Ultra Fast Screen Content Coding via Random Forest

Sik-Ho Tsang, Ngai-Wing Kwong, and Yui-Lam Chan*
The Hong Kong Polytechnic University, Hong Kong, China
*E-mail: enylchan@polyu.edu.hk

Abstract— **Screen content coding (SCC) is an extension to High Efficiency Video Coding (HEVC) used to compress screen content videos. Besides the conventional intra (INTRA) mode, new coding tools, intra block copy (IBC), palette (PLT) modes, and adaptive color-space transform (ACT) are introduced to encode screen content (SC) such as texts and graphics. However, the use of IBC, PLT and ACT increases the encoder complexity though coding efficiency can be improved. While there were numerous approaches for fast INTRA, fast IBC and fast PLT at coding unit (CU) level, we propose fast prediction unit (PU) decisions for INTRA and IBC. In addition, fast ACT approach is also proposed. Both approaches reduce the encoder complexity of SCC by making use of SC characteristics, neighbor correlations, and intermediate cost information via random forest (RF). Experimental results show that, with also fast CU mode decision approaches at CU level, our proposed approach, namely HoFastRF, can obtain 51.29% average encoding time reduction with only 1.45% increase in Bjøntegaard delta bitrate (BD-rate). It is the first fast SCC approach to obtain over 50% average encoding time reduction with less than 1.5% increase in BD-rate.**

I. Introduction

With the proliferation of networking and thin-client technology, computer screen sharing applications such as remote desktop, video conferencing with documents or slides sharing, and so on have become more and more popular. In the future, there will be more cloud services using screen sharing technology [1]. This type of video always contains a mixture of camera-captured content (CC) and screen content (SC). These applications induce a huge demand for the efficient compression of SC. In January 2014, there was Call for Proposal (CfP) [2] of screen content coding (SCC) [3]-[6] as the extension to High Efficiency Video Coding (HEVC) [7]-[9] by the Joint Collaborative Team on Video Coding (JCT-VC).

SC is the video content containing computer-generated content such as texts, computer graphics and graphical user interface while CC is the video content captured by camera. Videos can contain a mixture of SC and CC, as in Fig. 1. CC can be encoded by HEVC efficiently. However, SC has been proved to have discontinuous-tone characteristics [10]-[18], which is different from CC, such as complex structure with sharp edges, limited number of colors and sometimes high contrast between colors like texts.

The conventional HEVC intra (INTRA) mode [19]-[20] shown in Fig. 2(a) uses the neighbor boundary pixels to predict a coding unit (CU) with 33 directional predictions plus planar and DC predictions [19]. To reduce the complexity, rough mode decision (RMD) [20] is performed to select a subset of intra prediction candidates first. Then the optimal one is chosen by rate distortion optimization (RDO) where the full rate-distortion (RD) cost for every candidate in the subset is estimated. However, it cannot efficiently encode the CUs with screen content such as the example shown in Fig. 2(a) since neighbor boundary pixels cannot predict the pixels with abrupt change within the CU, even with the support of various prediction unit (PU) sizes. Hence, two new coding tools are introduced in SCC to solve this problem. They are palette (PLT) mode [21]-[22] and intra block copy (IBC) mode [23]-[25].

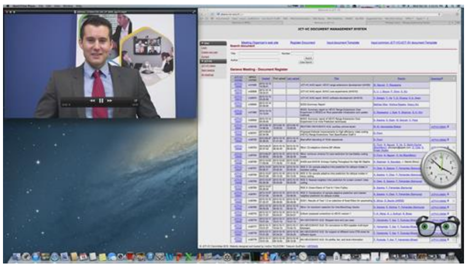
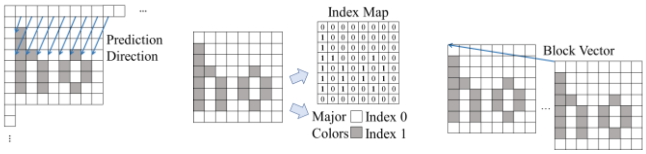


Fig. 1 A sequence with both screen content and camera-captured content.



(a) (b) (c)

Fig. 2 Illustrations of (a) INTRA, (b) PLT, and (c) IBC modes.

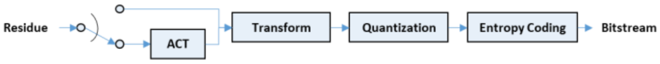


Fig. 3 Illustrations of Adaptive Color-space Transform (ACT).

For PLT [21]-[22] as demonstrated in Fig. 2(b), a CU is separated into color data and structural data. The color data consists of few major colors which are predicted from color table or from neighbor CUs. When the colors are not in the major color tables, they are regarded as escape colors and are explicitly coded. The structural data is then represented by an index map and the corresponding indices are entropy coded. Thus, PLT helps to encode SC which contains few colors like texts and icons.

IBC mode [23]-[25], as in Fig. 2(c), is a block matching technique to find the repeating patterns within the same frame which frequently occurs in SCs. For instance, repeating numbers and characters can be found within a document and spreadsheet. If IBC is used by one particular CU, each PU within the CU is encoded with a block vector (BV), as well as the residual signal of that CU which is similar to an inter mode in inter-frame motion estimation.

Besides the new coding modes, PLT and IBC, there is also a new adaptive color-space transform (ACT) [26] for encoding

The work described in this paper is supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant No. PolyU 152069/18E).

SC, as in Fig. 3. The coding residue is adaptively converted into a different color space to reduce the cross component redundancy prior to transform and quantization.

However, with additional coding tools, the computational complexity of SCC must be higher than that of HEVC. Therefore, numerous fast approaches [27]-[26] were proposed to reduce the complexity of SCC. Zhang *et al.* [27] proposed to exploit the CU mode from the collocated CUs of the previous frame. This method is mainly targeted for stationary CUs. Lei *et al.* [28] proposed to utilize the content property analysis, bits per pixel information as well as neighbor and collocated CUs' depth information to classify CUs into screen content CU (SCCU) or camera-captured content CU (CCCU), and then simplify INTRA mode, mode elimination and fast CU partitioning with numerous pre-defined thresholds. By checking whenever neighbor boundary pixels are exactly the same, [29] proposed to skip the RMD [10] and RDO for reducing the complexity of INTRA. IBC and PLT are also skipped when the residual error obtained by INTRA is zero. [30] proposed to use Bayesian decision rules based on corner point detection for fast CU mode and fast CU partitioning plus online learning approaches based on scene change detection. In [31], conditional probability based CU type classification was proposed to classify CU into several types for fast CU mode and fast CU partitioning. For machine learning approach, [32]-[34] used decision trees to reduce the encoding time of SCC. [35] proposed to use neural network while [36]-[37] proposed to use convolutional neural network for reducing the encoding time where [37] is proved to be faster than [38]. However, they mainly focus on fast CU mode and fast CU partitioning only.

In this paper, we propose a fast PU approach and a fast ACT approach using random forest (RF). By combining the RF based fast CU mode decision approach [39], our proposed approach, namely HoFastRF, obtains 51.29% encoding time reduction with only 1.45% increase in Bjøntegaard delta bitrate (BD-rate) [40]. With the best of our knowledge, it is the first fast SCC approach to obtain over 50% average encoding time reduction with less than 1.5% increase in BD-rate.

II. SCC Intra Coding

In SCC intra coding, each video frame is split into coding tree units (CTUs) of 64×64 pixels. A recursive quad-tree coding structure is applied to each CTU. For each CTU, it has a size of *2N*×*2N* and can be split into four smaller CUs of N×N, namely sub-CUs. This splitting process is repeated recursively until the smallest CU (SCU) size of 8×8 is reached. In SCC, *2N* can be chosen as 64, 32, 16 or 8. To find the best combination of CU sizes within a CTU, the encoder performs the RDO process that chooses the optimal CU size by comparing the RD cost obtained by the current CU and the sum of RD costs obtained by its four sub-CUs. In other words, for each CU size, the mode that obtains the least RD cost among INTRA, IBC and PLT, is selected as the optimal mode, m^*, as follows:

$$\begin{aligned} &J_m = D_m + \lambda \cdot R_m \\ &m^* = \arg \min_{m \in M} (J_m) \\ &M = \{INTRA, IBC, PLT\} \end{aligned} \tag{1}$$

where D_m, R_m and J_m, respectively, are the distortion, coding rate and RD cost obtained by the mode m, and λ is the Lagrangian multiplier controlled by the quantization parameter (QP). The CU is then split if the sum of the RD costs of the four sub-CUs, $J_{N,i}$, is smaller than the cost of the current CU, J_{2N}, or otherwise the CU is not split, as in:

$$\begin{cases} \text{Split} & \text{, if } \sum_{i=0}^{3} J_{N,i} < J_{2N} \\ \text{Not Split} & \text{, Otherwise} \end{cases} \tag{2}$$

where *i* is the index of sub-CU. As a result, the complexity of encoding a CTU is largely increased in SCC compared with HEVC because there are additional IBC and PLT modes for each CU candidate.

In addition, within a CU, there are various PU sizes for different coding modes. For INTRA, there are *2N*×*2N* and N×N PU modes. That means, when it is coded as *2N*×*2N*, single intra mode is used to encode the CU as a whole. Otherwise, four intra modes are used to encode the CU when it is coded as *N*×*N*. For IBC, there are *2N*×*2N*, 2*N*×*N* and *N*x2*N* PU modes. That means, there is one BV for *2N*×*2N* PU mode and there are two BVs for 2*N*×*N* and *N*x2*N* PU modes. Moreover, ACT is performed for INTRA and IBC modes. RDO is performed to choose whether ACT should be applied to the CU or not.

III. Proposed Fast PU and Fast ACT Approaches

A. Feature Selection

Three major types of features are extracted. They are screen content characteristics, neighbor CU modes, and intermediate cost information.

1) Screen Content Characteristics

As compared with CC, SC has different characteristics that has complex structure with sharp edges, has only limited number of colors and sometimes has high contrast between colors. Some SC also contain mainly horizontal or vertical edges. The horizontal activity Act_H and vertical activity Act_V are extracted as features as follows:

$$\begin{aligned} &Act_H = \sum_{p_Y \in P} |p_Y(i,j) - p_Y(i-1,j)| \\ &Act_V = \sum_{p_Y \in P} |p_Y(i,j) - p_Y(i,j-1)| \end{aligned} \tag{3}$$

where P is the set of pixels within the CU and $p_Y(i,j)$ is the luminance value at the relative location (i,j) within the CU. Act_H and Act_V are used because SC usually contains sharp edges.

In addition, the variance of a CU is also selected as a feature in the following:

$$Var = \frac{1}{(2N)^2} \sum_{p_Y \in P} \left(p_Y(i,j) - \bar{P}\right)^2 \tag{4}$$

where $\bar{P}$ is the mean intensity of all pixels in the CU. Variance can help to measure the diversity of pixels within the CU. With larger CU variance, smaller PU sizes are encouraged to encode the CU, or the chance of using IBC and PLT is larger.

Moreover, the number of high-gradient pixels is estimated as a feature. A pixel is defined as a high-gradient pixel if the luminance difference between itself and one of the neighboring pixels is larger than a pre-defined threshold TH_{HG} as below:

$$
\begin{aligned}
&p_Y(i,j) \in P_{HG}(TH_{HG}) \text{ if} \\
&\left|p_Y(i,j) - p_Y(i \pm 1, j)\right| > TH_{HG}, \text{ or} \\
&\left|p_Y(i,j) - p_Y(i, j \pm 1)\right| > TH_{HG}. \\
&N_{HG}(TH_{HG}) = \left|P_{HG}(TH_{HG})\right|
\end{aligned}
\tag{5}
$$

where $P_{HG}(TH_{HG})$ is the set of high-gradient pixels within the CU with different values of TH_{HG}. They are set to 4, 8, 16, and 32 in this paper. The number of elements in $P_{HG}(TH_{HG})$, $|P_{HG}(TH_{HG})|$, are counted as $N_{HG}(TH_{HG})$ which represents the number of high-gradient pixels. $N_{HG}(TH_{HG})$ with different values of TH_{HG}.

Besides, by concatenating the components of the luminance and chrominance values, i.e. p_Y, p_U and p_V, the number of distinct colors, N_{DC}, is counted since SCs often contain limited number of colors. Higher values of N_{DC} within the CU decreases the chance of becoming SC because CC normally contains sensor noise.

Finally, the number of background colors, N_{BC}, is also considered as a feature by concatenating p_Y, p_U and p_V. The background color is defined to be the most frequently occurred color within the CU. Higher values of N_{BC} within the CU is more likely to be the SC as the SC is computer captured which has no sensor noise.

2) *Neighbor CU Modes*

It is logical that when neighbor CUs contain SC, the current CU is likely to be SC. For instance, a word document which contains large amount of texts. Therefore, the modes of left, above, above right, left bottom and above left CUs are extracted to count the numbers of neighbor CU modes that are INTRA, IBC and PLT, respectively, as features. Besides, the number of unavailable modes due to frame boundary is also counted as a feature.

3) *Intermediate Cost Information*

The best mode m_{best} just before the mode being checked or skipped, as well as the corresponding RD cost $J_{m_{best}}$, distortion $D_{m_{best}}$, and the coding rate $R_{m_{best}}$ as in (1), are also selected as features. This is because if the RD cost, distortion or coding rate of the best mode is very small, it is able to give an insight to the encoder that the previously checked modes may be sufficiently effective that the current mode can be skipped. In contrast to CC, the chance of getting the exact or very close match within the same frame of SC is higher resulting in very low $J_{m_{best}}$, $D_{m_{best}}$, and $R_{m_{best}}$. Therefore, we collect the latest intermediate cost information, i.e. the RD cost, distortion and coding rate of the best mode just before the target mode to decide whether it should be skipped or not. For example, if a CU is already well coded by large PU size with a very low RD cost, it is likely to skip the all the small PU sizes with negligible impact to the coding efficiency. Or if a CU is already well coded without the use of ACT, the checking of ACT can be skipped without large impact to the coding efficiency.

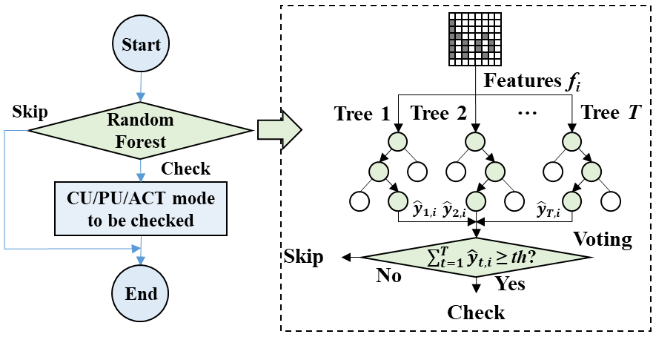


Fig. 4   Illustrations of our proposed random forest (RF) approach.

B. *Feature Extraction and Hyperparameter Tuning*

To train the RFs with the features mentioned in the previous sub-section, the training set is totally independent of the test set suggested in the common test condition (CTC) [41]. The training set are BigDuck, CadWaveform, Chinese-DocumentEditing, EBULupo-Candlelight, KristenAndSara-Screen, MissionControlClip1, ParkScene, PcbLayout, Real-TimeData, Seeking, VenueVu, Viking, and WordEditing. The validation set is VideoConferencingDocSharing. These sequences are either from JCT-VC [42] or from Joint Video Exploration Team (JVET) for Versatile Video Coding (VVC) [43]. Only the first frames of each second from the sequences are extracted and formed as frame-skipped sequences for training. The training set is acquired by encoding the training set using SCM-8.3 [44] with QPs of 22, 27, 32 and 37 and AI configuration which are recommended by CTC [41]. The CUs that are encoded into the bitstream are treated as positive while others are treated as negative samples.

Our proposed RFs are trained using the random forest package [45] in a free statistical computing language and software called R [46]. Suppose there are T number of trees for a RF and d number of features for each RF, $\sqrt{d}$ number of features (with rounded down) at each node of a tree are selected randomly out of d features with replacement. Each node is split using the best feature based on the Gini impurity or information gain among that subset of features. The split is terminated if either one of the leaf nodes, i.e. the left and right leaf nodes, has the number of samples smaller than s of the total samples to limit the depth of each tree [47]-[49]. After that, a threshold th is used in the voting to make skipping decision:

$$
\begin{cases}
\text{Skip} & \text{, if } \sum_{t=1}^{T} \hat{y}_{t,i} < th \\
\text{Not Skip} & \text{, Otherwise}
\end{cases}
\tag{6}
$$

where $\hat{y}_{t,i}$ is the predicted labels for i-th CU at tree t, with $\hat{y}_{t,i} \in \{0,1\}$. It is also illustrated in Fig. 4. The selection of

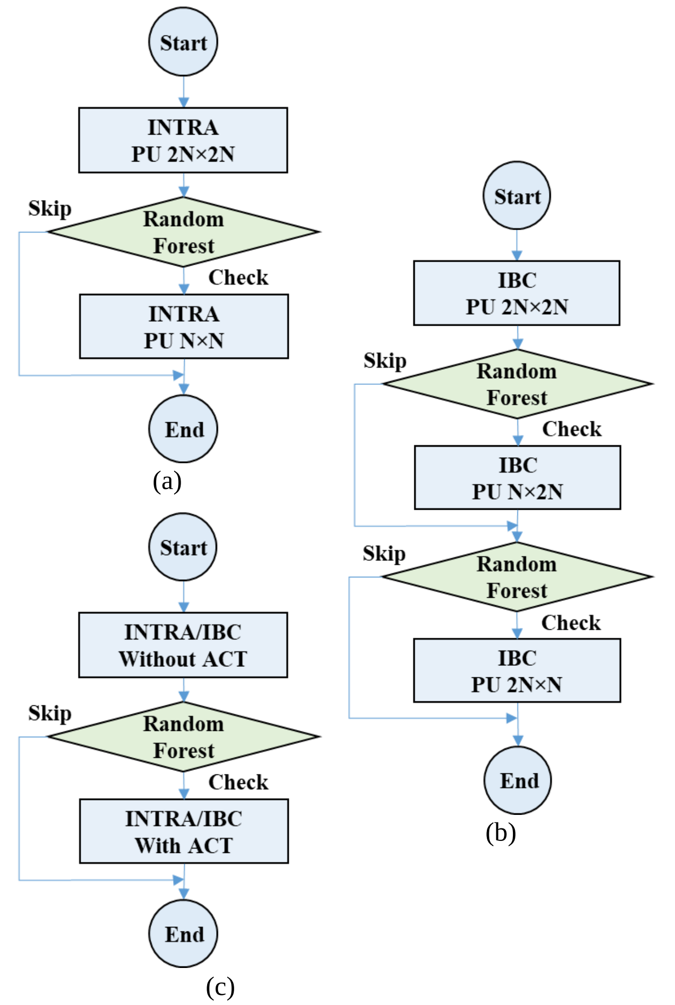


Fig. 5 Fast PU decisions for (a) INTRA, and (b) IBC, as well as (c) Fast ACT decision for INTRA and IBC using our proposed RF.

these model parameters, including *T*, *s*, and *th*, is optimized by hyperparameter tuning in our training process. To perform hyperparameter tuning of a RF, the out of bag (OOB) error rate is measured. The one with the lowest OOB error rate is chosen as optimal. For each tree, 63.2%, roughly two-third, of the total training samples *S*, by bootstrap sampling, are input for training. After training, the leftover 36.8% samples, $S_{OOB,t}$, are used for calculating the misclassification rate of *t*-th tree, i.e. the out of bag (OOB) error rate $ER_{OOB,t}$ where *t* is the tree index. And the OOB error rate of a RF $ER_{OOB,RF}$ can be obtained by getting the average OOB error rates from all trees as follows:

$$ER_{OOB,t} = \sum_{i \in S_{OOB,t}} \left|\hat{y}_{t,i} - y_i\right| / \left|S_{OOB,t}\right|$$
$$ER_{OOB,RF} = \frac{1}{T}\sum_{t=1}^{T} ER_{\ OOB,t} \qquad (7)$$

where y_i and $\hat{y}_{t,i}$ are the true and predicted labels respectively, and $\left|S_{OOB,t}\right|$ is the number of samples in the set $S_{OOB,t}$. That is the merit of random feature subspace and random data subset which makes RF not greedy [47]-[49].

TABLE I
BD-RATE (%) AND ΔTIME (%) OF OUR PROPOSED APPROACHES AGAINST THE CONVENTIONAL SCC (M: MIXED SC AND CC, TGM: TEXTS AND GRAPHICS WITH MOTIONS, A: ANIMATION, CC: CAMERA CAPTURED)

Sequences	Type	FastPU		FastPU+FastACT	
		BD-rate	ΔTime	BD-rate	ΔTime
Basketball_Screen	M	0.33	-6.34	0.24	-15.24
MissionControlClip2	M	0.22	-12.52	0.28	-20.05
MissionControlClip3	M	0.35	-8.77	0.38	-19.99
ChineseEditing	TGM	0.08	-8.67	0.02	-16.03
sc_console	TGM	0.10	-6.76	0.02	-21.39
sc_desktop	TGM	0.30	-3.79	0.06	-22.37
sc_flyingGraphics	TGM	0.65	-6.97	0.09	-21.13
sc_map	TGM	0.58	-6.17	0.20	-11.98
sc_programming	TGM	0.45	-9.03	0.21	-17.05
sc_SlideShow	TGM	0.48	-20.25	0.25	-23.65
sc_web_browsing	TGM	0.04	-4.64	-0.27	-24.44
sc_robot	A	0.20	-17.45	1.51	-26.12
EBURainFruits	CC	0.08	-20.76	0.70	-30.64
Kimono1	CC	0.01	-16.54	0.06	-29.17
Average (All)		**0.28**	**-10.62**	**0.27**	**-21.38**

TABLE II
AVERAGE BD-RATE (%) AND ΔTIME (%) OF VARIOUS APPROACHES AGAINST THE CONVENTIONAL SCC

Approaches	BD-rate	ΔTime	Approaches	BD-rate	ΔTime
HoFastRF (FastPU+ FastACT+FastCUMode)	**1.45**	**-51.29**	[32]	1.47	-27.72
[27]	1.17	-33.30	[33]	1.59	-44.50
[28]	2.22	-32.42	[34]	1.42	-47.62
[29]	1.52	-17.07	[35]	1.36	-49.33
[30]	1.08	-36.69	[36]	2.67	-53.21
[31]	1.20	-35.95	[37]	1.94	-53.44

With our trained RFs, the RF based fast PU and fast ACT decisions for INTRA and IBC are depicted in Fig. 5. The checking of INTRA with PU *N*×*N* can be skipped by RF as in Fig. 5(a). The checking of IBC with PU *N*×2*N* and with PU 2*N*×*N* can be skipped by RF as in Fig. 5(b). And the checking of ACT for INTRA or IBC can be skipped by RF as in Fig. 5(c).

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed mode skipping approaches using RF, we have performed simulations using the HEVC SCC reference software SCM-8.3 [44] on the testing sequences mentioned in CTC [41] with all intra (AI) configuration. The experiments were conducted on the Dell Precision T1700 computer with an Intel i7-4770 3.40GHz processor and 16GB memory.

Table I tabulates the BD-rate [40] and encoding time difference of our proposed approaches against the conventional SCC where FastPU denotes the proposed Fast PU approach and FastACT denotes the proposed fast ACT approach. With FastPU, 10.62% encoding time reduction is obtained with only 0.28% increase in BD-rate. With both FastPU and FastACT, 21.38% encoding time reduction is obtained with only 0.27% increase in BD-rate.

Table II tabulates the average BD-rate and encoding time difference of various fast SCC approaches against the conventional SCC. With FastPU+FastACT plus the fast CU mode decision using RF (FastCUMode) [39] as well, namely HoFastRF, 51.29% encoding time reduction is obtained with only 1.45% increase in BD-rate. Compared with the results of [27]-[37], to the best of our knowledge, it is the first fast SCC approach to obtain over 50% average encoding time reduction with less than 1.5% increase in BD-rate.

V. Conclusions

SCC is the extension to HEVC for efficient screen content video compression. However, the use of IBC, PLT and ACT increases the encoder complexity though coding efficiency can be improved. We propose fast PU decisions for INTRA and IBC. In addition, fast ACT approach is also proposed. Experimental results show that, our fast PU and fast ACT approaches plus the fast mode decision approaches at CU level via RFs, our proposed approach, HoFastRF, can obtain 51.29% average encoding time reduction with only 1.45% increase in BD-rate. It is the first fast SCC approach to obtain over 50% average encoding time reduction with less than 1.5% increase in BD-rate.

References

[1] Y. Lu, S. Li, and H. Shen, "Virtualized Screen: A Third Element for Cloud-Mobile Convergence," *IEEE Multimedia*, vol. 18, no. 2, pp. 4–11, Feb. 2011.

[2] "Joint Call for Proposals for Coding of Screen Content," *ISO/IEC JTC1/SC29/WG11*, N14175, San Jose, U.S.A., Jan. 2014.

[3] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the Emerging HEVC Screen Content Coding Extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50-62, Jan. 2015.

[4] W. H. Peng, F. G. Walls, R. A. Cohen, J. Xu, J. Ostermann, A. Maclnnis, and T. Lin, "Overview of Screen Content Video Coding: Technologies, Standards, and Beyond," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 393-408, Dec. 2016.

[5] S.-H. Tsang, Y.-L. Chan, and W. Kuang, "Standard Compliant Light Field Lenslet Image Coding Model Using Enhanced Screen Content Coding Framework," *J. Electron. Imaging*, vol. 28, no. 5, 053027, pp. 1-12, Oct. 2019.

[6] S.-H. Tsang, Y.-L. Chan, and W. Kuang, "Standard-Compliant HEVC Screen Content Coding for Raw Light Field Image Coding," Accepted, To Be Appeared, *Proc. Int. Conf. Signal Process. Commun. Syst.*, pp.1-6, Surfers Paradise, Gold Coast, Australia, Dec. 2019.

[7] G. J. Sullivan, J. Ohm, W. J. Han, and T. Wiegand. "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[8] 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)," *IEEE J. Sel. Topics. Signal Process*, vol. 7, no. 6, pp. 1001-1016, Dec. 2013.

[9] T.-K. Lee, Y.-L. Chan, and W.-C. Siu, "Adaptive Search Range for HEVC Motion Estimation based on Depth Information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2216-2230, Oct. 2017.

[10] T. Lin, Peijun Zhang, Shuhui Wang, Kailun Zhou, and Xianyi Chen, "Mixed Chroma Sampling-Rate High Efficiency Video Coding for Full-Chroma Screen Content," *IEEE Trans. Circuits Syst. Video Technol.*, vol.23, no.1, pp. 173-185, Jan. 2013.

[11] S. Wang, K. Gu, S. Ma, and W. Gao, "Joint Chroma Downsampling and Upsampling for Screen Content Image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1595-1609, Sep. 2015.

[12] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast and Efficient Intra Coding Techniques for Smooth Regions in Screen Content Coding Based on Boundary Prediction Samples," *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, pp.1409-1413, Brisbane, Australia, Apr. 2015.

[13] W. Kuang, S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast Mode Decision Algorithm for HEVC Screen Content Intra Coding," in *Proc. of Int. Conf. on Image Process.*, pp. 2473-2477, Beijing, China, Sept. 2017.

[14] S.-H. Tsang, W. Kuang, Y.-L. Chan and W.-C. Siu, "Fast HEVC screen content coding by skipping unnecessary checking of intra block copy mode based on CU activity and gradient," *Proc. APSIPA Annual Summit Conf.*, pp.1–5, Jeju, Korea, Dec. 2016.

[15] W. Kuang, Y.-L. Chan, and S.-H. Tsang, "Efficient Mode Decision for HEVC Screen Content Coding by Content Analysis," Accepted, To Be Appeared, *Proc. Int. Conf. Signal Process. Commun. Syst.*, pp.1-5, Surfers Paradise, Gold Coast, Australia, Dec. 2019.

[16] W. Kuang, Y.-L. Chan, and S.-H. Tsang, "Efficient Intra Bitrate Transcoding for Screen Content Coding Based on Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 107211-107224, Aug. 2019.

[17] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Fast HEVC to SCC Transcoder by Early CU Partitioning Termination and Decision Tree Based Flexible Mode Decision for Intra-Frame Coding," *IEEE Access*, vol. 7, pp. 8773-8788, Jan. 2019.

[18] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Fast HEVC to SCC Transcoding Based on Decision Trees," in Proc. IEEE Int. Conf. Multimedia Expo (ICME), pp.1-6, San Diego, California, USA., Jul. 2018.

[19] J. Lainema, F. Bossen, W. J. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol.22, no.12, pp. 1792-1801, Dec. 2012.

[20] L. Zhao, L. Zhang, S. Ma, and D. Zhao, "Fast Mode Decision Algorithm for Intra Prediction in HEVC," *Proc. IEEE Visual Comm. Image Process. (VCIP)*, pp. 1-4, Tainan, Taiwan, Nov. 2011.

[21] W. Pu, *et al.*,, "Palette Mode Coding in HEVC Screen Content Coding Extension," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 420-432, Dec. 2016.

[22] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Exploiting Inter-layer Correlations in Scalable HEVC for the Support of Screen Content Videos," *Proc. Int. Conf. Digital Signal Process.*, pp.888-892, Hong Kong, China, Aug. 2014.

[23] X. Xu, *et al.*,, "Intra Block Copy in HEVC Screen Content Coding Extensions," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 409-419, Dec. 2016.

[24] S.-H. Tsang, W. Kuang, Y.-L. Chan, and W.-C. Siu, "Decoder Side Merge Mode and AMVP in HEVC Screen Content Coding," *Proc. IEEE Int. Conf. Image Process.*, pp.260-264, Beijing, China, Sep. 2017.

[25] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Hash Based Fast Local Search for Intra Block Copy (IntraBC) Mode in HEVC Screen Content Coding," *Proc. APSIPA Annual Summit Conf.*, pp.396-400, Hong Kong, China, Dec. 2015.

[26] L. Zhang, *et al.*, “Adaptive Color-Space Transform in HEVC Screen Content Coding,” *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 446-459, Dec. 2016.

[27] H. Zhang, Q. Zhou, N. Shi, F. Yang, X. Feng, and Z. Ma, “Fast Intra Mode Decision and Block Matching for HEVC Screen Content Compression,” *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, pp. 1377-1381, Shanghai, China, Mar. 2016.

[28] 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,” *IEEE Trans. Broadcasting*, vol. 63, no. 1, pp. 48-58, Mar. 2017.

[29] S.-H. Tsang, Y.-L. Chan, W. Kuang, and W.-C. Siu, “Reduced-Complexity Intra Block Copy (IntraBC) Mode with Early CU Splitting and Pruning for HEVC Screen Content Coding,” *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 269-283, Feb. 2019.

[30] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “Online-Learning-Based Bayesian Decision Rule for Fast Intra Mode and CU Partitioning Algorithm in HEVC Screen Content Coding,” *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 170-185, Jan. 2020.

[31] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “Fast Intraprediction for High-Efficiency Video Coding Screen Content Coding by Content Analysis and Dynamic Thresholding,” *J. Electron. Imaging*, vol. 27, no. 5, 053029, pp. 1-18, Oct. 2018.

[32] F. Duanmu, Z. Ma, and Y. Wang, “Fast Mode and Partition Decision Using Machine Learning for Intra-Frame Coding in HEVC Screen Content Coding Extension,” *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 517-531, Dec. 2016.

[33] H. Yang, L. Shen, and P. An, “Efficient screen content intra coding based on statistical learning,” *J. Signal Process.: Image Commun.*, vol. 62, pp. 74-81, Mar. 2018.

[34] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “Machine Learning Based Fast Intra Mode Decision for HEVC Screen Content Coding Via Decision Trees”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1481-1496, May 2020.

[35] C. Huang, Z. Peng, F. Chen, Q. Jiang, G. Jiang, and Q. Hu, “Efficient CU and PU Decision Based on Neural Network and Gray Level Co-Occurrence Matrix for Intra Prediction of Screen Content Coding,” *IEEE Access*, vol. 6, pp. 46643-46655, Aug. 2018.

[36] L. Zhao, Z. Wei, W. Cai, W. Wang, L. Zeng, and J. Chen, “Efficient Screen Content Coding Based on Convolutional Neural Network Guided by a Large-Scale Database,” 2019 IEEE *Proc. Int. Conf. Image Process. (ICIP)*, pp. 2656-2660, Taipei, Taiwan, Sep. 2019.

[37] W. Kuang, Y.-L. Chan, and S.-H. Tsang, “Low-Complexity Intra Prediction for Screen Content Coding by Convolutional Neural Network,” *Proc. IEEE Int. Symp. Circuits Syst.*, Accepted, To Be Appeared, Seville, Spain, Oct. 2020.

[38] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “DeepSCC: Deep Learning Based Fast Prediction Network for Screen Content Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1917-1932, Jul. 2020.

[39] S.-H. Tsang, Y.-L. Chan and W. Kuang, “Mode Skipping for HEVC Screen Content Coding via Random Forest,” *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2433-2446, Oct. 2019.

[40] G. Bjøntegaard, “Calculation of Average PSNR Differences Between RD Curves,” *VCEG*, VCEG-M33, Austin, Texas, U.S.A., pp. 1-4, Apr. 2001.

[41] “Common Test Conditions for Screen Content Coding,” *JCT-VC*, JCTVC-X1015, Geneva, Switzerland, pp. 1-8, May-Jun. 2016.

[42] JCT-VC Screen Content Sequences. [Online]. Available at: ftp://ftp.tnt.uni-hannover.de/testsequences/, accessed Jan. 2020.

[43] J. Guo, L. Zhao, T. Lin, and H. Yu, “Response to B1002 Call for Test Materials: Five Test Sequences for Screen Content Video Coding,” *JVET*, JVET-C0044, pp. 1-9, Geneva, Switzerland, May 2016.

[44] HEVC Test Model Version 16.12 Screen Content Model (SCM) Version 8.3, HM-16.12+SCM-8.3, [Online], available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.12+SCM-8.3/, accessed Jan. 2020.

[45] A. Liaw, and M. Wiener, “Classification and Regression by randomForest,” *R News*, vol. 2, no. 3, pp. 18-22, Dec. 2002.

[46] R: The R Project for Statistical Computing. [Online]. Available at: https://www.r-project.org/, accessed Jan. 2020.

[47] G. Louppe, “Understanding Random Forests: From Theory to Practice,” *Ph.D. Dissertation*, University of Liege, pp. 1-211, Jul. 2014.

[48] T. K. Ho, “The Random Subspace Method for Constructing Decision Forests,” in *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 20, no. 8, pp. 832-844, Aug. 1998.

[49] L. Breiman, “Random Forests”, *J. Mach. Learn.*, vol. 45, no. 1, pp. 5-32, Oct. 2001.