International Conference on Communication and Signal Processing, July 28 - 30, 2020, India

Fast SCC in HEVC using a Palette Mode Decision Tree Classifier

Pavan Kumar G and Sreelekha G

Abstract—**Screen Content Coding (SCC) extensions launched as part of the High Efficiency Video Coding (HEVC) standard help in achieving high compression of screen content video with good quality. At the same time the SCC tools come with high computation cost. Such complexity would be a deterrent for the successful application in various real time applications like remote desktop screen sharing, cloud gaming, wireless displays etc. In order to reduce the computational complexity, this paper proposes a decision tree based classifier which classifies Coding Units as Palette or Non Palette blocks. Palette mode computations are skipped for Non palette blocks which results in time saving at the same time with negligible quality degradation. Feature selection and database training for the palette mode decision tree classifier are also discussed in detail. Experimental results show an average time saving of 38.86% with an average BDBR increase of 1.353% and an average BDPSNR degradation of 0.125 dB.**

Index Terms—**HEVC, palette mode, decision tree, Screen Content Coding**

I. Introduction

Videos containing non camera-captured content and artificial images or shapes usually termed as screen content video. Usage of screen content videos has increased in the past few years, with the increase in popularity of applications such as video conferencing, distance learning, cloud desktop/gaming, wireless displays etc.

The High Efficiency Video Coding (HEVC) standard [1] is a hybrid video codec launched by the Joint Collaborative Team on Video Coding (JCT-VC). The normal hybrid video coding approach may not provide high compression and good quality frames for screen content as it is targeted at natural images [2]. JCT-VC has launched a set of extensions to the HEVC standard called the SCC extensions [3] to effectively code screen content by exploiting some of the properties of screen content. The major tools introduced as part of the SCC extensions are the Palette mode and Intra Block Copy (IBC). Palette mode is an efficient method for representing blocks containing a small number of distinct colours which takes advantage of the fact that screen content data is usually sparse in colours. The encoding of such samples is done by constructing a palette table and representing each of the colour as a table entry and sending table index instead of entire pixel data. IBC is a predictive coding technique, where a block is predicted from neighbouring blocks in the same frame, which is efficient for video content having repetitive patterns.

Pavan Kumar G is with the Department of Electronics and Communication Engineering, National Institute of Technology Calicut, India (e-mail:gpavankumar8@gmail.com)

Dr. Sreelekha G is with the Department of Electronics and Communication Engineering, National Institute of Technology Calicut, India (e-mail: lekha@nitc.ac.in)

Although SCC techniques have managed to effectively compress screen content data, they come at the expense of increased computational complexity. Such complexity would be a deterrent for the successful application in various real time applications like remote desktop screen sharing, cloud gaming, wireless displays etc. Therefore there is a need to reduce the computational complexity by optimizing the algorithms involved and if necessary to explore alternative techniques.

There have been various attempts in literature that try to reduce computational complexity of SCC tools. The majority of the techniques aim to classify blocks as screen content or natural content by computing a metric or feature. Once classified, the screen content tools are applied on the screen content blocks. Lei *et al* [4] proposed a technique of classifying blocks into screen content and natural content by analyzing the histogram, intra mode decisions and smoothness of the block. Liu *et al* [5] proposed a technique for animation videos where they analyze the texture of the block using a metric based on DCT. A few methods have also adopted classifiers with multiple features to classify blocks. Support Vector Machine (SVM) [6], Decision trees [7] and Random forests [8] are some of the classifiers that have been used in the above context.

Apart from classification of CU blocks, certain techniques like [9] reduce computation time by improved IBC search techniques. Zhao *et al* proposed a technique called string matching [10] and Chen *et al* proposed intra line copy [11]. Zhang *et al* [12] developed an adaptive step-search algorithm for IBC in addition to a block depth prediction algorithm.

This paper proposes a novel attempt to reduce screen content coding computation time by classifying Coding Units (CUs) as Palette Mode (PLM) regions and Non Palette Mode (NPLM) regions. A Coding Unit (CU) is a basic unit of prediction in the HEVC encoding stream, it's size varying from 8×8 to 64×64. A decision tree classifier, which is computationally simple has been used to classify CUs into the PLM/NPLM classes. A collection of statistical features available in literature have been computed on the CUs and have been used to train the decision tree classifier. From the set of features, the decision tree automatically builds itself with features having good class separability.

The paper is organized as follows. Section II describes the methodology of the proposed classifier. The features that have been used as well as the classifier design aspects are discussed. The implementation of the classifier in the SCM-8.8 encoder [13] is also discussed. Section III discusses the experimental results as well as a comparison with state of the art techniques. The conclusion of this paper are presented in Section IV.

978-1-7281-4988-2/20/$31.00 ©2020 IEEE

II. Methodology

HEVC-SCC tools have been found to be time consuming and unfit for real time application. To reduce the time complexity, the approach of classifying a frame block CU into Screen Content Coding Unit (SCCU) and Natural Content Content Coding Unit (NCCU) has been adopted for the proposed technique. This approach saves time by avoiding SCC tools on the NCCUs, reducing the Rate Distortion Optimization (RDO) computation and at the same time preserves the quality and lowers bitrate by applying the SCC tools only on SCCUs. In the present work only palette mode tool has been studied and optimized. To develop the classifier, optimal features that discriminate between the two classes have to be identified and then a proper classifier has to be chosen. In the next subsection some features used for the present study have been described and the subsection following that describes classifier design.

A. Feature selection for classification

Screen content has generally got different characteristics than natural content video. The classifier needs proper features that discriminate between the two classes to properly classify the CU block. Some features that have been used in literature to differentiate SCCU and NCCU are discussed below.

1) **High frequency coefficients count using Discrete Cosine Transform (DCT)**
DCT is generally used to analyze the texture complexity of an image. Liu *et al* [5] used a metric K, which denotes the count of DCT coefficients below a threshold of 8. K is computed as

$$K = \sum_{j=0}^{7}\sum_{i=0}^{7} m_{ij} \quad \forall \quad i+j \neq 0$$
$$\text{where, } m_{ij} = \begin{cases} 1, & \text{if } |DCT_{ij}| \leq 8 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

DCT_{ij} is the ij^{th} DCT coefficient of the 8 x 8 DCT matrix.
Since CU size as per the HEVC standard can vary from 64 x 64 to 8 x 8, the metric K for a non 8 x 8 CU block is computed as a weighted average K_{avg} of all the 8 x 8 blocks in the larger CU. If the CU size is MxN then

$$K_{avg} = \sum_{m=0}^{M/8-1}\sum_{n=0}^{N/8-1} a_{mn}K_{mn} \tag{2}$$
$$\text{where, } a_{mn} = \frac{|DCT_{00}^{mn}|}{\sum_{m=0}^{M/8-1}\sum_{n=0}^{N/8-1}|DCT_{00}^{mn}|}$$

where K_{mn} is the K value for the mn^{th} 8 x 8 sub block of the CU block and DCT_{00}^{mn} corresponds to the DC coefficient of the mn^{th} sub block's DCT.

2) **Intensity difference**
Lei *et al* [4] proposed that screen content blocks have higher intensity differences than natural content blocks, because of sharp edges and colours. They suggest computing the intensity difference (I_d) as

$$I_d = V_{max} - V_{min} \tag{3}$$

where V_{max} and V_{min} are the maximum and minimum luminance values in a CU block.

3) **Histogram non zero pixel count**
The second feature that was proposed by Lei *et al* [4] was to count the number of non zero pixel values of the histogram of the block. Fig. 1 shows the typical histograms for a SCCU and an NCCU. Generally, the number of the non-zero histogram elements is high for NCCU and low for SCCU.

4) **Histogram peak values**
Another feature that was again proposed by Lei *et al* [4] based on the histogram shown in Fig. 1 is the observation that the sum of the top N ($N = 5$ suggested by author) histogram values is low in NCCU than SCCU because the intensity is distributed across many pixel values in NCCU. The summation P is given as

$$P = \sum_{i=0}^{N=5} \frac{n_i}{W \times H} \tag{4}$$

where n_i denotes the top i values of the histogram, W and H the Width and height of the CU block.

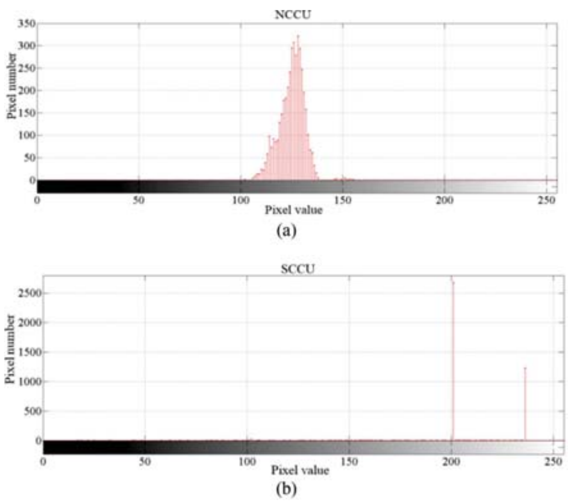

Fig. 1. Histogram of NCCU and SCCU [4]

5) **Neighbouring CU information**
If a block has been coded as palette mode, there is a high probability of neighbouring blocks being classified as palette mode [7]. For analysis purpose, the top and left side block palette mode information has been utilized. The neighbouring CU information feature PLM_{neigh} can be computed as

$$PLM_{neigh} = PLM_{left} + PLM_{top} \tag{5}$$

where PLM_{left} and PLM_{top} are the left and top CU palette mode decisions and take values 0 if block is non palette mode and 1 if block is palette mode. The neighbouring CU information feature for Intra mode $Intra_{neigh}$ has also been computed, as described above.

6) **Mean Absolute Deviation (MAD)**
MAD can be used to analyze whether the texture is smooth or rough [6]. The formula for MAD (X_T) is

$$X_T = \frac{1}{N_B} \sum_{i,j} |I(i,j) - \frac{1}{N_B} \sum_{i,j} I(i,j)| \tag{6}$$

where N_B is the number of elements in the block and $I(i,j)$ is the intensity of ij^{th} pixel in the block I

7) **High Gradient pixel Number (HGN)**
Pan *et al* [14] proposed HGN as a means to detect sharp edges in a CU. A pixel (i,j) is defined as a high gradient pixel if the luminance (Y) difference of the current pixel and one of the neighbor pixels is greater than a threshold Th. HGN is computed as the count of the number of high gradient pixels.

$$\begin{gathered} HGN = \sum_{i,j} p_{ij} \\ \text{where, } p_{ij} = \\ \begin{cases} 1, & \text{if } |Y_{i,j} - Y_{i\pm1,j}| \text{ or } |Y_{i,j} - Y_{i,j\pm1}| \geq Th \\ 0, & \text{otherwise} \end{cases} \end{gathered} \tag{7}$$

HGN for three different thresholds $Th = 4, 32, 64$ has been computed and the features have been named as $HGN4, HGN32, HGN64$ respectively.

B. Classifier design

There are several models to classify a given data for a particular application. In the context of the problem of NCCU/SCCU classification, computationally simple classifiers are required, because one of the main goals of the classifier to be designed is to decrease computational complexity in the video coding process. A simple classifier that has been used previously in video coding scenarios is the decision tree classifier [7].

Decision tree is a tree structure composed of a root node, internal nodes and leaf nodes. A decision tree solves a classification problem by testing a feature of the dataset at each internal node and finally arrive at the leaf nodes which assign a class label. The features and corresponding thresholds of a node are arrived through a training process.

1) Database generation: In order to train the classifier, a database containing the ground truth data has to be developed. In the HEVC-SCC codec, the three coding tools Intra, IBC and palette are all applied on a CU and the best tool is selected based on an RDO process. The RDO process selects the palette mode only if the rate distortion (RD) cost associated with palette mode for that particular block is lower than other modes. These mode decisions taken after the RDO process of the encoder can be extracted to form a database of each CU and it's corresponding mode (Intra/IBC/Palette) in which it was encoded. Some examples of frames and the corresponding CU blocks and the mode information extracted is shown in Fig. 2. The samples for the database have been obtained by encoding four videos SlideShow (Text with Graphics and Motion (TGM)) , ChinaSpeed (Animation (A)), SlideEditing (TGM) and BasketballDrillText (Mixed (M)) using the original SCM-8.8 encoder under the All Intra (AI) configuration, selecting the Quantization Parameter (QP) as 22, 27, 32 and 37. The four videos that are part of the JCT-VC test sequences [15] are selected such that they cover all the sub categories of screen content. To avoid redundant samples, ten non consecutive frames with different characteristics were extracted from the videos and encoded.

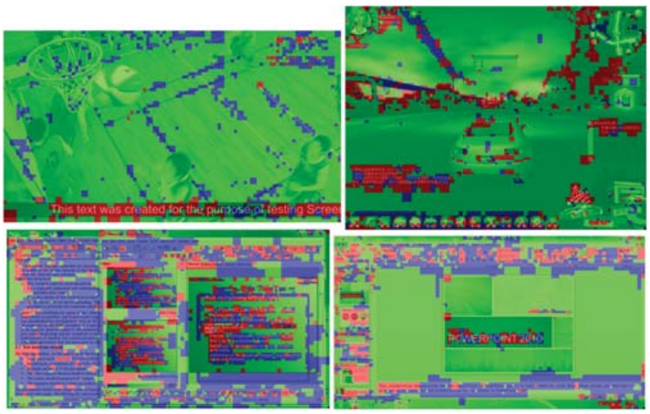

Fig. 2. Mode information extracted (Red - Palette mode, Green - Intra mode, Blue - IBC mode). Frames from test sequences (clockwise from top left): BasketballDrillText, ChinaSpeed, SlideShow, SlideEditing

2) Decision tree training: The features computed from the ground truth CU data are fed to train the decision tree classifier. At each node of the tree, the feature value which gives the best separability or least impurity is selected as the threshold for that particular node. The Gini impurity criterion [16] has been chosen as the tree split criterion.

There are several approaches for building a decision tree. First one is to build a single decision tree for all CU sizes. The feature values can be obtained by scaling the values of all CU sizes to an 8x8 CU. The second approach is to have different decision trees for each CU size as have been used in [7]. Increasing the decision tree levels beyond 10 was found to give only minor accuracy increase and overfit to the training samples, therefore maximum tree depth is fixed at 10.

A balanced dataset with equal samples is necessary to prevent the classifier getting biased to the majority class. In the database generated, there are a large number of blocks belonging to the Non Palette Mode (NPLM) class than the Palette Mode (PLM) class (around 10:1) as can be observed from Fig. 2. To balance the dataset for training phase, random undersampling of the NPLM class to as many samples are there in PLM class was done. The classifier is then trained on the balanced dataset using 5 fold cross-validation. An example of the decision tree generated after the training process is shown in Fig. 3.

3) Implementation in HEVC encoder: A simplified flow chart of the HEVC CU block encoding process with the proposed classifier inserted is shown in Fig. 4.

The encoding process starts by first dividing a frame into Coding Tree Units (CTUs) which are partitioned further into

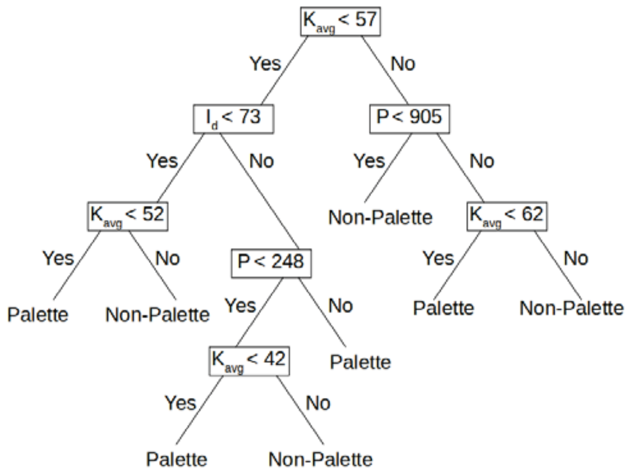

Fig. 3. Trained palette mode decision tree classifier

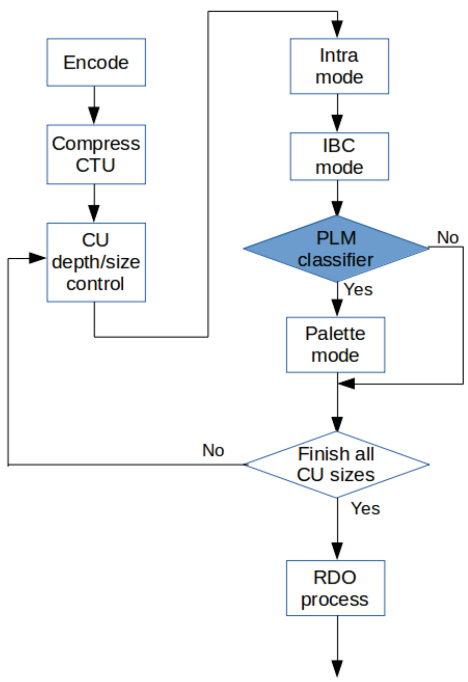

Fig. 4. HEVC CU encoding flowchart

Coding Units (CUs). The CU depth/size control block iteratively varies the size of the CU from 8 × 8 to 64 × 64. For any CU size, the first tool used to compress data is the Intra prediction coding tool, followed by the Intra Block Copy (IBC) tools and then finally the palette mode (PLM). In the proposed modification a palette mode classifier is inserted just before palette mode computations. If the CU is classified as Palette mode then Palette mode is enabled else it is disabled. After the palette mode tool the entire process iterates for different CU sizes. Finally a Rate Distortion Optimization (RDO) process decides the optimal mode and CU depth.

III. Results and Inferences

The proposed algorithm has been compared with the original SCM-8.8 reference software in the All Intra (AI) configuration to evaluate it's performance. The performance metrics used are the Bjontegaard-Delta Bit Rate (BDBR%), the Bjontegaard-Delta Peak Signal to Noise Ratio (BDPSNR(dB)) and Percent Time Saving (TS%) computed as

$$TS\% = \frac{T_{orig} - T_{mod}}{T_{orig}} \tag{8}$$

where T_{orig} is the time taken by the original SCM-8.8 encoder and T_{mod} is the time taken by the modified SCM-8.8 encoder which includes the PLM classifier. BDBR% indicates the change in bitrate with respect to the bitrate obtained by a reference codec. BDPSNR (dB) indicates the change in PSNR with respect to PSNR obtained by a reference codec. The reference codec in this case is the original SCM-8.8 codec. The BDBR and BDPSNR metrics have been computed by encoding the videos at QP = 22, 27, 32, 37 as per [15-17].

Table I shows the test videos chosen for experimentation. The first set belongs to the JCT-VC test sequences and the second set are self made videos. Training was performed on the four JCT-VC sequences and the rest of the videos have been used to check performance on non trained videos. All the experiments in this section have been run on an Intel Core i7-3770 processor based Dell Optiplex 7010 system running a 64-bit Ubuntu 18.04 OS with 16 GB RAM.

TABLE I
SCC Test Sequences

Sequence	Resolution	No. of frames encoded	YUV
SlideShow	1280 × 720	0-99	4:2:0
ChinaSpeed	1024 × 768	0-99	4:2:0
SlideEditing	1280 × 720	0-99	4:2:0
BasketballDrillText	832 × 480	0-99	4:2:0
Kimono	1920 × 1080	0-99	4:2:0
Browsing	1920 × 1080	0-99	4:2:0
Programmer	1920 × 1080	0-99	4:4:4
BrowsingNature	1920 × 1080	0-99	4:4:4

Table II shows the results with the Single Tree Classifier (STC) approach and the Multiple Tree Classifier (MTC), which uses different tree for different CU sizes. The average results for each method are shown and compared with Ref. [7] which uses classifiers for palette mode, IBC and Intra modes. The MTC approach has better quality metrics, but lesser time saving than the single tree approach. The palette only classifiers have good BDBR % and BDPSNR metrics, but the time savings are low compared to Ref. [7] and similar methods because this method focuses only on improving the palette mode. Other techniques generally apply optimizations on the other tools of SCC as well as other speed improvement techniques. To compare the performance of the proposed method along with the other coding tools, the palette mode classifier described in Ref. [7] (Implemented in SCM-8.3) was replaced by the proposed MTC. The results are shown in the column named Kuang modified and gives better quality than the unmodified method [7] with a small reduction in time saving.

TABLE II
COMPARISON AMONG DIFFERENT APPROACHES

Test Sequence	STC method TS%	BDBR/%	BDPSNR/dB	MTC method TS%	BDBR/%	BDPSNR/dB	Kuang [7] TS%	BDBR/%	BDPSNR/dB	Kuang modified TS%	BDBR/%	BDPSNR/dB
SlideShow	9.59	0.55	-0.04	13.54	0.29	-0.02	38.71	3.05	-0.24	35.96	2.38	-0.18
ChinaSpeed	20.04	0.57	-0.05	15.88	0.36	-0.03	45.29	1.95	-0.16	30.94	0.83	-0.07
SlideEditing	8.01	0.35	-0.05	5.00	0.21	-0.03	39.68	1.48	-0.20	31.82	0.75	-0.10
BasketballDrillText	23.33	0.47	-0.02	19.25	0.29	-0.01	51.46	2.67	-0.13	42.34	2.01	-0.10
Kimono	14.72	0.00	-0.00	13.08	0.00	-0.00	43.76	0.15	-0.01	35.60	0.14	-0.01
Browsing	5.17	0.99	-0.11	5.15	0.17	-0.02	41.87	0.96	-0.11	39.67	0.92	-0.11
Programmer	3.12	1.03	-0.13	5.36	0.67	-0.08	43.76	1.45	-0.18	44.21	1.94	-0.24
BrowsingNature	6.3	1.13	-0.11	5.89	0.63	-0.06	50.43	0.93	-0.09	47.10	1.86	-0.19
Average	**11.28**	**0.64**	**-0.07**	**10.39**	**0.33**	**-0.03**	**44.46**	**1.58**	**-0.14**	**38.86**	**1.35**	**-0.13**

Table III shows the results compared with some of the state of the art methods. Although the method does not have the best time savings, it generally outperforms the other methods in quality metrics without compromising much on the time savings.

TABLE III
COMPARISON WITH STATE OF THE ART TECHNIQUES - SLIDESHOW SEQUENCE

Sequence	TS%	BDBR/%	BDPSNR/%
Proposed	35.96	2.38	-0.18
Lei *et al* [4]	62.3	3	-0.22
Zhang *et al* [12]	50.58	2.1	-0.15
Liu *et al* [5]	12.49	1.68	-0.12
Zhang *et al* [17]	25	0.23	-
Xue *et al* [6]	40	1.61	-0.14
Tsang *et al* [8]	46.53	2.77	-
Kuang *et al* [7]	35.67	3.76	-

IV. CONCLUSION

The paper discussed the application of a decision tree based classifier to reduce computation time in HEVC-SCC. The classifier is trained on a set of statistical features extracted from the CUs and classifies CUs as PLM/NPLM blocks. Palette mode technique is applied only on the PLM blocks which helps in reducing time complexity at the same without degrading the quality much. Experimental results show an average time saving of 38.86% with an average BDBR increase of 1.353% and an average BDPSNR degradation of 0.125 dB.

This method can be further extended to SCC tools like intra mode decisions and IBC mode. New features that represent the nature and properties of SCC blocks can be derived. Other advanced and computationally simple classifiers like Bagged trees or random forests may provide improved results with the method described in this paper. A well designed classifier could help in reducing the exhaustive RDO process thus reducing the time and maintaining good quality.

REFERENCES

[1] G. J. Sullivan, J. Ohm, W. 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.

[2] W. Peng et al., "Overview of Screen Content Video Coding: Technologies, Standards, and Beyond," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 393-408, Dec. 2016.

[3] J. Xu, R. Joshi and R. A. Cohen, "Overview of the Emerging HEVC Screen Content Coding Extension," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50-62, Jan. 2016.

[4] J. Lei, D. Li, Z. Pan, Z. Sun, S. Kwong and C. Hou, "Fast Intra Prediction Based on Content Property Analysis for Low Complexity HEVC-Based Screen Content Coding," in *IEEE Transactions on Broadcasting*, vol. 63, no. 1, pp. 48-58, March 2017.

[5] Y. Liu, C. Fang, J. Sun and X. Huang, "Fast Palette Mode Decision Methods for Coding Game Videos With HEVC-SCC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3061-3067, Oct. 2019.

[6] Y. Xue, X. Wang, L. Zhu, Z. Pan and S. Kwong, "Fast Coding Unit Decision for Intra Screen Content Coding Based on Ensemble Learning," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 8543-8547.

[7] W. Kuang, Y. Chan, S. Tsang and W. Siu, "Machine Learning Based Fast Intra Mode Decision for HEVC Screen Content Coding Via Decision Trees," in *IEEE Transactions on Circuits and Systems for Video Technology*.

[8] S. Tsang, Y. Chan and W. Kuang, "Mode Skipping for HEVC Screen Content Coding via Random Forest," in *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2433-2446, Oct. 2019.

[9] S. Tsang, Y. Chan and W. Siu, "Hash based fast local search for Intra Block Copy (IntraBC) mode in HEVC screen content coding," *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Hong Kong, 2015, pp. 396-400.

[10] L. Zhao, K. Zhou, J. Guo, S. Wang and T. Lin, "A Universal String Matching Approach to Screen Content Coding," in *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 796-809, April 2018.

[11] C. Chen and W. Peng, "Intra Line Copy for HEVC Screen Content Intra-Picture Prediction," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1568-1579, July 2017.

[12] H. Zhang et al., "Fast intra mode decision and block matching for HEVC screen content compression," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Shanghai, China, 2016, pp. 1377–1381.

[13] HM-16.20 + SCM-8.8, HEVC test model version 16.20 screen content model version 8.8, [Online] Available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.20+SCM-8.8/

[14] Z. Pan, H. Shen, Y. Lu, S. Li, and N. Yu, "A low-complexity screen compression scheme for interactive screen sharing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 6, pp. 949–960, Jun. 2013.

[15] H. Yu, R. Cohen, K. Rapaka, and J. Xu, "Common test conditions for screen content coding," in *Document of Joint Collaborative Team on Video Coding, JCTVC-U1015*, Warsaw, Poland, June 2015.

[16] P. Sen, "Gini diversity index, Hamming distance, and curse of dimensionality", *Metron-International Journal of Statistics*, vol.63, no. 3, pp. 329-349, 2005.

[17] M. Zhang, S. Wang and B. Li, "Selective motion estimation strategy based on content classification for HEVC screen content coding," *2017 IEEE International Conference on Image Processing (ICIP), Beijing*, 2017, pp. 2463-2467.