bross, K.Suhring, D.Flynn, HEVC Complexity and Implementation Analysis, *IEEE Transactions on Circuits and Systems for Video Technology,22(12)*, 2012, 1685-1696.

[4] X. Tang,S. Dai, C. Cai., An analysis of TZ Search Algorithm in JMVC, International Conference on Green Circuits and Systems (ICGCS), 2010, pp. 516-520.

[5] R. H.Gweon, and Y. L.Lee, Early Termination of CU Encoding to Reduce HEVC Complexity, *Joint Collaborative Team on Video Coding*, F045, 2011, 1-4.

[6] K. Choi, S. H.Park, E. S.Jang, Coding Tree Pruning Based CU Early Termination, *Joint Collaborative Team on Video Coding*, F092, 2011, 1-4.

[7] J.Yang, J.Kim, K.Won, H.Lee, B.Jeon, Early SKIP Detection for HEVC, *Joint Collaborative Team on Video Coding*, G543, 2011, 1-4.

[8] L.Shen, Z.Liu, X.Zhang, W.Zhao, Z.Zhang, An Effective CU Size Decision Method for HEVC Encoders, *IEEE Transactions on Multimedia, 15(2)*, 2013, 465-470.

[9] J.Lee, S.Kim, K.Lim, S.Lee, A Fast CU Size Decision Algorithm for HEVC, *IEEE Transactions on Circuits and Systems for Video Technology, 25(3),* 2015, 411-421.

[10] S.Ahn, B.Lee, M.Kim, A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding. *IEEE Transactions on Circuits and Systems for Video Technology, 44(3),* 2015, 422 - 435.

[11] L.Xu, Z.Pan, H.Yuan, X.Wang, S.Kwong, Y.Zhang, Machine Learning-Based Coding Unit Depth Decisions for Flexible Complexity Allocation in High Efficiency Video Coding, *Delay Differential Equations: Recent Advances and New Directions*, DOI: 10.1109/TIP.2015.2417498.

[12] Y.Li, G.Yang, Y.Zhu, X.Ding, X.Sun, Adaptive Inter CU Depth Decision for HEVC Using Optimal Selection Model and Encoding Parameters, *IEEE Transactions on Broadcasting, 63(3),* 2017, 535-546.

[13] J. H.Bae, M. H.Sunwoo, Adaptive Early Termination Algorithm Using Coding Unit Depth History in HEVC, *Journal of Signal Processing Systems*, DOI: 10.1007/s11265-018-1399-y.

[14] X.Huang, Q.Zhang, X.Zhao, W.Zhang, Y.Zhang, Y.Gan, Fast Inter-Prediction Mode Decision Algorithm for HEVC, *Signal, Image and Video Processing, 11(1),* 2017, 33-40.

[15] J.Vanne, M.Viitanen, T. D.Hamalainen, Efficient Mode Decision Schemes for HEVC Inter Prediction, *IEEE Transactions on Circuits and Systems for Video Technology, 24(9),* 2014, 1579-1593.

[16] B.Z. Li, and H.Li, An Efficient Fast Mode Decision Method for Inter Prediction in HEVC, *IEEE Transactions on Circuits and Systems for Video Technology, 26(8),* 2016, 1502-1515.

[17] T.Mallikarachchi, D. S.Talagala, H. K.Arachchi, Fernando A., Content-Adaptive Feature-Based CU Size Prediction for Fast Low-Delay Video Encoding in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology, 28(3),*2018, 693-705.

[18] R. Duvar, *Fast Motion Estimation and Mode Decision Approaches for HEVC,* doctoral diss., Kocaeli University Institute of Science and Technology, Kocaeli, 2019.

[19] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, *document VCEG-M33*, 2001, 1-4.

Ramazan Duvar." A Survey of Fast Inter Mode Decision Algorithms for HEVC. " IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol. 9, no. 3, 2019, pp. 11-17.