The following publication W. Kuang, Y. -L. Chan, S. -H. Tsang and W. -C. Siu, "Fast HEVC to SCC Transcoding Based on Decision Trees," 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 2018 is available at https://doi.org/10.1109/ICME.2018.8486461.

FAST HEVC TO SCC TRANSCODING BASED ON DECISION TREES

Wei Kuang, Yui-Lam Chan, Sik-Ho Tsang, and Wan-Chi Siu

Centre for Signal Processing, Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
wei.kuang@connect.polyu.hk; enylchan@polyu.edu.hk; sik-ho.tsang@polyu.edu.hk; enwcsiu@polyu.edu.hk

ABSTRACT

Screen Content Coding (SCC) is an extension of the High-Efficiency Video Coding (HEVC) for encoding screen content videos. However, there are many legacy screen content videos already encoded by HEVC. To efficiently migrate screen content videos from the existing HEVC to the emerging SCC, a machine learning based fast transcoding algorithm is proposed by using decision trees in this paper. To speed up the transcoding process, the intermediate data from both the HEVC decoder side and the SCC encoder side are jointly analyzed. Then the optimal coding unit (CU) sizes are mapped from HEVC to SCC while the mode candidates are adaptively checked according to the decision tree outcomes in the re-encoding process. Experimental results show that an average of 48.20% re-encoding time reduction is achieved with only 1.47% Bjøntegaard delta bitrate loss using All Intra (AI) configuration.

***Index Terms*—** Transcoding, High Efficiency Video Coding, Screen Content Coding, machine learning

1. INTRODUCTION

Screen content videos refer to video sequences captured from the display screen of computers or smart phones, and they are playing an essential role in many applications, such as remote desktop, online education, video conference with document sharing and WIFI display. To meet the challenges of limited bandwidth and computing resources, the Joint Collaborative Team on Video Coding (JCT-VC) launched the Screen Content Coding (SCC) extension [1] in early 2014 on top of High Efficiency Video Coding (HEVC).

Compared with the conventional camera-captured videos, screen content videos show different characteristics, and several typical examples are shown in Figure 1. It is observed that besides the camera-captured content which can be efficiently encoded by the original Intra mode in HEVC, there are also many computer-generated text and graphics. In the development of SCC, 4 major coding tools are adopted beyond HEVC to efficiently encode those new contents. They are intra block copy (IBC) mode [2, 3], palette (PLT) mode [4], adaptive motion vector resolution (AMVR) [5] and color transforms (ACT) [6]. With these coding tools, it is reported that the SCC reference software Screen Content Model version 4.0 (SCM-4.0) can provide over 50% Bjøntegaard delta bitrate (BD-Rate) [7] saving over the HEVC reference software HM-16.4 for typical screen content sequences at the expense of significant complexity increase.

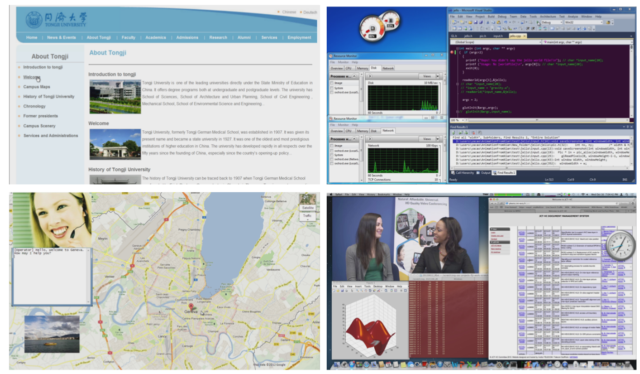

Fig 1. Screen content frame examples.

Despite the great compression efficiency gain of SCC, HEVC is still the most widely adopted video compression standard, and the full adoption of SCC for encoding screen content videos may take several years. In the long period when SCC and HEVC coexist, an efficient transcoding scheme from HEVC to SCC is in great demand. On the one hand, there are many legacy screen content videos already encoded by HEVC, and it is necessary to convert screen content bitstreams from HEVC to SCC to improve coding efficiency. On the other hand, due to the hardware limitation of user terminals, it is always difficult to implement the SCC encoding in real-time. With the rapid development of cloud-based video streaming, it is desirable to put a transcoder in a cloud sever such that the screen content videos can be uploaded to a cloud sever and let the sever convert the bitstreams from HEVC to SCC for low-cost storage. When users download screen content videos, they can either use a device with a SCC decoder or let the sever convert the bitstreams back to HEVC.

A trivial way for performing HEVC to SCC transcoding is to decode the HEVC bitstream firstly from the original video format and then re-encode it by the SCC encoder. Although this scheme can provide high coding efficiency, it is not desired in terms of computational complexity. SCC and HEVC share the same coding tree unit (CTU) hierarchical partitioning structure and intra directional modes. Therefore, it makes fast transcoding algorithms meaningful and possible by re-using the decoding information from HEVC.

In the literature, many efforts have been devoted to study the fast video transcoding algorithms [8-15]. In [10, 11], machine learning based approaches were proposed to speed up the transcoding process from MPEG-2 to H.264/AVC. In [12], a MPEG-2 to HEVC transcoder was proposed to map the CU depth from MPEG-2 to HEVC. It uses the first few frames of each sequence to train a content dependent model, and then the model maps the CU depth from MPEG-2 to HEVC. In [13, 14, 15], fast H.264 to HEVC transcoding schemes were proposed. The incoming intra modes of H.264 are mapped to larger HEVC CUs and prediction units (PUs) in [13]. In [14, 15], the decoder side information from H.264 is utilized to make fast CU size decisions in HEVC.

Although these approaches work well for their own transcoding tasks, they did not consider the special characteristics of screen content. To reduce the computational complexity of a SCC encoder, various fast algorithms have been proposed in [16, 17, 18, 19] by utilizing the SCC encoder side information only. However, with the HEVC decoder side information available, the re-encoding time of SCC can be further reduced. In the literature, there is only one work [20] which has investigated the fast HEVC to SCC transcoding process, where a fast mode decision and CU partition decision approach was proposed for YUV4:4:4 screen content videos. Although it achieves 47.93% re-encoding complexity reduction, it introduces a great BD-Rate increase of 2.14%. Considering that video sequences are usually encoded with YUV4:2:0 format by HEVC in practice, we proposed a machine learning based fast transcoding algorithm for YUV4:2:0 videos in this paper. First, the CU depth mapping from HEVC to SCC is analyzed to early terminate the CU partition process. Then the intermediate data from both the HEVC decoder and the SCC encoder are jointly analyzed to make early mode decisions by using decision trees. Experiment results show that an average of 48.20% re-encoding time reduction is achieved with only 1.47% increase in BD-Rate.

The rest of the paper is organized as follows. Section 2 briefly reviews the intra encoding decisions in SCC and the benefits brought by the HEVC to SCC transcoding. Section 3 presents the proposed fast HEVC to SCC transcoding algorithm in detail. Section 4 gives the experimental results and discussions. Finally, Section 5 concludes this paper.

2. REVIEW ON SCC INTRA ENCODING DECISIONS AND TRANSCODING BENEFITS

To address the unique characteristics of screen content videos, such as sharp edges, limited colors and many repeated patterns, IBC mode and PLT mode are the two most important coding tools in SCC. Due to these two new coding modes, significant bitrate reduction can be achieved by using HEVC to SCC transcoding.

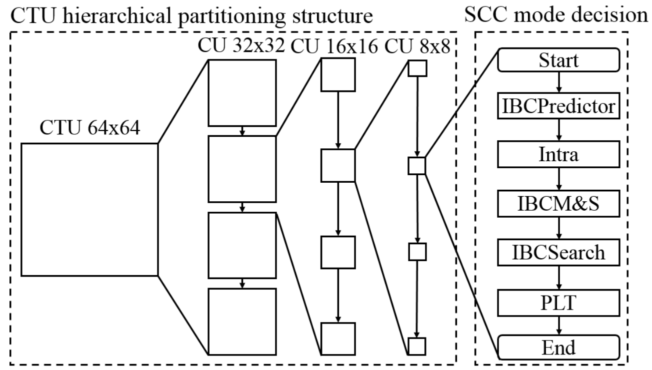

Fig 2. SCC intra encoding block diagram.

2.1 Intra encoding decisions in SCC

In addition to the CTU hierarchical partitioning structure and the Intra mode inherited from HEVC, IBC mode and PLT mode are developed in SCC to further improve the coding efficiency of the screen content. To illustrate the intra encoding process, a block diagram is shown in Figure 2.

IBC mode is a block matching based mode and it is applied for CU sizes from 64×64 (depth level of 0) down to 8×8 (depth level of 3). Unlike the conventional camera-captured videos, screen content videos have many repeated patterns within one frame. Therefore, when compressing a CU, IBC mode is designed to search in the reconstructed samples of the current frame and find the best matched block for prediction. IBC mode contains three steps, which are IBCPredictor, IBCMerge&Skip (IBCM&S) and IBCSearch. While IBCPredictor only checks several block vectors (BVs) predicted from neighbor CUs, IBCM&S and IBCSearch are the intra version of the inter Merge&Skip mode and motion estimation, respectively, in HEVC. Therefore, IBC Predictor comes with low complexity while IBCM&S and IBCSearch are relatively computationally expensive.

PLT mode compresses a CU by a palette table with an index map, and it is applied for CU sizes from 32x32 down to 8×8. Since screen content CUs contain a limited number of colors, several representative colors in a CU are selected to form a palette table, and then an index map is used to denote the position of each color.

Although SCC and HEVC share the same CTU hierarchical partitioning structure, the new IBC mode and PLT mode make CUs in inhomogeneous screen content select larger sizes. Therefore, mapping CU sizes and mode candidates accurately from HEVC to SCC is different from previous transcoding design and is the key issue for speeding up the transcoding process.

2.2 Benefits of HEVC to SCC transcoding

To understand the importance of performing HEVC to SCC transcoding for screen content videos, 13 typical screen content sequences in YUV4:2:0 format were firstly encoded by the HEVC reference software HM-16.12 with quantization parameters (QPs) at 22, 27, 32, and 37, and then the decoded

Table 1. Bitrate reduction brought by HEVC to SCC transcoding.

Sequences	No. of Frame	Frame Rate (Hz)	Bitrate Reduction (%)
ChineseEditing	0-599	60	33.88
Console	0-599	60	57.24
Desktop	0-599	60	64.99
FlyingGraphics	0-299	60	41.43
Map	0-599	60	14.53
Programming	0-599	60	42.62
SlideShow	0-499	20	60.91
WebBrowsing	0-299	30	19.57
BasketballScreen	322-621	60	39.02
MissionControlClip2	120-419	60	23.45
MissionControlClip3	0-299	60	39.37
Robot	0-299	30	5.82
ChinaSpeed	0-499	30	17.85
Average			35.44

Table 2. Ratio of optimal CUs decided by HEVC continue partitioning in SCC.

CU size in HEVC	Partitioned to 32x32 CUs (%)	Partitioned to 16x16 CUs (%)	Partitioned to 8x8 CUs (%)
64x64	11.02	0.91	0.23
32x32		4.83	0.25
16x16			3.25

HEVC videos were re-encoded by the SCC reference software HM-16.12+SCM-8.3 with the same QPs to analyze the bitrate reduction. The screen content sequences were recommended by the experts in the JCT-VC group, and the experimental conditions follow the Common Testing Conditions (CTC) under All Intra (AI) configurations [21]. The bitrate reduction for each test sequence is shown in Table 1. It is observed that the bitrate saving brought by HEVC to SCC transcoding is up to 64.99%, and 35.44% on average. Therefore, an efficient transcoding scheme is essential to improve the coding efficiency of screen content videos.

3. PROPOSED FAST HEVC-SCC TRANCODING

The coding efficiency gain brought by HEVC to SCC transcoding mainly comes from the adoption of IBC and PLT modes. However, the re-encoding process of SCC also become computationally expensive because more mode candidates need to be checked. Therefore, the information from both the HEVC decoder side and SCC encoder side are jointly utilized to adaptively check CU partitions and mode candidates of SCC.

3.1. Early CU partitioning termination

While the new modes introduced by SCC make it possible for inhomogeneous screen content selecting larger CU sizes, the optimal CUs from HEVC rarely continue partitioning in SCC. To verify this statement, the ratio of the optimal CUs decided by HEVC which continue partitioning in SCC is analyzed by encoding the testing sequences in Table 1, and the results are shown in Table 2. It is observed that the ratios of CUs which continue partitioning are relatively small, expect for 64x64 CUs in which 11.02% of them is partitioned to 32x32 CUs in SCC. Based on this analysis, early CU partitioning termination can be made by extracting the optimal CU sizes from HEVC decoder. For CUs with HEVC optimal sizes of 16x16 and 32x32, further partitions are terminated if the current CU size in SCC is equal to the optimal size decided by HEVC. For CUs with HEVC optimal size of 64x64, the SCC encoder only allows it to be partitioned to 32x32 CUs, and then further partitions are terminated.

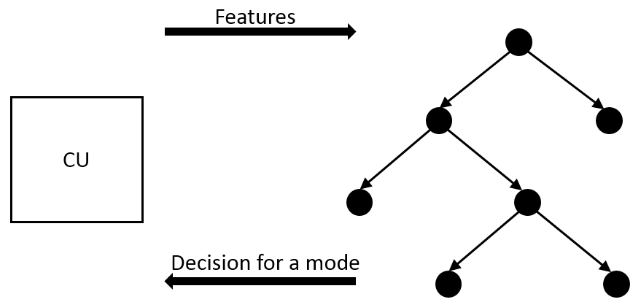

Fig 3. An example of a decision tree.

3.2. Early mode decision

To reduce the computational complexity brought by the mode decisions of SCC, several features are extracted from both the HEVC decoder and the SCC encoder to build decision trees, and then mode candidates in SCC are checked adaptively according to the decision tree outcomes.

As shown in Figure 3, a typical decision tree consists of three kinds of nodes, which are a root node, internal nodes and leaf nodes. Starting from the root node, an incoming CU goes through a series of weak classifiers stored at internal nodes, and it finally comes to one of the leaf nodes, where a decision is made. In our paper, the decision tree training process was implemented by C4.5 algorithm in the Waikato Environment for Knowledge Analysis (WEKA) [22] version 3.8. The training data for building decision trees came from 8 sequences, which are “ChineseEditing”, “Console”, “Desktop”, “FlyingGraphics”, “Map”, “SlideShow”, “MissionControlClip2” and “Robot”. For each training sequence, only 10 frames were extracted with equal time interval to generate training data. As IBCPredictor comes with low complexity while other modes have relatively high computational complexity, decision trees were trained for Intra mode, IBCM&S, IBCSearch and PLT mode, respectively, to reduce computational complexity. When training a decision tree for a mode, the positive data come from CUs encoded by this mode and the negative data come from CUs encoded by other modes. If the decision tree outcome for a mode is 1, the mode should be checked. Otherwise the mode is skipped. To avoid the data imbalance problem which is caused by more training data in one class than the other, the numbers of positive and negative training data were set to be equal during training.

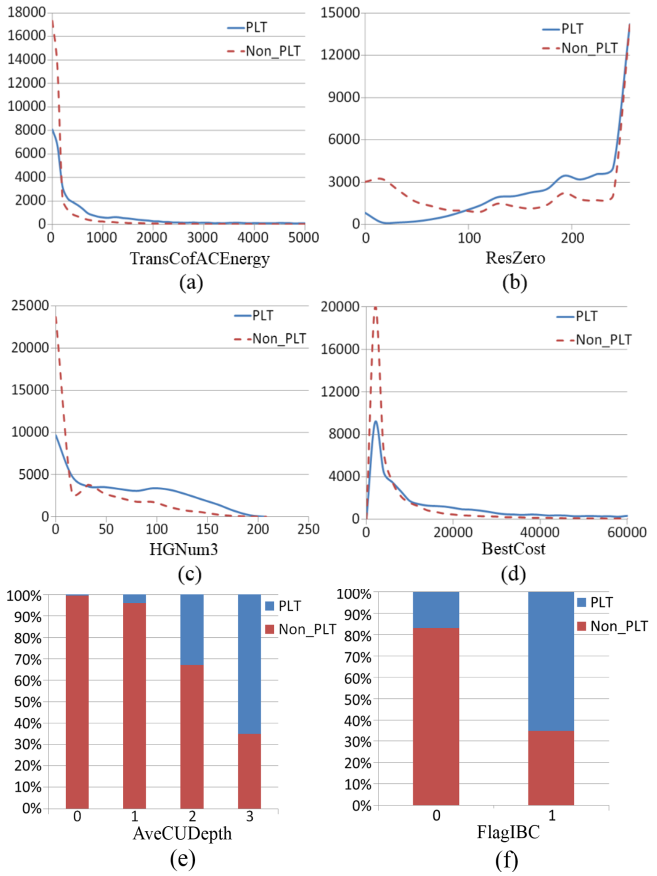

Fig 4. PLT and Non_PLT 16x16 CU distributions in terms of (a) TransCofACEnergy, (b) ResZero, (c) HGNum3, (d) BestCost, (e) AveCUDepth, and (f) FlagIBC.

When generating training data, the optimal mode of each CU was recoded with 10 features from both of the HEVC decoder and the SCC encoder.

Features from HEVC decoder:

Feature 1: The average CU depth level of HEVC AveCUDepth. It is observed that screen content tends to be encoded by small CUs in HEVC because of their inhomogeneity. Therefore, CUs with higher AveCUDepth are more likely to select IBC or PLT mode.

Feature 2: The AC coefficient energy TransCofACEnergy of the transform matrix, which is defined as the sum of square of the AC coefficients. Because screen content CUs have more high frequency components, they contain higher TransCofACEnergy than camera-captured CUs.

Feature 3: The number of zero pixels in the residual block ResZero. Screen content CUs have many uniform background pixels. Therefore, they tend to have larger values of ResZero.

Features from SCC encoder:

Feature 4-7: High gradient pixel number HGN_0, HGN_1, HGN_2, HGN_3. The high gradient pixel detection is defined as

$$|C_{Luma\_cur} - C_{Luma\_nei}| > TH_{HG}. \quad (1)$$

Table 3. Classification accuracy of each decision tree.

CU size	Intra (%)	IBCM&S (%)	IBCSearch (%)	PLT (%)
64x64	81.63	79.66		
32x32	92.15	93.66		82.82
16x16	90.53	83.77	84.33	81.76
8x8	84.34	86.34	86.36	81.24

If the luminance difference of the current pixel C_{Luma_cur} and one of its neighbor pixels C_{Luma_nei} is larger than a threshold TH_{HG}, the current pixel is detected as a high gradient pixel. Screen content CUs have many sharp edges, and they come with larger high gradient pixel number. To detect high gradient pixels with different strength, HGN_0, HGN_1, HGN_2, HGN_3 are counted with TH_{HG} at 8, 16, 32, 64, respectively.

Feature 8: Distinct color number DistColorNum, which is calculated by counting the pixels with different luminance values. Screen content CUs contain limited colors, and they have smaller DistColorNum than camera-captured CUs.

Feature 9: The rate-distortion (RD) cost of the best mode BestCost before checking the target mode. If BestCost before checking the target mode is small, the current CU may has been efficiently encoded and further modes may be unnecessary. Therefore, CUs with small BestCost tend to skip the target mode.

Feature 10: The IBC mode flag of the best mode FlagIBC before checking the target mode. If the best mode before checking the target mode is IBC mode, it is very likely that the CU is a screen content CU.

The PLT and Non_PLT (other modes) 16x16 CU distributions are shown in Figure 4 in terms of (a) TransCofACEnergy, (b) ResZero, (c) HGNum3, (d) BestCost, (e) AveCUDepth, and (f) FlagIBC, which support our statement. For example, it is observed in Figure 4(a) that CUs with higher TransCofACEnergy tend to select PLT mode because they are more likely to be screen content CUs. It should be noted that in Figure 4(b), while CUs with larger values of ResZero tend to be encoded by PLT mode, it is difficult to make classifications for CUs with ResZero of 256. The reason is that 16x16 CUs with ResZero of 256 are CUs filled with a single color. However, the classification accuracy for CUs with ResZero of 256 is not a key issue, because they can be encoded efficiently by all modes including PLT, IBC, and Intra.

As SCC supports 4 different CU sizes from 8x8 to 64x64, decision trees for all modes were trained for CUs with different sizes, respectively, and the average classification accuracy given by the 10-fold cross-validation for each decision tree is shown in Table 3. It should be noted that IBCSearch is not applied for CU sizes of 64x64, 32x32, and PLT mode is not applied for CU size of 64x64. It is observed from the table that the classification accuracies vary from 79.66% to 92.15%. To avoid the case that a CTU cannot be encoded because all modes are skipped, the original full RD mode decision is performed if all modes are skipped in the last valid depth level (the last CU depth level before the CU partition is terminated as in Section 3.1). Besides, to provide a higher classification accuracy, we set a confidence

Table 4. Performance of the proposed fast HEVC to SCC transcoding algorithm.

Sequences	Early CU partitioning termination		Early mode decision		Overall algorithm		Overall algorithm without TH_{node}	
	ΔBD-Rate (%)	ΔTime (%)	ΔBD-Rate (%)	ΔTime (%)	ΔBD-Rate (%)	ΔTime (%)	ΔBD-Rate (%)	ΔTime (%)
ChineseEditing	0.13	-6.21	0.86	-29.35	0.78	-34.27	3.41	-43.33
Console	1.82	-9.06	2.21	-32.95	4.00	-39.51	7.64	-47.71
Desktop	0.25	-5.95	1.44	-33.05	1.57	-36.83	3.57	-45.46
FlyingGraphics	0.31	-5.65	1.65	-34.06	1.85	-38.62	7.06	-49.33
Map	0.35	-14.03	0.50	-35.68	0.47	-48.07	5.01	-62.86
Programming	0.18	-14.25	0.99	-33.81	0.86	-44.75	3.89	-52.08
SlideShow	0.30	-46.65	1.04	-30.72	0.82	-67.93	4.97	-72.64
WebBrowsing	0.35	-11.79	1.73	-35.67	1.91	-44.89	4.38	-52.57
BasketballScreen	0.15	-16.11	1.10	-36.09	0.95	-48.95	4.63	-58.47
MissionControlClip2	0.37	-23.11	1.63	-33.21	1.19	-52.46	4.50	-59.06
MissionControlClip3	0.83	-13.26	1.53	-33.68	1.79	-44.72	4.43	-53.05
Robot	2.55	-34.82	0.11	-36.55	2.10	-67.78	2.24	-78.83
ChinaSpeed	0.23	-24.12	0.67	-34.45	0.76	-57.87	2.50	-66.05
Average (ALL)	0.60	-17.31	1.19	-33.79	1.47	-48.20	4.48	-57.03

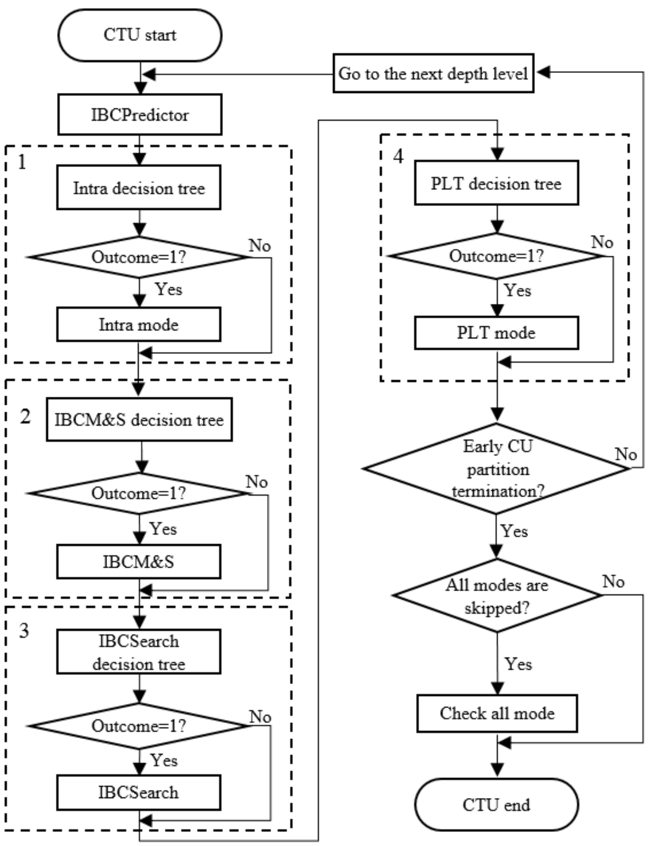

Fig 5. Flowchart of the proposed fast HEVC to SCC transcoding.

threshold TH_{node} when a CU reaches its last valid depth level. If the accuracy of a classification made by the decision tree leaf node is smaller than TH_{node}, which is set to 85% empirically, the target mode will be checked regardless of the leaf node outcome.

As a summary, the flowchart of the proposed fast HEVC to SCC transcoding algorithm is shown in Figure 5, where 1-4 denote the mode decision models for Intra mode, IBCM&S, IBCSearch, and PLT mode, respectively.

4. EXPRIMENTAL RESULTS

To evaluate the performance of the proposed fast HEVC to SCC transcoding algorithm, 13 testing sequences in YUV4:2:0 format were encoded by the HEVC reference software HM-16.12 with quantization parameters (QPs) at 22, 27, 32, and 37, and then the HEVC bitstreams were decoded into YUV videos with decoder side information recoded. Finally, the HEVC decoded videos were re-encoded by the SCC reference software HM-16.12+SCM-8.3 with the same QPs using our proposed algorithm. The test platform used for simulations was a HP EliteDesk 800 G1 computer with a 64-bit Microsoft Windows 10 OS running on an Intel Core i7-4790 CPU of 3.6 GHz and 32.0 GB RAM. The encoding time and BD-Rate of the proposed algorithm is compared with the straightforward transcoding approach, which directly re-encodes the decoded HEVC video without any decoder side information.

Table 4 shows the performance of the proposed early CU partitioning termination technique, early mode decision technique and overall algorithm. To investigate the impact of the confidence threshold TH_{node} at the last depth level, the performance of the overall algorithm without TH_{node} is also shown in Table 4. It is observed that the proposed overall algorithm achieves up to 67.93% re-encoding time reduction in the transcoding process. On average, it provides 48.20% re-encoding time reduction with only 1.47% increase in BD-Rate. More specifically, the early CU partitioning termination technique provides 17.31% re-encoding time reduction with 0.60% increase in BD-Rate, while the early mode decision technique achieves 33.79% re-encoding time reduction with 1.19 % increase in BD-Rate. From the table we can see that by adopting the confidence threshold TH_{node}, the increase in BD-Rate is significantly reduced from 4.48% to 1.47% while the re-coding time reduction is slightly decreased from 57.03% to 48.20%, which proves that TH_{node} is essential to reduce the coding efficiency loss brought by the proposed algorithm.

5. COUCLUSION

In this paper, a fast HEVC to SCC transcoding algorithm is proposed by analyzing the intermediate data from both the HEVC decoder side and the SCC encoder side. To avoid the exhaustive CU partition and mode decision process, optimal CU sizes are mapped from HEVC to SCC while decision trees are built for making early mode decisions. The performance of the proposed algorithm is validated by implementing it in HM-16.12+SCM-8.3. Experimental results show that the proposed algorithm achieves 48.20% re-encoding time reduction with only 1.47% BD-Rate increase on average.

6. ACKNOWLEDGEMENTS

This work was supported by the Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University (PolyU), and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant No. PolyU 152530/16E).

7. REFERENCES

[1] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50–62, 2016.

[2] X. Xu et al., "Intra block copy in HEVC screen content coding extensions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, 2016.

[3] S.-H. Tsang, W. Kuang, Y.-L. Chan, and W.-C. Siu "Decoder side merge mode and AMVP in HEVC screen content coding," in *ICIP*, 2017, pp.260-264.

[4] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Exploiting inter-layer correlations in scalable HEVC for the support of screen content videos," in *Digital Signal Processing (DSP)*, 2016, pp.1–5.

[5] B. Li, J. Xu, G. J. Sullivan, Y. Zhou, and B. Lin, "Adaptive motion vector resolution for screen content," in *Joint Collaborative Team on Video Coding(JCTVC)*, *JCTVC-S0085*, 2014.

[6] L. Zhang, J. Chen, J. Sole, and M. Karczewicz, "AHG8: In-loop color-space transform," in *Joint Collaborative Team on Video Coding(JCTVC)*, *JCTVC-Q0112*, 2014.

[7] G. Bjontegaard, "Calculation of average PSNR differences between rd-curves," *VCEG-M33*, 2001.

[8] K.-T. Lai, Y.-L. Chan, and W.-C. Siu, "New architecture for dynamic frame-skipping transcoder," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 886–900, 2002.

[9] Y.-L. Chan, H.-K Cheung and W.-C. Siu, "Compressed-domain techniques for error-resilient video transcoding using RPS," *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 357–370, 2009.

[10] G. Fernandez-Escribano, H. Kalva, J. Martinez, P. Cuenca, L. Orozco-Barbosa, and A. Garrido, "An MPEG-2 to H.264 video transcoder in the baseline profile," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 5, pp. 763-768, 2010.

[11] C. Holder, T. Pin, and H. Kalva, "Improved machine learning techniques for low complexity MPEG-2 to H.264 transcoding using optimized codecs," in *International Conference on Consumer Electronics*, 2009, pp. 1-2.

[12] T. Shanableh, E. Peixoto, and E. Izquierdo, "MPEG-2 to HEVC video transcoding with content-based modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1191-1196, 2013.

[13] D. Zhang, B. Li, J. Xu, and H. Li, "Fast transcoding from H.264 AVC to high efficiency video coding," in *ICME*, 2012, pp. 651-656.

[14] F. Zhang, Z. Shi, X, Zhang, and Z, Guo, "Fast H.264/AVC to HEVC transcoding based on residual homogeneity," in *International Conference on Audio, Language and Image Processing*, 2014, pp. 765 – 770.

[15] H. Yuan, C. Guo, J. Liu, X, Wang, and S. Kwong, "Motion-homogeneous based fast transcoding method from H.264/AVC to HEVC," *IEEE Transactions on Multimedia*, vol. 19, no 7, PP. 1416 – 1430, 2017.

[16] 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," in *ICASSP*, 2015, pp.1409–1413.

[17] 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," in *APSIPA ASC*, 2015, pp. 396–400.

[18] 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," in *APSIPA ASC*, 2016, pp.1–5.

[19] W. Kuang, S.-H. Tsang, Y.-L. Chan and W.-C. Siu, "Fast mode decision algorithm for HEVC screen content intra coding," in *ICIP*, 2017, pp. 2473 – 2477.

[20] F. Duanmu, Z. Ma, W. Wang, M. Xu, and Y. Wang, "A novel screen content fast transcoding framework based on statistical study and machine learning," in *ICIP*, 2016, pp. 4205 – 4209.

[21] H.-P. Yu, R. Cohen, K. Rapaka, and J. -Z Xu, "Common test conditions for screen content coding," in *Joint Collaborative Team on Video Coding(JCTVC)*, *JCTVC-X1015-r1*, 2016.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.