FAST TEXTURE INTRA SIZE CODING BASED ON BIG DATA CLUSTERING FOR 3D-HEVC

Hamza HAMOUT [1] *and Abderrahmane ELYOUSFI* [2]
[1]Computer Systems and Vision Laboratory, Faculty of Sciences,
Ibn-Zohr University, Agadir, Morocco
[2]Computer Science Department, National Engineering School of Applied Sciences,
Ibn-Zohr University, Agadir, Morocco
{[1]hamzahamout@gmail.com, [2]elyousfiabdo@yahoo.fr}

ABSTRACT

High Efficiency Video Coding (HEVC) based 3D video coding (3D-HEVC) is the latest new joint standardization effort of ISO/IEC MPEG and ITU-T Video Coding Experts Group (VCEG) for 3D video coding. This new standard provides a significant coding improvement, especially for high resolution videos. However, one the most important challenges in *3D-HEVC* is time complexity. In technical terms, *3D-HEVC* is a hybrid video coding approach using quad-tree based block partitioning with more flexible Coding Unit (*CU*) size selection, which increases the coding efficiency of *3D-HEVC* significantly, but also brings huge computational complexity due to the Rate Distortion (*RD*) cost calculation of all possible dimensions of *CU* to select the optimal one. To reduce this computational complexity, this paper proposes an efficient fast intra coding unit based on big data analysis. The experimental results demonstrate that the novel approach provides a significant trade-offs between computational complexity and RD performance.

***Index Terms*—** Coding unit decision, big data, 3D-HEVC, JCT-3V, AM-PCM.

1. INTRODUCTION

With the explosive increasing of 3D video applications, a new generation video coding standard, called 3D video extension of High Efficiency Video Coding (*3D-HEVC*) is currently being developed by *JCT-3V* [1]-[3], jointly established by the *ISO/IEC MPEG* and *ITU-T VCEG* [4]. *3D-HEVC* support a new 3D video representation, commonly known as Multiview texture Videos plus Depth maps (*MVD*) format [5]. The motivation for *MVD* usage, is to reduce the bandwidth for a 3D video transmission in which, at the receiver side, only a small number of captured texture videos as well as associated depth maps are used for synthesizing virtual views suitable for displaying the *3D* content on an auto-stereoscopic 3D using Depth Image Based Rendering (*DIBR*) techniques [6].

In the *3D-HEVC*, a new flexible coding unit partition structure is used, namely Coding Tree Unit (CTU) [7], in which each *CTU* can be recursively splited to form a quadtree structure with new unit types: Coding Unit (*CU*), Prediction Unit (PU) [8]. Fig. 1 shows the recursive splitting structure of *CU* and *PU*.

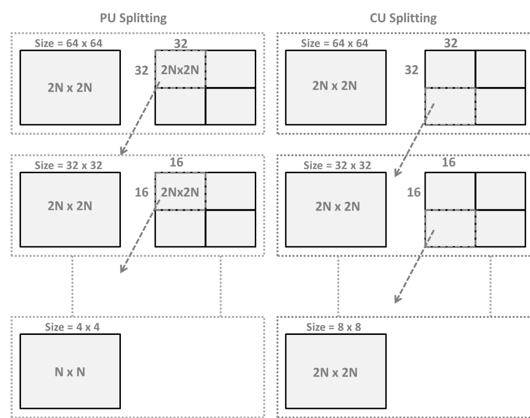

Fig. 1. Recursive splitting structure of Coding Unit and Prediction Unit.

In *3D-HEVC* intra-coding, the recursive partition process is very time-consuming, in which the *3D-HEVC* encoder search recursively for the best *PU* size decision by evaluating the Rate-Distortion (*RD*) performance [3] of all intra size modes [9]. Therefore, this evaluation provides much higher compression efficiency and better synthesis quality [10], but results in a significant increase of the computational complexity at the encoder.

Recently, there has been some researches focusing on fast depth map coding in *3D-HEVC* to accelerate the *CU* intra size decision process [11]-[12]. A fast intra coding algorithm is designed in [11] to speed up the quad-tree decision by the good feature Corner Point. In [12], the authors propose a strategies for fast *CU* size decision, in which the encoder ignores the evaluation of the smaller *CU* size if some specific conditions are satisfied.

Consequently, this paper proposes a fast texture intra coding size decision based on a robust Automatic Merging Possibilistic Clustering (AM-PCM) for big data analysis developed by [13]. The main idea of the proposed algorithm is to use AM-PCM on selective training data, next, we create a unified model decision based on massive data, Finally this model is used to accelerate the *3D-HEVC* intra coding. The remainder of the paper is organized as follows. Section 2 gives an overview of the clustering algorithm and the features selections. In Section 3, we describe briefly the process used for big data analysis followed by the proposal size decision model. Section 4 presents the simulation results, while Section 5 concludes the work of this paper.

2. *AM-PCM* ALGORITHM AND FEATURES SELECTION

Clustering analysis is a method to find groups within data with most similar objects in the same cluster. *AM-PCM* is a robust clustering method proposed by [13] to improve the weakness of the Possibilistic C-Means (*PCM*) clustering [14]. In the rest, we briefly describe the *AM-PCM* and we present the hole clustering process in Alg. 1. Please for more information, refere to [13].

Let $X=\{x_1, x_2, ..., x_N\}$ be a set of N data, $U=[u_1, u_2, ..., u_C]^T=[\mu_{ij}]_{C\times N}$ partition matrix, $A=\{a_1, a_2, ..., a_C\}$ cluster centers and C number of cluster. The suggested objective function is defined as follow:

$$J_{AM\text{-}PCM}(U,A) = \sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}[d_{ij}^2 - \gamma(1-\frac{k}{k+1}.\mu_{ij}^{1/k})], k > 1. \tag{1}$$

By differentiating Eq. 1 with respect to μ_{ij} and a_i and setting it respectively to 0, we get:

$$\mu_{ij} = (\frac{\gamma - d_{ij}^2}{\gamma})^k, a_i = \frac{\sum_{j=1}^{N}\mu_{ij}x_j}{\sum_{j=1}^{N}\mu_{ij}}. \tag{2}$$

We set $d^2(x_i, x_j)$ the *Euclidian* distance between x_i and x_j and let set

$$\Omega = \{d^2(x_1,x_2), ..., d^2(x_1,x_N), ..., d^2(x_{N-1},x_N)\},$$
$$\Delta = \text{sort}\{\Omega\} = \{D_{(0)}, D_{(1)}, ..., D_{(N\cdot(N-1)/2)}\},$$
$$q = \frac{1}{N-1}[N\cdot\sum_{i=1}^{C}(\sum_{j=1}^{N}\mu_{ij}/\sum_{l=1}^{C}\sum_{j=1}^{N}\mu_{ij})^2 - 1], \tag{3}$$

$$k = \frac{log(1-(1/p))}{log(1-(D_{((N\cdot(N-1)/2)\cdot q)}/p^2\gamma))}, p > 1, \tag{4}$$

$$D = \frac{D_{((N\cdot(N-1)/2)\cdot q)}}{p^2}. \tag{5}$$

To avoid bad initials, we plan to use all data points as initial cluster centers. The correlation coefficient matrix r^2 and the densest cluster centers R are as follow:

$$r_{il}^2 = \frac{u_i^T u_l}{\|u_i\|\|u_l\|}, R_i = \sum_{l=1}^{C} r_{il}^2. \tag{6}$$

The merging step is carried out as follows, let G to be the clusters no yet merged. Let

$$E_m = \{l : g = \underset{i\in G}{\text{Argmax}}(R_i), r_{gl}^2 > \rho, l \in G\}. \tag{7}$$

where, $G = \{1,2,...,C\}\backslash(E_1 \cup E_2 \cup \; \cup E_{m-1})$, and $m = 1,2,...,C^{new}$.

For the new partition of the dataset, the algorithm compute the center of each cluster for $1 \le m \le C^{new}$:

$$\mu_{mj}^{new} = \frac{\sum_{l\in E_m}\mu_{lj}}{|E_m|}, a_m^{new} = \frac{\sum_{l\in E_m} a_l}{|E_m|}. \tag{8}$$

Algorithm 1: AM-PCM algorithm

Input : Dataset $X=\{x_1, x_2, ..., x_N\}$
Output: Cluster centers A
Initial $D^{(0)}=D_{((N\cdot(N-1)/2)/\sqrt{N})}/2$, $p=3$, ρ=0.9, ε=0.001;
Set t=0, $A^{(0)}=X$, $C=N$, $\gamma=\max\{_{1\le i,l\le N} d^2(x_i,x_l)\}$ and $\Delta=\text{sort}\{\Omega\}=\{D_{(0)}, D_{(1)}, ..., D_{(N\cdot(N-1)/2)}\}$;
while *(True)* **do**
    Compute k by Eq. 4, U and $A^{(t+1)}$ by Eq. 2;
    if $\|A^{(t+1)} - A^{(t)}\| < \varepsilon$ **then**
        Break;
    end
    Set m=1 and $G=\{1,2,...,C\}$;
    Compute r^2 with U and R with r^2 by Eq. 6;
    while $|D > 0|$ **do**
        Compute E_m with R by Eq. 7;
        Set $G = G \setminus E_m$ and m=m+1;
    end
    Set $C^{new} = m$;
    Compute $U = U^{new}$ and $A = A^{new}$ with Eq. 8;
    Compute q by Eq. 3 and $D^{(t+1)}$ by Eq. 5;
    if $D^{(t+1)} < D^{(t)}$ **then**
        $D^{(t+1)} = D^{(t)}$;
    end
    t=t+1;
end
return $A^{(t)}$;

The *CU* sizes modes are mostly related the screen video homogeneity, in which, the homogenous *CUs* are likely to be encoded in larger size and non-homogeneous *CUs* are likely to be partitioned into small size to be encoded efficiently. Thus, extracting the features that can describe the video homogenous can be used to prediction the *CU* sizes. Motivated by state of the art works on video coding [15]-[16], we select the variance (*Var*) [15] and Amplitude of Simplified Mass Center Vector (ASMCV) [16] as the extracted features. Let $I(i, j)$ be the luminance value of the pixel at (i, j) in 64×64 CU, the *Var*, *ASMCV* and the amplitude are respectively defined by Eq. 9, Eq. 10 and Eq. 11 as follows:

$$Var = \frac{1}{64\times 64}\sum_{i=0}^{63}\sum_{j=0}^{63}[I(i,j)]^2 - [\frac{1}{64\times 64}\sum_{i=0}^{63}\sum_{j=0}^{63}I(i,j)]^2. \tag{9}$$

$$d_{x_{i,j}} = \textstyle\sum_{j=0}^{64/2-1}\sum_{i=0}^{63}(\frac{64}{2}-j)\times|I(i,j) - I(i,64\text{-}j\text{-}1)|,$$
$$d_{y_{i,j}} = \textstyle\sum_{i=0}^{64/2-1}\sum_{j=0}^{63}(\frac{64}{2}-i)\times|I(i,j) - I(64\text{-}i\text{-}1,j)|,$$

$$ASMCV = d_{x_{i,j}} + d_{y_{i,j}}. \tag{10}$$

$$Amp = \sqrt{ASMCV^2 + Var^2}. \tag{11}$$

3. BIG DATA ANALYSIS AND PROPOSED TEXTURE INTRA SIZE DECISION MODEL

In this section, we describes in details and step by step, the selecting training data set for *AM-PCM* clustering, the big data analysis and finally the establishment of the proposed model decision for fast texture intra *CU* coding.

The training data set was composed by multiple video sequences, "Kendo," "Balloons," "Newspaper," "Ponznan-Hall2," "PonznanStreet," "Undodancer," "Shark" and "GT-Fly" proposed by the *JCT-3V* [17]. We compute the *Var* (Eq. 9) and *ASMCV* (Eq. 10) features for all *CU* of the first texture frame from every 24 texture frames (intra coding period) and sorted according to amplitude (Eq. 11), totally of 100554 64×64 *CUs*. From this sorted data, we start selecting the features vectors by a frequency of 25 as the input *AM-PCM* training to take into consideration all possible content, totally of 4000 vectors. As described in Alg. 1, the outputs of the *AM-PCM* are the cluster centers, totally of 976 cluster centers sorted according to the amplitude.

The goal of this paper is to estimate a model size decision to predict *CU* splitting flags based on the big data analysis and then reduce the texture intra coding computational complexity. For that purpose, we construct our big data set using all texture *CUs* of all eights sequences [17], totally of 2430900 64×64 *CUs*, each element of this big data set is composed by variance, *ASMCV* and *CU* size label. We combine the big data set and the center clusters by distributing each element of the big data to its center cluster how minimize the Euclidian distance. For each center cluster, we determine the dominant sizes modes based on the percentage distribution sizes modes, if the size mode percentage is great than 10%, it well be considered as dominant mode, else, it will be ignored. Regrouping the clusters that have the same dominant sizes modes, we get exactly five regions. Please note that this process is done for all texture *QPs* (25, 30, 35 and 40). Tab. 1 presents the distribution result after regrouping process. Taking into consideration the distribution result presented in Tab. 1, we define a unified *3D-HEVC* fast intra coding size for texture. Tab. 2 presents the unified model according to four thresholds, the symbol "X" represents the supported sizes for a given thresh-

Table 1. The intra size modes distribution for all *QPs* in each cluster for texture frames

Index	QP	Coding Unit Size				
		64×64	*32×32*	*16×16*	*8×8*	*4×4*
1	25	96.60%	3.40%	0.00%	0.00%	0.00%
	30	97.56%	2.39%	0.04%	0.00%	0.00%
	35	96.66%	3.34%	0.00%	0.00%	0.00%
	40	93.40%	6.16%	0.42%	0.03%	0.00%
Average		96.06%	3.82%	0.12%	0.01%	0.00%
2	25	52.32%	41.18%	6.07%	0.33%	0.10%
	30	67.48%	28.17%	3.85%	0.04%	0.10%
	35	72.21%	23.43%	3.59%	0.67%	0.10%
	40	70.51%	24.73%	4.07%	0.62%	0.08%
Average		65.63%	29.38%	4.40%	0.42%	0.10%
3	25	23.07%	52.63%	18.93%	4.43%	0.94%
	30	27.16%	47.07%	19.79%	5.08%	0.90%
	35	25.04%	46.80%	21.15%	6.05%	0.96%
	40	25.31%	47.94%	21.57%	4.82%	0.36%
Average		25.15%	48.61%	20.36%	5.10%	0.79%
4	25	10.91%	46.55%	28.49%	11.22%	2.83%
	30	11.54%	46.29%	29.72%	10.43%	2.03%
	35	11.72%	44.12%	29.78%	12.22%	2.15%
	40	8.92%	47.67%	33.79%	9.05%	0.58%
Average		10.77%	46.16%	31.45%	10.73%	1.90%
5	25	2.92%	32.10%	34.23%	21.53%	9.23%
	30	3.61%	33.29%	36.14%	20.51%	6.44%
	35	3.76%	34.78%	37.99%	19.13%	4.34%
	40	4.66%	39.05%	40.32%	14.68%	1.29%
Average		3.74%	34.81%	37.17%	18.96%	5.32%

Table 2. Unified fast intra coding size for texture frames

Amp	Coding Unit Size				
	64×64	*32×32*	*16×16*	*8×8*	*4×4*
$Amp < Th1$	X	-	-	-	-
$Th1 \leq Amp < Th2$	X	X	-	-	-
$Th2 \leq Amp < Th3$	X	X	X	-	-
$Th3 \leq Amp < Th4$	-	X	X	X	-
$Amp \leq Th4$	-	X	X	X	X

Table 3. The thresholds according to *QP* for texture

QP	*Thresholds*			
	Th1	*Th2*	*Th3*	*Th4*
25	5657.00	98727.6	426771.41	504187.06
30	16275.99	206813.62	621200.25	671023.108
35	16275.99	330096.36	875510.53	960894.37
40	165472.06	488527.10	1381988.19	1550151.40

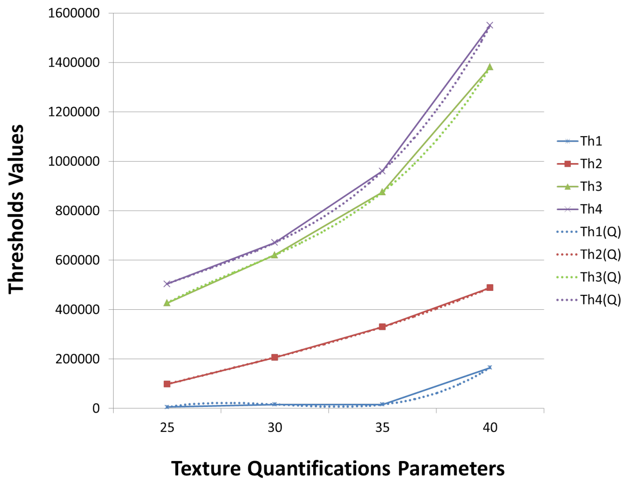

Fig. 2. The curves of the thresholds and their mathematical modeling functions.

old. The thresholds are calculated based on the amplitude of the regrouping clusters centers, which are defined in Tab. 3 for all texture *QPs*. Let set *Q=(QP-20)/5*, The mathematical modeling of the four proposed thresholds are described in Eq 12 and their curves are presented in Fig. 2.

$$\begin{cases} Th1(Q) = 26636 \cdot Q^3 - 165125 \cdot Q^2 + 319542 \cdot Q - 175396. \\ Th2(Q) = 3325.2 \cdot Q^3 - 12353 \cdot Q^2 + 121868 \cdot Q - 14113. \\ Th3(Q) = 32048 \cdot Q^3 - 162345 \cdot Q^2 + 457131 \cdot Q + 99938. \\ Th4(Q) = 29392 \cdot Q^3 - 114833 \cdot Q^2 + 305593 \cdot Q + 284036. \end{cases} \tag{12}$$

4. EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed fast texture intra coding size decision in comparison with *3D-HEVC* encoder. All the experiments are based on Common Test Conditions (*CTC*) [17] and executed in reference software *HTM-16.2* [18]. The video test sequences used are presented as follow: "Kendo," "Balloons," "Newspaper," "PonznanHall2," "PonznanStreet," "Undodancer," "Shark" and "GT-Fly" in which each sequence contains three texture. For all experiments, we use all-intra configuration and recommended CTC *QPs* for texture, namely 25, 30, 35 and 40. The coding performance is evaluated based on the coding time reduction (ΔT), (*BD-BR*, *BD-PSNR*)[19] and (ΔBR, $\Delta PSNR$) of the texture using *YUV-PSNR*. The test platform is Intel(R) Xeon(R) *CPU* E3-1225 v5 @ 3.30GHz, 8GB RAM with Microsoft VS C++ 2010 compiler.

In Tab. 4, we summaries the performance of the proposed size decision model. Up to 54.12% and 20.43% respectively as the maximum and the minimum time saving. The proposed method achieves an average complexity reduction of 36.33%.

Table 4. Experimental results of the proposed algorithm

Sequences	*BD-BR* (%)	*BD-PSNR* (dB)	ΔBR (%)	$\Delta PSNR$ (dB)	ΔT (%)
Balloons	0.26	-0.013	0.12	-0.008	28.94
Kendo	0.43	-0.019	0.30	-0.007	33.45
Newspaper1	0.16	-0.010	0.08	-0.007	20.43
GTFly	1.93	-0.094	0.98	-0.044	47.25
PonznanHall2	1.04	-0.025	0.52	-0.014	54.12
PonznanStreet	0.34	-0.013	0.26	-0.009	29.49
UndoDancer	0.30	-0.016	0.12	-0.011	33.78
Shark	1.17	-0.049	0.36	-0.033	43.21
1024x768	0.28	-0.030	0.17	-0.007	27.61
1920x1088	0.95	-0.039	0.45	-0.022	41.57
Average	0.70	-0.030	0.34	-0.017	36.33

Table 5. The performance of the proposed algorithm under different QPs

QP	ΔBR (%)	$\Delta PSNR$ (dB)	ΔT (%)
25	0.12	-0.006	25.21
30	0.25	-0.011	31.97
35	0.37	-0.017	38.67
40	0.63	-0.032	49.48

This coding time reduction is particularly high for low motion sequences "PonznanHall2" (54.12%), but is still obvious for rich motion sequences such as "UndoDancer" (33.78%) and "Shark" (43.21%). Moreover, the proposed algorithm bring 0.70%, 0.34% increase respectively for *BD-BR* and ΔBR and 0.030dB, 0.017dB decrease respectively for *BD-PSNR* and $\Delta PSNR$ when compared to HTM-16.2. Tab. 5 presents more details experiments results (ΔBR, $\Delta PSNR$ and ΔT) of the proposed overall algorithm under different *QP* compared to *3D-HEVC*. As shown in Tab.5, with the *QP* increase, the encoder runtime savings increase, ΔBR increases and the quality drop increase.

Comparing the proposed algorithm to related works that also propose *3D-HEVC* fast intra coding unit, all related works are focusing only on fast depth map intra coding unit, it is not possible to make any fair comparison with the present work.

5. CONCLUSION

The proposed method presented in this work performs the big data analysis based on Automatic Merging Possibilistic Clustering Method to extract a *CU* size model decision for texture intra coding. Based on this model, we predict *CU* splitting flags and then reduce the computational complexity of the encoder. The comparative experimental results demonstrate that the proposed model reduce significantly the computational complexity into 36.33% on average for limited *BD-BR*, ΔBR, *BD-PSNR* and $\Delta PSNR$ losses.

6. REFERENCES

[1] K. Müller et al., ”3D High-Efficiency Video Coding for Multi-View Video and Depth Data,” in IEEE Transactions on Image Processing, vol. 22, no. 9, pp. 3366-3378, Sept. 2013.

[2] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall and A. Vetro, ”Standardized Extensions of High Efficiency Video Coding (HEVC),” in IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1001-1016, Dec. 2013.

[3] G. J. Sullivan, J. R. Ohm, W. J. Han and T. Wiegand, ”Overview of the High Efficiency Video Coding (HEVC) Standard,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[4] G. Tech, Y. Chen, K. Mller, J. R. Ohm, A. Vetro and Y. K. Wang, ”Overview of the Multiview and 3D Extensions of High Efficiency Video Coding,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 35-49, Jan. 2016.

[5] P. Kauff, N. Atzpadin, C. Fehn, M. Mller, O. Schreer, A. Smolic, R. Tanger, ”Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability,” in Signal Processing: Image Communication, vol. 22, no. 2, pp. 217-234, Feb. 2007.

[6] C. Fehn, ”Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV,” in Proc. Stereoscopic Displays and Virtual Reality Systems XI, pp. 93-104, Jan. 2004.

[7] E. G. Mora, J. Jung, M. Cagnazzo and B. Pesquet-Popescu, ”Initialization, Limitation, and Predictive Coding of the Depth and Texture Quadtree in 3D-HEVC,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 9, pp. 1554-1565, Sept. 2014.

[8] H. Hamout and A. Elyousfi, ”An efficient edge detection algorithm for fast intra-coding for 3D video extension of HEVC,” in Journal of Real-Time Image Processing, doi: 10.1007/s11554-017-0718-z.

[9] H. Hamout and A. Elyousfi, ”Low Complexity Intra Mode Decision Algorithmfor 3D-HEVC,” in Proc. European Signal Processing Conference (EUSIPCO), Kos island, Greece, Aug. 2017, pp. 1520-1524.

[10] G. Chi, X. Jin and Q. Dai, ”A quad-tree and statistics based fast CU depth decision algorithm for 3D-HEVC,” in Proc. IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, China, Jul. 2014, pp. 1-5.

[11] H. B. Zhang, Y. L. Chan, C. H. Fu, S. H. Tsang and W. C. Siu, ”Quadtree decision for depth intra coding in *3D-HEVC* by good feature,” in Proc. IEEE International Conference on Acoustics, Speech and SignalProcessing (ICASSP), Shanghai, China, Mar. 2016, pp. 1481-1485.

[12] K. K. Peng, J. C. Chiang and W. N. Lie, ”Low complexity depth intra coding combining fast intra mode and fast CU size decision in 3DHEVC, ” in Proc. IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, Sept. 2016, pp. 1126-1130.

[13] M. S. Yang and C. Y. Lai, ”A Robust Automatic Merging Possibilistic Clustering Method, ” in IEEE Transactions on Fuzzy Systems, vol. 19, no. 1, pp. 26-41, Feb. 2011.

[14] R. Krishnapuram and J. M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst., vol. 1, no. 2, pp. 98110, May 1993.

[15] Y. H. Huang, T. S. Ou and H. H. Chen, ”Fast Decision of Block Size, Prediction Mode, and Intra Block for H.264 Intra Prediction,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 8, pp. 1122-1132, Aug. 2010.

[16] A. Elyousfi, H. Hamout and A. E. Hachimi, ”An efficient intra block size decision for H.264/AVC encoding optimization,” in Proc. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Morocco, Agadir, Nov. 2016, pp. 1-5.

[17] K. Müller and A. Vetro, ”Common test conditions of 3DV core experiments,” Joint Collaborative Team on 3D Video Coding Extension Development Document JCT3V-G1100, 7th Meeting, San Jos, US, January, 2014.

[18] 3D-HEVC Reference Software: HTM Version 16.2. [Online]. Available: https://hevc.hhi.fraunhofer.de/trac/3d-hevc/browser/3DVCSoftware/tags/HTM-16.2, accessed May 27, 2016.

[19] G. Bjntegaard, Calculation of Average PSNR Differences Between RD-Curves, ITU-T SG 16 Q.6 VCEG, document VCEG-M33, Apr. 2001.