An HEVC-Compliant Fast Screen Content Transcoding Framework Based on Mode Mapping

Fanyi Duanmu, Zhan Ma, Meng Xu, and Yao Wang, *Fellow, IEEE*

***Abstract*—** This paper presents a novel fast transcoding framework to efficiently bridge the state-of-art High Efficiency Video Coding (HEVC) standard and its Screen Content Coding (SCC) extension to support the bitstream compatibility over the legacy HEVC devices. By exploiting the side information from the SCC bitstream, fast mode and partition decisions are made to accurately translate the novel SCC modes to conventional HEVC modes based on statistical mode mapping techniques. Compared with the "Full-Decoding-Full-Encoding" (FDFE) solution, the proposed framework achieves on average 51% and 82% complexity reductions with 0.57% Bjøntegaard-Delta Rate (BD-Rate) loss and 9.74% BD-Rate gain under All-Intra (AI) and Low-Delay (LD) configurations, respectively. Compared with the direct transcoding reusing Intra mode and Inter motion, the proposed mode mapping framework introduces additional 23% and 6% complexity reductions for AI and LD encoding configurations with 0.43% BD-Rate loss and 1.10% BD-Rate saving, respectively. The proposed solution is extended to support the Single-Input-Multiple-Output (SIMO) screen content adaptive streaming at the edge clouds, where an SCC bitstream coded in high quality is transcoded into multiple HEVC bitstreams in reduced qualities. Our proposed solution achieves on average 49% and 76% complexity reductions with 0.78% BD-Rate loss and 7.40% BD-Rate gain under AI and LD configurations, respectively.

***Index Terms*—** High Efficiency Video Coding (HEVC), Screen Content Coding (SCC), Video Transcoding, Fast Mode Decision, Mode Mapping.

I. INTRODUCTION

A. Motivation

SCREEN content (SC) videos have become popular in recent years with the development and advances in mobile technologies and cloud applications, such as shared screen collaboration, remote desktop interfacing, cloud gaming, wireless display, animation streaming, online education, etc. These emerging applications create an urgent demand for better compression technologies and low-latency delivery solutions for screen content videos.

To exploit the unique signal characteristics of screen content and develop efficient SC compression solutions, the ISO/IEC Moving Picture Expert Group and the ITU-T Video Coding Experts Group, also referred as "Joint Collaborative Team on Video Coding" (JCTVC), has launched the standardization of SCC extension [1] on top of the latest video standard - High Efficiency Video Coding (HEVC) [2] since January 2014 and this extension is concluded in 2016 with significant research efforts involved from both academia and industry.

The official JCTVC Screen Content Model software (SCM) [3] is reported to provide >50% BD-Rate saving over the HEVC Range Extension (RExt) [1] for computer-generated contents. Four major coding tools were introduced and adopted during the standardization, known as "Intra Block Copy" (IBC) [4] [5], "Palette Coding Mode" (PLT) [6], "Adaptive Color Transform" (ACT) [7] and "Adaptive Motion Compensation Precision" (AMCP) [8] [9], respectively.

Recognizing the market demands and SCC efficiency, industrial companies are currently following this new extension and mostly likely may include these new coding techniques into their future products. From the consumers' perspective, software-based solutions are desired to accommodate the new bitstream encoded with HEVC-SCC. Therefore, it is critically important to develop efficient algorithms to bridge the existing HEVC and its incoming HEVC-SCC extension using video transcoding (VTC) techniques, especially during the phase when HEVC and the novel HEVC-SCC bitstreams coexist.

VTC is a useful and mature technology to realize video adaptation. It converts the incoming bitstream from one version to another. During the conversion, many properties from the source video may change, such as video format, video bitrate, frame rate, spatial resolution and coding standards used. In the literature, the conversion within the same standard (e.g., the spatial re-scaling in H.264/AVC) is referred as "homogeneous transcoding", while the conversion between different standards (e.g., between H.264/AVC and HEVC) is referred as the "heterogeneous transcoding". Beyond that, even additional information could be inserted during transcoding, such as watermarking, error resilience, etc. In practice, a transcoding server can be used to periodically examine the client's constraints (e.g., bandwidth, power limit, display resolution, etc.) and tailor the suitable bitstreams accordingly.

Even though it is possible to use the "trivial" approach, which first decodes the source bitstream and then completely re-encodes into the target bitstream, however, such approach proves inefficient from the complexity point of view. A reasonable solution should utilize the decoded side information from the source bitstream to facilitate the re-encoding such that

Manuscript received on Feb 11, 2018; revised on Aug 8 and Sep 23, 2018, accepted on Sep 28, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61571215 and in part by the Fundamental Research Funds for the Central Universities under Grant 021014380053, and Grant 021014380091. (Corresponding author: Zhan Ma)

F. Duanmu and *Y. Wang* are with Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA. (e-mail: fanyi.duanmu@nyu.edu; yaowang@nyu.edu).

Z. Ma is currently with Nanjing University, 163 Xianlin Ave., Nanjing 210093, China. (e-mail: mazhan@nju.edu.cn)

M. Xu is currently with Tencent America, 661 Bryant St., Palo Alto, CA 94301, USA. (e-mail: mengxxu@tencent.com)

both the coding efficiency is preserved and the re-encoding speed is significantly improved.

B. Previous Work

There are a great amount of prior works for HEVC encoder accelerations, which are quite related to this work. They are summarized into the following categories.

Category 1: Mode Reduction. A gradient-based fast mode decision framework was proposed in [11], which bases on CU directional histogram analysis to reduce the number of intra candidates before mode selection. A reported 20% complexity reduction over HM-4.0 is achieved under this scheme with negligible coding performance loss for Intra-frame coding. Another fast intra mode decision algorithm was proposed in [12], which exploits the directional information of neighboring blocks to reduce the Intra candidates of the current CU. Up to 28% complexity reduction is reported over HM-1.0 with insignificant coding performance loss for Intra-frame coding. The HM test model software adopted [13] to reduce Intra-frame coding candidates. Firstly, a rough mode decision (RMD) is performed using Hadamard cost to choose fewer candidates out of 35. Then the extra most probable modes (MPMs) derived from spatial neighbors will be added to the previous candidate set if they are not yet included.

Category 2: Cost Replacement. An entropy-based fast Coding Tree Unit partition algorithm was proposed in [14], which replaces heavy Rate-Distortion optimization (RDO) calculation by Shannon entropy calculation. A 60% complexity reduction is reported using this algorithm with a BD-rate loss of 3.8% for Intra-frame coding. In [13], Hadamard cost is used for Intra RMD without fully formulating the RD cost. This approach significantly reduces the intra coding complexity.

Category 3: Fast Partition Termination. A fast CU splitting decision scheme was proposed in [15], using weighted SVM decision for early CU partition termination for both Intra-frame and Inter-frame coding. A complexity reduction of over 40% is reported over HM-6.0. Another fast termination algorithm was proposed in [16], using texture complexity of neighboring blocks to eliminate unnecessary partition of the current CU. A 23% encoder speed-up on average is reported over HM-9.0 for Intra-frame coding. Another work by Zhang and Ma [17] includes a set of early termination criteria for HEVC intra coding based on experimental observation and simulation results. To determine the splitting decision, encoder will do a 1-level RD evaluation by comparing current CU Hadamard cost with the combined Hadamard cost of 4 sub-CUs without further splitting. Zhang and Ma further proposed an improved 3-step fast HEVC Intra coding algorithm in [18]. At the RMD step, a 2:1 down-sampled Hadamard transform is used to approximate the encoding cost followed by a progressive mode refinement and early termination verification. It reports on average 38% complexity reduction over HM-6.0 with 2.9% BD-rate loss for Intra-frame coding.

Category 4: Fast Search Algorithm. A number of fast motion estimation (ME) algorithms have been proposed in the past years, including multi-step search [19] [20], diamond search [21], cross-diamond search [22], hexagon search [23], etc. These algorithms follow different search patterns to reduce the number of search points for inter-frame coding. In HEVC Test Model software (HM), Enhanced Predictive Zonal Search (EPZS) [24] is incorporated to reduce encoder complexity, in which prediction is continuously refined within local search using a small diamond or square pattern and the updated best vector becomes the new search center.

These prior works were mainly proposed for natural video coding without considering the unique signal properties of screen contents, which typically contain limited distinct colors, sharper edges, repetitive graphical patterns, less complicated textures and irregular motion fields. Besides, these works did not take into account the newly-introduced coding modes (e.g., IBC or PLT). Therefore, the conventional fast algorithms cannot be directly applied onto SCC.

There are a few recent works proposed specifically for SCC fast encoding. Li and Xu presented a fast algorithm for AMCP [25], to quickly determine the frame type (namely, SC image or Natural image), based on the percentage analysis of “smooth blocks”, “collocated blocks”, “matched blocks” and “other blocks” (i.e., the blocks that do not belong to the previous three categories). Kwon and Budagavi proposed a fast IBC search algorithm [26], by imposing restrictions on IBC search range, search directions and motion compensation precision. There are also several works on hash-based fast search algorithms for IBC mode and Inter mode coding [27] [28] [29]. Tsang, Chan and Siu proposed a Simple Intra Prediction (SIP) scheme [30] to bypass Rough Mode Decision (RMD) and RDO processing for the smooth SC regions, whose CU boundary samples are exactly the same. Lee et al. proposed a fast Transform Skip Mode Decision framework for SCC [31], by enforcing IBC block with zero coded block flag (CBF) to be encoded with transform skip mode. Zhang, Guo and Bai proposed a Fast Intra Partition Algorithm [32] for SCC, using the CU entropy and the CU coding bits to determine the CU partition for Intra-frame coding. In a recent work [33] proposed by Zhang and Ma, temporal CU depth correlations are exploited to determine the CU partition. In our previous work [10] and [34], we propose to use supervised machine learning (ML) techniques to make fast CU partition and mode decisions based on the CU low-level statistical features.

Beyond these above-mentioned fast encoding solutions, VTC fast algorithms benefit greatly from re-utilizing the decoded side information, including block partitions, coding modes, residuals, transform coefficients, etc. In [35], a video transcoding overview is presented from a system perspective, with spatial and temporal resolution reduction, DCT-domain down-conversion introduced. When HEVC is introduced in 2013, a huge amount of VTC studies were redirected into “H.264/AVC - HEVC” conversion. For instance, Peixoto, et al. proposed several machine learning and statistics based schemes (e.g., [36] [37] [38]) to improve HEVC re-encoding speed. In their papers, H.264/AVC Macroblocks (MBs) are mapped into HEVC coding units based on the distribution of motion vectors (MVs) through online or offline training. Incorporated with statistics-based fast termination criteria, the proposed schemes could introduce a >3x encoder speedup with a 4% BD-Rate loss compared with the trivial transcoder. Diaz-Honrubia, et al. also proposed a series of fast VTC schemes (e.g., [39] [40]) to exploit H.264/AVC decoded side information for HEVC CU partition decision based on a Naïve-Bayes (NB) classifier, specifically for CUs with size 32x32 and 64x64, whereas for the smaller CUs, the proposed transcoder simply mimics the H.264/AVC coding behaviors. A quantitative speed-up of 2.5x is reported with 5% BD-Rate penalty. In [41], a HEVC fast transcoder is proposed based on block homogeneity prediction. Residuals and MV consistencies are utilized to represent the homogeneity of target region and decide the CU partition. Another similar work [42] proposed by Zheng uses mean absolute deviations (MAD) of residual and sum of absolute residual (SAR) as the homogeneity indicator to early terminate CU partition. A 57% complexity reduction is achieved with 2.2% BD-Rate loss. In [43], a mode merging and mapping solution is presented using H.264/AVC block motion vector (MV) variance and mode conditional probabilities to predict HEVC merge decisions. A 50% complexity reduction is reported with negligible BD-Rate loss.

C. *Our Contributions*

Though there have been substantial prior research efforts in video coding and video transcoding acceleration, to our best knowledge, we are the first research group addressing screen content transcoding. Our previous paper [14] is the first work for accelerating SC transcoding, focusing on the HEVC-SCC forward transcoding for bandwidth reduction consideration. In this work, we focus on the fast transcoding from SCC bitstream into HEVC bitstream for the backward compatibility over the legacy HEVC devices, as illustrated in Figure 1.

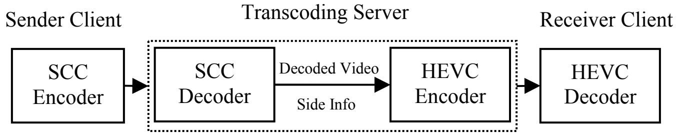


Figure 1. SCC-HEVC Transcoding Framework

Our contributions in this paper are three folds:

Firstly, we conduct extensive statistical studies to analyze the behaviors of HEVC and SCC encoders over different screen contents. Such information enables us to better understand the relationships between screen content characteristics and the codec behaviors (e.g., mode preferences).

Secondly, we propose an ultra-fast SCC-HEVC transcoding solution based on statistical mode mapping techniques. This is the first work addressing both Intra-frame and Inter-frame SCC-HEVC transcoding. The experimental results demonstrate that the proposed solution achieves a significant transcoding complexity reduction while preserving the coding efficiency.

Finally, we further generalize the proposed framework to support Single-Input-Multiple-Output (SIMO) transcoding for practical applications over the cloud.

From the hardware perspective, the industry has moved forward extensively with HEVC-compatible chip deployment. Though HEVC-SCC provides the state-of-the-art compression efficiency with remarkable bitrate reduction beyond HEVC at similar visual quality, the mainstream devices (e.g., smartphone, tablet, etc.) are only equipped with hardware-accelerated HEVC decoder, rather than HEVC-SCC decoder. Therefore, it is extremely useful and demanding to provide SCC to HEVC transcoding solutions, to potentially support millions of users whose devices are SCC-incompatible but HEVC-compatible.

From the application perspective, the basic “Full Decoding Full Encoding” (FDFE) solution might be sufficient for some non-realtime applications, e.g., video streaming. However, in recent years, we have witnessed the explosive growth of low-delay and real-time screen content applications, such as cloud gaming, multi-party screen sharing, remote desktop interfacing, etc. These applications have more stringent latency requirements and need adaptive bitstream support for diverse network conditions and diverse device constraints. Compared with the FDFE solution, our proposed framework demonstrates the following advantages:

Firstly, the proposed framework can significantly reduce the transcoding complexity and therefore reduces the end-to-end (E2E) processing delay, which is important for the low-latency applications, e.g., cloud gaming, remote desktop interfacing.

Secondly, the proposed framework fully utilizes the decoded bitstream side information to make fast mode, partition and motion search decisions, leading to a significant power saving, which is critical for battery-powered mobile applications and software-based transcoding applications.

Finally and importantly, the proposed solution significantly reduces enterprise maintenance costs for transcoding devices (e.g., streaming servers). The proposed solution achieves a 5x speedup, potentially leading to 80% server reductions to fulfill the same workload requirement.

The sequel of the paper is structured as follows. Section II briefly reviews SCM coding structure, SCC new coding tools and discusses about the major technical challenges. Section III provides the statistical studies and behavior analyses of SCC and HEVC encoders over typical screen contents. Section IV presents our proposed SCC-HEVC transcoding algorithms. In Section V, a Single-Input-Multiple-Output (SIMO) transcoding framework is presented and discussed. In Section VI, the experimental results and analyses are presented. This paper concludes in Section VII with some future work summarized.

II. HEVC Screen Content Model (SCM): A Brief Review

SCM is the JCTVC official test model software for SCC extension development. This software is developed upon HEVC-RExt codebase and supports YUV4:4:4, YUV4:2:0 and RGB4:4:4 sampling formats. Beyond HEVC, new SCC coding tools (e.g.: IBC, PLT, ACT, etc.) are introduced to improve the coding efficiency. Within the scope of this paper, we are working on SCM-4.0 software and our fast algorithms can be easily migrated into other SCM releases.

A. *SCM Mode and Partition Decision*

SCM inherits the same flexible quadtree block partitioning scheme from HEVC, which enables the flexible combinations of CUs, Prediction Units (PUs) and Transform Units (TUs) to adapt to diverse picture contents. CU is the square basic unit for mode decision. The Coding Tree Unit (CTU) is the largest CU, of 64x64 pixels by default.

At encoder, pictures are divided into non-overlapping CTUs and each CTU can be recursively divided into four equal-sized smaller CUs, until the maximum hierarchical depth is reached, as shown in Figure 2. At each CU-level, to determine the optimal encoding parameters (e.g.: partition, mode, etc.), an exhaustive search is currently employed by comparing RD costs among different coding modes at the current level and recursively comparing the minimum RD cost at the current CU depth against the sum of RD costs of its sub-CUs (each using best mode and partition). For the rest of this paper, we will use "CU64" (i.e., CTU), "CU32", "CU16" and "CU8" to denote CUs at different depths.

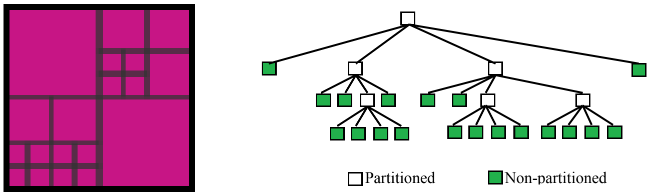


Figure 2. CU Hierachitical Quadtree Partitioning Structure of SCM

B. SCM New Coding Tools beyond HEVC

Beyond HEVC, four major encoding tools are integrated into SCM to compress SC more efficiently.

Intra Block Copy (IBC) [4] [5] is an Intra-frame version of motion estimation and compensation scheme. To compress the current PU, the encoder will search over the previously coded areas (either in restricted area or globally) in the same frame and find the best matching block. If chosen, a "Block Vector" (BV) will be signaled, either explicitly or implicitly, to indicate the relative spatial offset between the best matching block and the current PU location.

Palette Mode [6] encodes the current CU as a combination of a color table and the corresponding index map. Color table stores representative color "triplets" of RGB or YUV. Then the original pixel block is translated into a corresponding index map indicating which color entry in the color table is used for each pixel location.

Adaptive Color Transform [7] converts residual signal from original RGB or YUV to YCoCg color space. It decorrelates the color components, reduces the residual signal energy and therefore improves the coding efficiency.

Adaptive Motion Compensation Precision [8] [9] analyzes Inter-frame characteristics and categorizes the current frame into either a "natural video frame" (NVF) or a "screen content frame" (SCF). For SCF, integer-pixel precision is applied for motion estimation. For NVF, sub-pixel precision is applied.

C. HEVC-SCC Fast Transcoding Challenges

Different from the conventional HEVC Intra-modes, SCC modes are highly dependent on the repetitive graphical patterns and image colors that previously appeared. This "historical dependency" makes fast partition decision and VTC mode mapping much more complicated and challenging. For example, in IBC blocks, depending on whether similar pattern appeared previously, encoding costs of the CUs with the same or similar pattern but at different locations may vary significantly. Similarly, for PLT coding mode, depending on whether similar colors appeared before and how frequent these colors are, the PLT coding costs of the CUs with the same or similar pattern but at different locations may vary significantly. Furthermore, the newly-introduced SCC modes and coding options enable "inhomogeneous" blocks to be encoded as a larger block without splitting. As shown in Figure 3, 16x16 textual CUs in the top row are encoded using PLT mode (in green) directly without splitting into smaller 8x8 Intra CUs (in red) in the bottom row. To conclude, due to the unique signal characteristics of screen contents and the designs of PLT and IBC algorithms, existing VTC mode mapping and fast splitting termination algorithms cannot be applied to SC transcoding directly. How to accurately map SCC modes to HEVC modes and efficiently determine the HEVC partition is a challenging problem, even for human judgment.

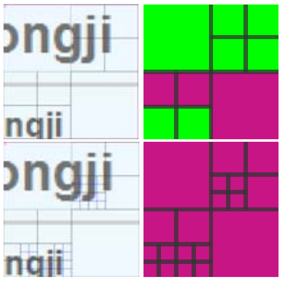


Figure 3. CTU Partition Decision Comparison between SCM and HEVC (Top: Text CTU coded by SCM-4.0; Bottom: Text CTU coded by HEVC)

III. SCC and HEVC Coding Statistical Study

In this section, we conduct statistical studies on SC mode and partition distribution, using HEVC and SCC encoders.

A. Dataset Preparation

Our statistical studies are based on HM-16.4 and SCM-4.0 encoding data using the standard sequences selected by the experts from JCTVC. These SC sequences cover the most typical screen contents, such as "Desktop", "Console", "Map", "SlideShow", "WebBrowsing", etc., as shown in Figure 4.

For Intra-frame coding statistical study, we reused the same sample frames as in our previous work [10]. For inter-frame coding statistical study, the first 10 frames from each sequence are encoded using Low-Delay with P-frame (LDP) coding structure. Only data from the Inter-coded frames are collected. We assume the mode distribution can be generalized to the new SC videos. Since that the mode and partition decisions are quantization dependent, simulation data for QP=22 and QP=37 are provided for comparison.

Figure 4. Sample Frames from SCC Standard Sequences

B. Intra-frame Mode and Partition Distribution

The Intra-frame partition distribution using HM-16.4 and SCM-4.0 are provided in Table I. The Intra-frame mode distribution statistics using SCM-4.0 is provided in Table II. Besides, the Intra-frame sub-mode selection distribution using HM-16.4 is provided in Table III.

TABLE I
INTRA-FRAME PARTITION STATISTICS

CU Width	QP	Partition % HM-16.4	Non-Partition % HM-16.4	Partition % SCM-4.0	Non-Partition % SCM-4.0
64	22	91.27%	8.73%	90.37%	9.63%
64	37	86.14%	13.86%	84.80%	15.20%
32	22	79.88%	20.12%	82.75%	17.25%
32	37	73.24%	26.76%	78.50%	21.50%
16	22	69.77%	30.23%	49.06%	50.94%
16	37	65.14%	34.86%	44.84%	55.06%

TABLE II
SCM-4.0 INTRA-FRAME MODE STATISTICS

CU Width	QP	IBC Merge	IBC Skip	IBC Inter	Intra	PLT
64	22	0.46%	46.01%	Disabled	53.53%	Disabled
64	37	0.43%	29.44%	Disabled	70.13%	Disabled
32	22	0.49%	37.49%	4.29%	29.01%	28.74%
32	37	0.91%	29.70%	3.01%	33.31%	33.06%
16	22	1.39%	34.75%	10.52%	26.74%	26.61%
16	37	1.40%	34.37%	10.70%	27.04%	26.49%
8	22	5.41%	31.60%	18.03%	22.63%	22.33%
8	37	3.96%	32.49%	19.88%	22.02%	21.65%

TABLE III
HM-16.4 INTRA SUB-MODE STATISTICS

CU Width	QP	Intra(0) Planar	Intra(1) DC	Intra(10) Horizontal	Intra (26) Vertical	Intra Others
64	22	7.79%	7.04%	27.39%	47.99%	9.80%
64	37	14.40%	9.02%	27.85%	32.59%	16.14%
32	22	5.83%	10.17%	45.86%	25.22%	12.92%
32	37	9.49%	9.81%	43.38%	21.74%	15.59%
16	22	8.37%	5.92%	43.52%	21.30%	20.89%
16	37	10.71%	6.37%	37.91%	21.07%	23.95%
8	22	8.05%	5.39%	30.73%	25.61%	30.23%
8	37	10.26%	6.20%	26.08%	23.07%	34.39%

C. Inter-frame Mode and Partition Distribution

The Inter-frame partition distribution using HM-16.4 and SCM-4.0 are provided in Table IV. The Inter-frame mode distribution using SCM-4.0 and HM-16.4 are provided in Table V and Table VI. Similar to our previous work [10], we also analyze the encoder Inter-frame coding complexity distribution using CPU tick counters to document the CPU clock cycles consumed by the target coding mode. Though the complexity profiling results may differ from platform to platform, we assume the percentage of each mode will not vary significantly and should reflect the encoder complexity distribution. The profiling is conducted over each CU size using different coding modes during the mode selection process. The distribution is summarized in Table VII.

TABLE IV
INTER-FRAME PARTITION STATISTICS

CU Width	QP	Partition % HM-16.4	Non-Partition % HM-16.4	Partition % SCM-4.0	Non-Partition % SCM-4.0
64	22	19.06%	80.94%	19.09%	80.91%
64	37	13.41%	86.59%	14.99%	85.01%
32	22	47.10%	52.90%	47.82%	52.18%
32	37	44.95%	55.05%	44.49%	55.51%
16	22	51.38%	48.62%	33.76%	66.24%
16	37	49.18%	50.82%	31.74%	68.26%

TABLE V
SCM-4.0 INTER-FRAME MODE STATISTICS

CU Width	QP	Intra	PLT	IBC	Merge/Skip	Inter
64	22	3.58%	0.00%	0.84%	94.29%	1.29%
64	37	5.56%	0.00%	0.69%	91.67%	2.09%
32	22	10.40%	2.03%	8.82%	65.26%	13.48%
32	37	6.27%	0.90%	8.48%	69.46%	14.89%
16	22	8.43%	1.45%	13.62%	63.48%	13.03%
16	37	2.36%	0.39%	9.61%	69.39%	18.25%
8	22	3.02%	1.70%	16.12%	58.42%	20.75%
8	37	0.89%	0.13%	12.55%	66.76%	19.68%

TABLE VI
HM-16.4 INTER-FRAME MODE STATISTICS

CU Width	QP	Intra	Skip	Merge	Inter
64	22	1.74%	96.20%	1.67%	0.39%
64	37	2.07%	95.76%	1.46%	0.71%
32	22	5.83%	80.27%	8.85%	5.05%
32	37	4.73%	86.55%	4.96%	3.76%
16	22	9.19%	76.18%	8.65%	5.98%
16	37	6.74%	84.65%	3.82%	4.79%
8	22	8.69%	64.21%	18.53%	8.56%
8	37	5.32%	72.25%	14.92%	7.49%

TABLE VII
SCM-4.0 MODE COMPLEXITY STATISTICS IN PERCENTAGE

CU Width	Intra	IBC	PLT	Inter Hash	Inter Merge/Skip	Inter	Total
64	2.80%	0.37%	0.16%	0.58%	6.12%	12.53%	22.56%
32	2.16%	0.40%	0.40%	0.16%	4.02%	14.56%	21.70%
16	1.42%	1.89%	0.29%	0.84%	5.99%	16.87%	27.30%
8	1.77%	3.51%	0.25%	2.88%	2.99%	16.84%	28.24%

D. Statistical Study Discussions

The following conclusions can be drawn:

1. From Table I, the total non-partitioned block percentage of SCM-4.0 is greater than the percentage using HM-16.4. It coincides with the reasoning as illustrated in Figure 3, that the inhomogeneous SC block can be more efficiently coded using IBC or PLT without further splitting. Larger CUs are more likely to be partitioned. Till CU16, the partition decision using SCC becomes almost unpredictable.

2. From Table II, SCM-4.0 uses a large proportion of SCC modes to compress screen contents, particularly in smaller CU sizes, since smaller CUs can find perfect or good matching blocks more easily using the IBC modes. It is therefore desirable to design fast algorithms to accelerate the SC blocks transcoding, particularly for the IBC-coded blocks.

3. Table III implies the directionality distribution of typical screen contents. Different from natural videos, SC videos are more dominated by the purely horizontal and vertical graphical patterns. The four major Intra sub-modes, i.e., Intra-Planar, Intra-DC, Intra-Horizontal and Intra-Vertical consume a large percentage of Intra mode usage (i.e., from >65% up to >90%).

4. For Inter-frame coding, as shown in Table IV, V and VI, *Merge* and *Skip* modes cover a large proportion. Since the computer-generated contents are mostly noise-free, therefore, the stationary areas within SC videos are more likely to find perfect temporal matches than natural videos and therefore coded in *Skip* mode.

5. As shown in Table VII, the major Inter-frame complexity is consumed by Inter modes (i.e., approximately 80%). This coincides with HM and SCM codec implementation that the Intra-frame modes can be fast terminated when temporarily a perfect matching block is found for the current CU. The major complexity is consumed during the "motion estimation" stage, in which the motion vectors (MV) are refined progressively until a convergence at sub-pixel precision.

IV. ULTRA-FAST SCC-HEVC TRANSCODING

In this section, we present our HEVC-SCC fast transcoding framework, as shown in Figure 5. The proposed fast algorithm is designed based on mode mapping technique, in which SCC modes are efficiently mapped into HEVC coding modes.

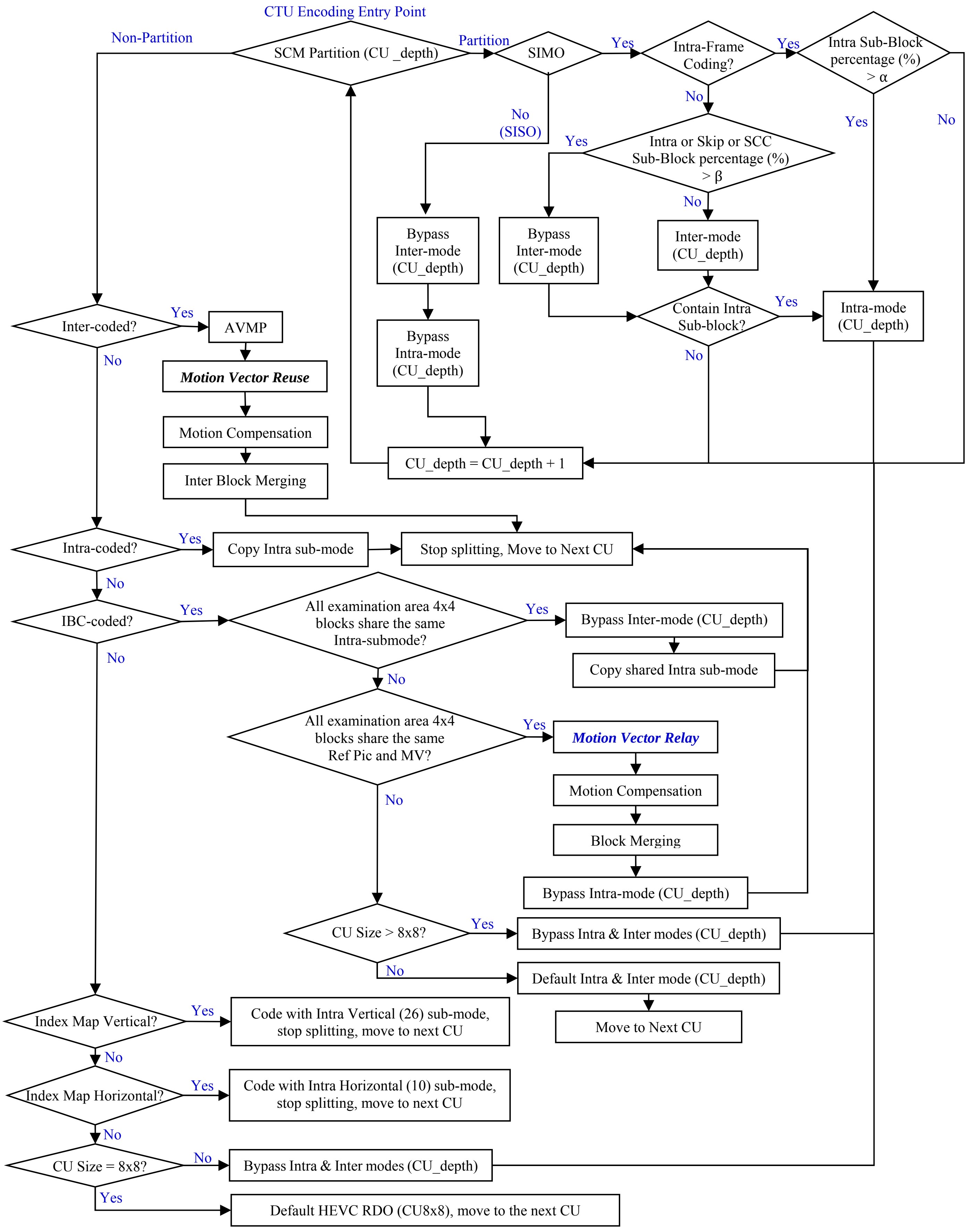

CTU Encoding Entry Point
Non-Partition
SCM Partition (CU _depth)
Partition
SIMO
Yes
Intra-Frame Coding?
Yes
Intra Sub-Block percentage (%) > α
No
No (SISO)
Yes
Intra or Skip or SCC Sub-Block percentage (%) > β
Yes
No
No
Bypass Inter-mode (CU_depth)
Bypass Inter-mode (CU_depth)
Inter-mode (CU_depth)
Bypass Intra-mode (CU_depth)
Contain Intra Sub-block?
Yes
Intra-mode (CU_depth)
No
Inter-coded?
Yes
AVMP
No
Motion Vector Reuse
Motion Compensation
Inter Block Merging
CU_depth = CU_depth + 1
Intra-coded?
Yes
Copy Intra sub-mode
Stop splitting, Move to Next CU
No
IBC-coded?
Yes
All examination area 4x4 blocks share the same Intra-submode?
Yes
Bypass Inter-mode (CU_depth)
Copy shared Intra sub-mode
No
No
All examination area 4x4 blocks share the same Ref Pic and MV?
Yes
Motion Vector Relay
Motion Compensation
Block Merging
No
Bypass Intra-mode (CU_depth)
CU Size > 8x8?
Yes
Bypass Intra & Inter modes (CU_depth)
No
Default Intra & Inter mode (CU_depth)
Move to Next CU
Index Map Vertical?
Yes
Code with Intra Vertical (26) sub-mode, stop splitting, move to next CU
No
Index Map Horizontal?
Yes
Code with Intra Horizontal (10) sub-mode, stop splitting, move to next CU
No
CU Size = 8x8?
No
Bypass Intra & Inter modes (CU_depth)
Yes
Default HEVC RDO (CU8x8), move to the next CU

Figure 5. Ultra-Fast SCC-HEVC Transcoding System Workflow Diagram

Firstly, over the flat, smooth or directional SC blocks, HEVC and SCC will both use Intra mode without further partitions. Therefore, the transcoder may directly copy the Intra sub-mode from SCC bitstream and apply to HEVC.

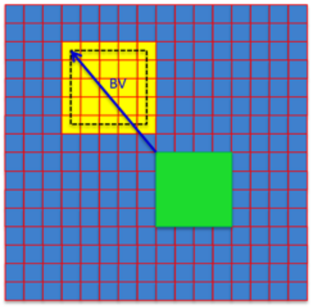


Figure 6. Block Vector Reuse for IBC-Intra Mode Mapping
Green Block: the current CU; Black Block in dashed line: the matching Block; Blue Arrow: IBC Block Vector; Yellow Blocks: Mode Examination Area; Each small square represents a 4x4 image block.

Secondly, over IBC-coded blocks, the decoded block vectors (BV) can be utilized to locate the matching block in the previously-coded area or in the reference frame. Considering that the matching block is the same as or similar to the current CU, therefore, our transcoder can copy or predict the mode and partition from the matching block and its neighbors. As shown in Figure 6, if all the 4x4 blocks covered by the matching block (inside yellow "examination area") are coded using the same Intra sub-mode, the current CU (marked as a green box) can directly copy this Intra sub-mode and terminate CU splitting. If all the 4x4 blocks are Inter-coded and share the same MV, the same reference picture list and reference frame index, the current CU can directly reuse the MV and reference frame, as illustrated in Figure 7. The final motion vector mv_{final} with respect to the reference frame can be calculated as in (1), where bv denotes the block vector from the current CU to its IBC matching block (IBC-MB) inside the current frame and $mv_{matching}$ denotes the motion vector from the IBC-MB to its Inter-frame matching block (Inter-MB) in the previously-coded frame. The sum of the two terms indicates the final motion vector from the current CU to its temporal matching block (i.e., the spatial-temporal relayed translational offset). Otherwise, if the 4x4 blocks do not share the same coding mode or the same motion vector, the current CU will be directly partitioned without going through the current level CU RDO processing.

$$mv_{final} = mv_{matching} + bv \qquad (1)$$

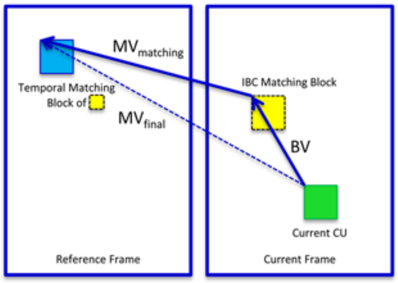


Figure 7. Block Vector Reuse for IBC-Inter Mode Mapping
Green Box: the current CU; Yellow Box: IBC Matching Block; Blue Box: IBC Matching Block's Inter-frame matching block; Blue Dashed Line: the final "relayed" motion vector from the current block to the temporal matching block.

Thirdly, over PLT-coded blocks, the decoded index map (IM) reflects the CU structure and texture directionality, as shown in Figure 8. When the index map structure is flat, horizontal or vertical, during transcoding, the HEVC encoder can directly trigger the corresponding Intra sub-mode and then terminate CU splitting. Otherwise, the encoder can safely bypass the Intra mode coding at the current CU depth.

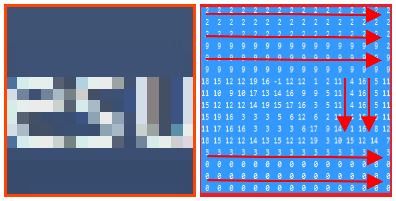


Figure 8. Sample PLT Block and Corresponding Index Map Illustration

Fourthly, over temporally-predictable blocks, both HEVC and SCC use Inter mode (e.g., *Merge*, *Skip* or *Inter*) for coding. Therefore, the transcoder may directly reuse the motion vector (MV) decoded from the SCC bitstream. Though *Merge*, *Skip* and *Inter* modes are neighbor-dependent (i.e., the same block may derive a different motion vector when its AMVP or merge candidates change), however, the MV derivation can be mostly bypassed and therefore the major complexity can be saved during the motion estimation (ME) stage.

Finally, as previously illustrated in Figure 3, PLT and IBC modes enable the "inhomogeneous" blocks to be encoded at a larger CU size. Therefore, an intuitive yet safe transcoding heuristic is that the coding depth in HEVC should be greater than the depth in SCC over the same block. For example, in Figure 3, the optimal coding depth for this SC block is 3 using SCC modes but 4 using only HEVC Intra mode.

To summarize, in our proposed framework, the Intra-mode and Inter-mode mode decisions are directly inherited from the SCC bitstream. To be specific, HEVC Intra coding will directly copy SCC Intra sub-mode. HEVC Inter coding directly reuses the decoded motion vector. Even though "Advanced Motion Vector Prediction" (AMVP) and "Block Merging" are both dependent on the spatial and temporal candidates, however, the most computationally-expensive motion estimation (ME) stage, including Enhanced Predictive Zonal Search (EPZS) and pixel interpolation can be avoided. To better preserve rate-distortion performance, merge/skip mode is always evaluated since it is computationally-light and efficient.

For the blocks coded in PLT mode, for simplicity, fast termination is implemented only over blocks with horizontal and vertical patterns, which are dominant in SC videos, whereas the non-directional blocks (e.g., text, icon, etc.) have to be split into very small Intra blocks to be homogeneous, and therefore can be safely fast-bypassed at larger CU sizes. Please note that sometimes flat blocks are sporadically coded using PLT mode. Such blocks can be treated as a special horizontal or vertical block in our framework.

For the blocks coded in IBC mode, three fast coding scenarios are considered. If (a) the blocks in examination area are all coded in the same Intra mode, the current CU can directly copy this Intra sub-mode without further partitioning. If (b) the blocks in the examination area are coded using the same temporal candidate block (i.e., requiring the same reference list, reference picture index and motion vector), then the "relayed" motion vector (as illustrated in Figure 7) will be used to derive the final MV from the current CU to its temporal matching block. Since IBC mode is used frequently in both Intra-frame and Inter-frame coding, such design provides a significant speedup. If neither condition (a) nor (b) is met, the IBC-coded block can be directly partitioned. Please note that in SCM-4.0, IBC and Inter are unified for hardware re-utilization. Namely, IBC mode is treated as a special Inter mode with the reference frame restricted to the current frame and the reference area restricted to the previously-encoded area in the current frame. Therefore, for the "mixed" blocks (i.e., partially coded in Inter mode and partially coded in IBC mode), we still treat them as "partitioned blocks" and bypass the current level RDO.

V. SINGE-INTPUT-MULTIPLE-OUTPUT SC TRANSCODING

To accommodate heterogeneous end users over different networks (e.g., WiFi, LTE, etc.), as illustrate in Figure 9, video contents are typically generated with multiple copies coded in different quality levels (e.g., spatial resolution, frame rate, bitrate, etc.), to support the adaptive video streaming services, in which subscribers could request the most suitable version given the network and device conditions, such as the bandwidth, display resolution, battery life, computing capacity, etc.

SC transcoding is one of the most straightforward solutions, where a high-quality bitstream can be utilized to produce multiple bitstreams with reduced quality levels using different combinations of spatial and temporal resolutions and bitrates. The generated bitstreams can be chunked into video fragments to feed in adaptive streaming frameworks, such as the HTTP Live Streaming (HLS) [48], the Dynamic Adaptive Streaming over HTTP (DASH) [49], etc.

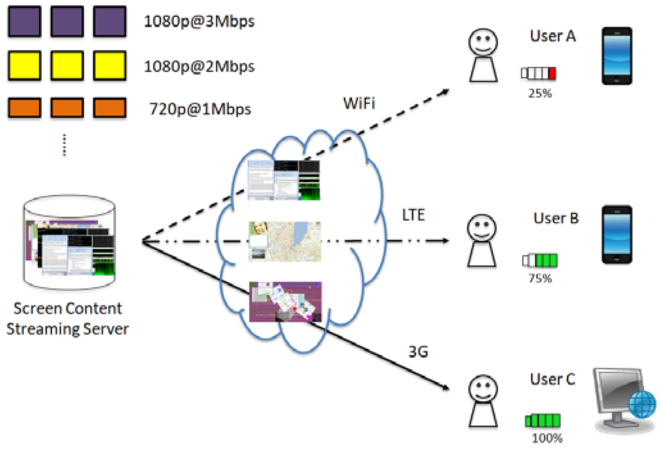


Figure 9. Illustration of on-demand SC video streaming
(Multiple copies of the same content with different quality levels are archived at the SC streaming server, and adaptation is performed according to the end client conditions to decide the video tier to request.)

Intuitively, the overall SC transcoding complexity is roughly n × the complexity of transcoding a single SC bitstream, where n is the total number of different quality levels required to be achieved at the streaming server. (Please note n is a loose approximation, e.g., the transcoding may introduce spatial and temporal resolution reduction.) Single-Input-Single-Output (SISO) scheme is inefficient and it imposes significant system complexity, buffer storage, processing delay, and backbone bandwidth (since all SC video copies are likely to be requested by the end users.) As an alternative, leveraging on our previous work [50], we propose an SC Single-Input-Multiple-Output (SIMO) framework, to transcode a single high-quality SC video stream into multiple HEVC bitstreams in different qualities. Exploiting the side information from the incoming bitstream and the correlations among the output videos, the re-encoding complexity is significantly reduced, while simultaneously the coding efficiency is well-preserved. Besides the computational complexity, the proposed framework also reduces the system processing delay and potentially would decrease the backbone network traffic from the central SC content server to the edge tower, where the SIMO transcoder is deployed to respond to different user requests.

Different from [50], in this work, SIMO is implemented between two heterogeneous bitstreams (i.e., SCC and HEVC). The SC content characteristics and codec behaviors are taken into account when we customize the SC transcoding algorithm. In this work, we only consider the quality-based SCC-HEVC transcoding, in which a high-quality HEVC-SCC bitstream is transcoded into multiple HEVC bitstreams with reduced qualities. Specifically, following the JCTVC common test conditions [46], in our configuration, a high-bitrate SCC-coded bitstream (i.e., coded with QP=22) is transcoded into four HEVC bitstreams (i.e., QP=22, 27, 32, 37, respectively). In [50], a simple yet effective heuristic is used to relate the HEVC depth decisions among coding bitrates, as summarized in (2), where $\mathrm{d(L)}$, $\mathrm{d(H)}$ and $\mathrm{d(I)}$ represent the coding depths of the low bitrate, high bitrate and input bitrate, respectively.

$$\mathrm{d(L)} \leq \mathrm{d(H)} \leq \mathrm{d(I)} \tag{2}$$

In our configuration, the same relationship holds among the generated HEVC bitstreams, as shown in (3), where $\mathrm{d(QP_{37})}$, $\mathrm{d(QP_{32})}$, $\mathrm{d(QP_{27})}$, $\mathrm{d(QP_{22})}$ denote the coding depths of QP=37, 32, 27 and 22, respectively.

$$\mathrm{d(QP_{37})} \leq \mathrm{d(QP_{32})} \leq \mathrm{d(QP_{27})} \leq \mathrm{d(QP_{22})} \tag{3}$$

A similar approach as in [50] could be used to accelerate transcoding. Firstly, the transcoder caches the coding depths of QP=22 bitstream. To encode QP=37 bitstream, the maximum coding depths can be narrowed down by (3). Finally, to encode QP=27 and QP=32, the coding depth upper bound, i.e., $\mathrm{d(QP_{22})}$, and lower bound, i.e., $\mathrm{d(QP_{37})}$, can be implicitly retrieved from the previous coding decisions in QP=22 and QP=37 bitstreams. Such approach is general and can be used safely regardless of video contents. However, such framework imposes sequential dependencies among generated bitstreams. For example, to accelerate QP=27 encoding, depth decisions of both QP=22 and QP=37 are needed. Therefore, this approach is more useful for sequential transcoding and the coding decisions need to be cached for future lookup during transcoding.

In this work, a parallel transcoding scheme is proposed to directly convert SCC bitstream into multiple HEVC bitstreams. Based on our simulation statistics, the SC blocks are relatively insensitive to QP settings. On one hand the partition decisions of SC blocks coded with different QP settings are very similar. On the other hand, a minor SC block partition mismatch will not introduce visible BD-Rate difference. As demonstrated in Figure 10, SC videos usually contain a large proportion of "QP-insensitive areas" (QPIA). For Intra-frame coding (in the central column), in "Desktop", the QPIA percentage is the highest (i.e., 94.31%). In "SlideShow", the QPIA percentage is the lowest but still significant (i.e., 67.06%). In "QP-sensitive area" (QPSA), the majority of blocks are coded in Intra-mode. For Inter-frame coding (in the right column), the QPIA percentage varies depending on the content temporal variations. For example, the QPIA percentages are 92% and 56% for "Desktop" and "Console" sequences, respectively.

Accordingly, in our SIMO transcoding configuration, the QP=22 HEVC bitstream can be directly transcoded using the default SISO algorithm as illustrated in Figure 5 (i.e., QP=22 SIMO and SISO cases are identical). For the other bitstreams (i.e., QP=27, 32 and 37), additional mode checking are applied over the partitioned blocks in the SCC bitstream (with QP=22), particularly those blocks consisting of "sensitive" sub-blocks, as illustrated in Figure 5 under the SIMO workflow.

For Intra-frame coding, if the current block is mostly coded with SCC-coded sub-blocks, the current block is more likely to be a "QP-insensitive" SC block and can be directly partitioned. Otherwise, if the current block is mostly coded with Intra sub-blocks, this block is more likely a "QP-sensitive" natural image block and therefore goes through the current level Intra mode. For Inter-frame coding, if the current block contains a large proportion of Intra sub-blocks, indicating there is no good temporal matching block available, this block can safely bypass the Inter mode at the current CU depth. If the current block contains a large proportion of Skip or SCC mode, such block is more likely to be a "QP-insensitive" SC block and therefore can be directly partitioned.

For simplicity, we introduce two tuning parameters α and β, as shown in Figure 5 under the SIMO workflow. α and β are pre-defined percentage thresholds used to examine the current block quantization sensitivity for Intra-frame and Inter-frame, respectively. The two parameters can be tuned to trade off the additional mode checking complexity and the coding efficiency. As shown in Table VIII, larger α or smaller β setting leads to additional Intra or Inter mode-bypass and therefore boosts the complexity-saving, but compromises the coding efficiency. Smaller α or larger β configuration leads to additional mode checking at larger block sizes and therefore preserves the coding performance better but simultaneously degrades the complexity reduction. In our configuration, α and β are chosen empirically with values of 0.9 and 0.5, respectively.

TABLE VIII
QUANTIZATION SENSITIVITY THRESHOLD JUSTIFICATION COMPARED WITH DEFAULT SETTING (α=0.9, β=0.5)

Sequence	All-Intra (α=0.8)		Low-Delay (β=0.3)	
	BD-Rate Saving	Complexity Increase	BD-Rate loss	Complexity Decrease
ChineseEditing	0.02%	0%	0.58%	6%
Desktop	0.10%	2%	0.54%	7%
Console	0.04%	1%	1.40%	11%
WebBrowsing	0.11%	4%	0.01%	3%
Map	0.13%	4%	0.11%	3%
Programming	0.11%	1%	0.62%	8%
SlideShow	0.07%	2%	0.40%	5%
BasketballScreen	0.14%	8%	0.56%	7%
MissionControlClip2	0.09%	5%	0.49%	4%
MissionControlClip3	0.05%	5%	1.25%	10%

Besides, our proposed SCC-HEVC transcoding algorithm is self-adaptive. For example, if the matching block region coding depths are adjusted due to the QP change, the current IBC block coding depth will be automatically updated, following our IBC mode mapping algorithms illustrated in Figure 6 and 7.

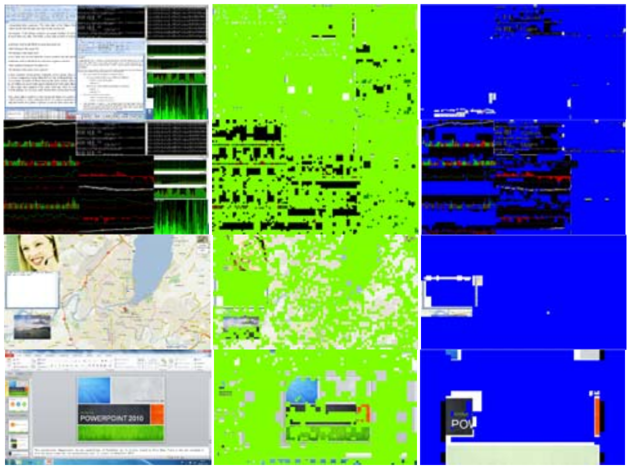


Figure 10. HEVC Decision Sensitivity Illustration between QP=22 and QP=37 Left: Sample Video Frames from "Desktop", "Console", "Map", "SlideShow"; Middle: Intra-frame Sensitivity Map using green blocks indicating consistent mode and partition decisions between QP=22 and QP=37; Right: Inter-frame Sensitivity Map using blue blocks indicating consistent mode and partition decisions between QP=22 and QP=37.

VI. Experimental Results

Our proposed SCC-HEVC fast transcoding framework is evaluated and compared with HM-16.4 anchor software for re-encoding performance evaluation.

For SISO configuration, 10 JCTVC standard SC sequences are coded using SCM-4.0 following the CTC [46], with 4 QPs (i.e., 22, 27, 32 and 37). At the transcoder, the SCC bitstreams are decoded into individual YUV videos with side information cached. Finally, our proposed transcoder will load the cached side information and re-encode the decoded videos into HEVC bitstreams in the same QP for AI and LD configurations.

For SIMO configuration, 10 JCTVC standard SC sequences are coded using SCM-4.0 with QP=22. At the transcoder, the SCC bitstream is decoded into a YUV video (corresponding to QP=22) with side information cached. Finally, our proposed transcoder will load the side information from the decoded SCC bitstream and re-encode the decoded videos (corresponding to QP=22) into HEVC using QP=22, QP=27, QP=32 and QP=37, respectively, for AI and LD configurations.

The coding performances are evaluated using homogeneous Windows 7 (64-bit) desktops with Intel-i5 CPU (2.67 GHz dual cores) and 4GB RAM.The coding efficiency is measured using BD-Rate [51]. The complexity saving is measured directly using the relative reduction of the re-encoding times, as defined in (4), where T_{Anchor} is the re-encoding time using HM-16.4 encoder software and $T_{Proposed}$ is the re-encoding time of our proposed framework.

$$\Delta C = \frac{T_{Anchor} - T_{Proposed}}{T_{Anchor}} \times 100\% \qquad (4)$$

Compared with HM-16.4 anchor re-encoding, our proposed fast transcoding framework can achieve complexity reductions of 51% and 49% for AI re-encoding and 82% and 76% for LD re-encoding using SISO and SIMO configurations, respectively. Given the space limitation, only the YUV-444 results are provided. However, the proposed framework can be easily

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2874475, IEEE Transactions on Circuits and Systems for Video Technology

generalized to RGB4:4:4 color space and YUV4:2:0 sampling format. In Table IX and X, detailed simulation results for SISO and SIMO configurations are provided. From the experimental results, the following conclusions can be drawn:

1. The proposed fast SCC-HEVC transcoding framework achieves a remarkable transcoding speedup. Please note that the proposed framework is purely software-based and therefore can be further improved with hardware acceleration. Besides, in this work, only the high-level framework is presented without specifically optimizing individual encoding module. Therefore, other fast video encoding or transcoding algorithms can be incorporated into our framework for an additional speedup.

2. The Intra-frame coding acceleration mainly comes from Intra mode reuse and BV-based fast mode and partition reuse. Besides, the fast directional Intra mode selection over the PLT-coded blocks also provides a visible speedup (i.e., ranging from 1%-5%). The speedup ratio also depends on the contents. For sequences mainly coded in larger blocks, e.g., "SlideShow", the complexity reduction is more (≈ 70%), whereas over sequences mainly coded using smaller blocks, e.g., "Map", the complexity reduction is less (≈50%).

3. The Inter-frame coding acceleration mainly comes from the MV reuse and the CU fast bypass/termination. Since SCC uses full-frame hash-based IBC search over CU8 blocks and Inter-frame hash-based motion search, the inherited MVs from SCC bitstream sometimes outperform the local ME results in the anchor HEVC configuration (with a default search range of 64). Therefore, over some sequences (e.g., "WebBrowsing"), we observe a significant BD-Rate saving even after the transcoding acceleration. Besides, the "motion vector relay" technique applied over IBC blocks provides a significant speedup (i.e., ranging from 6%-18%), depending on the IBC utilization in the Inter-frames.

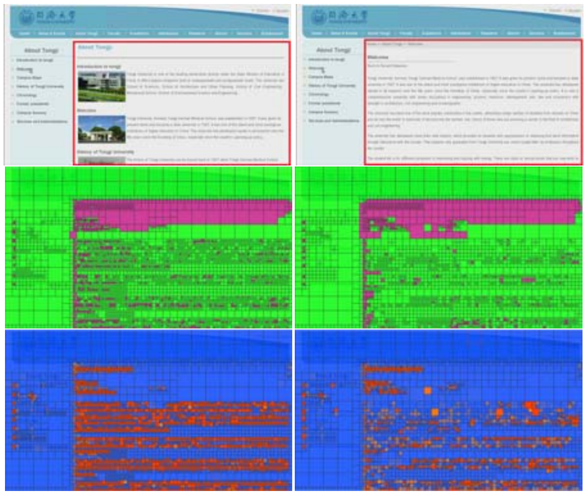


Figure 11. SCC-HEVC Transcoding Analysis of "WebBrowsing" (QP=22) Top Left: 8th Frame; Top Right: 9th Frame; Mid Left: Anchor Transcoding Mode and Partition Decisions; Mid Right: Proposed Transcoding Mode and Partition Decisions; Bottom Left: Anchor Transcoding Bit-Allocation Heatmap. Bottom Right: Proposed Transcoding Bit-Allocation Heatmap. Red box: Intra-coded blocks; Green box: Inter-coded blocks; Blue box: low-bit blocks; Orange box: high-bit blocks with color depth indicating bit-consumption level.

4. The proposed transcoding framework outperforms anchor HEVC Inter-frame coding significantly over the screen content regions with dominant temporal changes (e.g., scene-cut). As shown in Table IX, for "WebBrowsing" sequence, a 48% BD-Rate saving is achieved after transcoding speedup. This significant gain is achieved from several transition frames, in which the inherited MVs from SCC bitstreams significantly outperform the MVs derived from HEVC restricted motion search. As illustrated in Figure 11, during the content transition (e.g., from 8th frame to 9th frame) in the webpage bottom panel (as enclosed inside the red square), the Inter-frame mode and partition distribution have changed drastically. Since SCM uses hash-based IBC and Inter search, therefore the inherited motion vectors are more accurate than anchor HEVC ME candidates. Consequently, more blocks are skip-coded or merge-coded and the bit-consumption is much lower (as shown clearly from the heatmap). For this exemplar transition frame, anchor HEVC spends 566344 bits while our proposed transcoding algorithm only spends 164272 bits. Similar behaviors are also observed over "Desktop", "Console", etc.

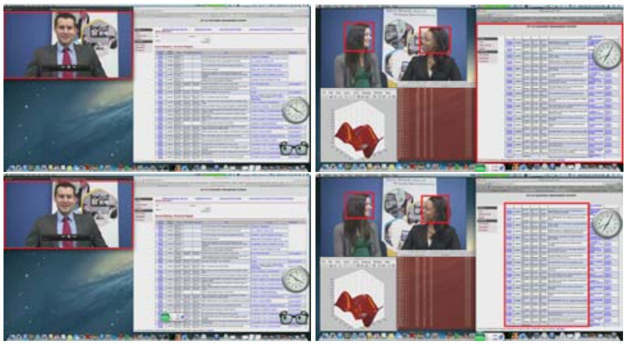


Figure 12. Sample frames in "MissionControlClip2" (left) and "MissionControlClip3" (right). Top and bottom: 1st and 11th frame, respectively. Red bounding boxes indicate regions with large temporal motion.

5. For "Mixed-Content" Inter-frame coding, for example, "MissionControlClip2", "MissionControlClip3", depending on the content temporal variation locations, different behaviors are observed. As shown in Figure 12, though the two SC sequences appear similar, in "MissionControlClip2", mainly the natural region (e.g., the man enclosed in the red box) has temporal motion, while the text region on the right side is relatively static. In such case, the natural video region is dominated by intra mode and inter mode with local search. Therefore, after transcoding, we do not observe any BD-Rate saving. This also applies for the "BasketballScreen" sequence, in which the temporal motion is mainly located in natural video regions (e.g., the basketball player window). Over "MissionControlClip3", a large proportion of text regions have temporal motion instead. In such case, the inherited motion vectors from SCC bitstream mostly outperform the anchor HEVC local ME and therefore leads to a coding efficiency improvement. To conclude, when transcoding the SC regions (e.g., text, graphics, icon, etc.), the proposed MV inheritance and MV relay schemes using side information from SCC bitstream can outperform the anchor HEVC motion estimation results, whereas when transcoding the natural video regions (e.g., natural picture), the derived MV does not differ much from the anchor HEVC motion derivation and therefore does not introduce BD-Rate saving.

TABLE IX

CODING EFFICIENCY AND COMPLEXITY REDUCTION FOR SINGLE-INPUT-SINGLE-OUTPUT (SISO) SCC-HEVC TRANSCODING

Index	Sequence	QP	HM-16.4 (AI)			Proposed (AI)					HM-16.4 (LD)			Proposed (LD)				
			Rate	*PSNR*	*Time*	*Rate*	*PSNR*	*Time*	△*R*	△*T*	*Rate*	*PSNR*	*Time*	*Rate*	*PSNR*	*Time*	△*R*	△*T*
1	Desktop 1080p, YUV444 Text & Graphics 150-frame	22	187871	49.40	5936	188545	49.35	3331	+0.62%	-44%	3956	50.28	5550	3274	50.31	1048	-18.16%	-81%
		27	153474	44.69	5686	154063	44.63	3150			3485	45.81	5457	2869	45.81	1027		
		32	124247	39.56	5341	124647	39.51	3034			3088	40.75	5339	2516	40.80	1013		
		37	89900	34.30	4921	90135	34.24	2816			2566	35.22	5179	2095	35.31	992		
2	Console 1080p, YUV444 Text & Graphics 150-frame	22	92030	50.64	5076	92559	50.54	2603	+1.03%	-50%	9970	50.53	8492	9005	50.36	1747	-12.14%	-79%
		27	75228	45.68	4885	75652	45.62	2474			8085	45.51	8143	7143	45.42	1668		
		32	60140	40.60	4649	60503	40.53	2359			6392	39.83	7583	5575	39.99	1565		
		37	44254	34.98	4436	44502	34.74	2144			4620	34.73	6825	4028	34.79	1431		
3	ChineseEditing 1080p, YUV444 Text & Graphics 150-frame	22	260550	46.02	6366	261015	46.00	3599	+0.45%	-44%	8324	45.78	5835	7290	45.82	1166	-9.52%	-81%
		27	201712	41.54	6229	202194	41.51	3400			6062	41.34	5620	5491	41.29	1094		
		32	151439	36.98	5552	151779	36.94	3169			4088	36.63	5417	3708	36.61	1026		
		37	97739	32.36	4967	98015	32.33	2876			2340	32.02	5122	2166	32.12	941		
4	WebBrowsing 720p, YUV444 Text & Graphics 300-frame	22	31361	50.71	4411	31394	50.72	2141	+0.20%	-51%	756	50.19	4309	396	50.44	694	-47.81%	-84%
		27	24598	46.14	4187	24626	46.13	2046			599	45.37	4267	313	45.66	678		
		32	17604	42.17	3908	17632	42.19	1936			411	40.38	4198	226	41.03	668		
		37	9382	36.74	3556	9405	36.70	1775			238	35.23	4077	144	35.90	660		
5	Map 720p, YUV444 Text & Graphics 300-frame	22	68215	46.76	4846	68324	46.78	2662	+0.42%	-48%	2508	45.82	5613	2505	45.78	1019	-0.23%	-82%
		27	44655	42.41	4334	44797	42.43	2336			1563	41.66	5208	1564	41.62	923		
		32	27596	38.92	3860	27811	38.92	1968			922	38.25	4850	921	38.30	837		
		37	16962	35.90	3466	17196	35.90	1650			535	35.30	4536	536	35.43	762		
6	Programming 720p, YUV444 Text & Graphics 300-frame	22	62167	48.70	4706	62379	48.66	2383	+0.72%	-51%	6813	48.04	8192	6793	48.03	1487	-1.02%	-83%
		27	42291	44.62	4345	42454	44.58	2162			3624	43.79	7036	3618	43.84	1230		
		32	28621	40.05	4018	28706	40.01	1950			1713	39.54	6054	1713	39.67	1018		
		37	19591	35.77	3748	19713	35.74	1784			805	35.61	5282	817	35.58	866		
7	SlideShow 720p, YUV444 Text & Graphics 300-frame	22	4388	54.49	3238	4412	54.50	1050	+1.17%	-69%	845	51.57	6439	863	51.70	1158	+1.36%	-82%
		27	2903	50.43	3101	2928	50.43	971			488	47.64	5881	502	47.79	1047		
		32	2005	46.21	3009	2025	46.19	902			286	43.63	5433	294	43.69	956		
		37	1370	41.94	2918	1389	41.78	838			172	39.53	5064	177	39.59	869		
8	BasketballScreen 1440p, YUV444 Mixed-Content 150-frame	22	212370	48.86	4669	212897	48.86	2325	+0.44%	-52%	7220	48.92	5315	7339	48.61	955	+1.96%	-82%
		27	150110	44.83	4343	150586	44.83	2084			3906	45.25	4861	3930	45.25	853		
		32	102654	40.71	4043	103023	40.70	1913			2319	41.32	4493	2338	41.31	796		
		37	63978	36.55	3735	64275	36.52	1705			1388	37.05	4415	1427	36.90	747		
9	MissionControlClip2 1440p, YUV444 Mixed-Content 150-frame	22	221700	50.18	4670	222303	50.09	2241	+0.43%	-53%	3813	50.21	4392	3828	50.11	761	+0.93%	-83%
		27	164565	45.23	4374	164903	45.21	2053			2545	45.69	4171	2550	45.68	716		
		32	114858	40.68	4094	115189	40.67	1901			1700	41.55	4011	1707	41.46	681		
		37	70381	36.11	3805	70635	36.10	1720			1048	37.02	3947	1052	36.96	660		
10	MissionControlClip3 1080p, YUV444 Mixed-Content 150-frame	22	165880	49.06	5313	166182	49.05	2768	+0.19%	-48%	3268	48.25	5694	2955	48.33	1029	-12.76%	-82%
		27	123646	44.27	5033	124060	44.30	2598			2123	44.16	5168	1876	44.24	916		
		32	87469	39.64	4650	87762	39.66	2426			1401	39.71	4795	1219	39.79	843		
		37	55290	35.22	4304	55560	35.24	2214			890	35.18	4548	779	35.25	783		
							Average		+0.57%	-51%					Average		-9.74%	-82%

Rate: in kbps; ***PSNR***: Y-component PSNR in dB; ***Time***: Re-encoding time in Second. ***△R***: BD-Rate Increment in Percentage. ***△T***: Encoding Time Reduction in Percentage.

6. We investigate the source of the achieved complexity reductions to understand the contributions from HEVC mode reuse (e.g., Intra mode copy, Inter motion copy) and SCC mode mapping (e.g., IBC block vector reuse, IBC motion relay, PLT block mapping, etc.). As shown in Table XI, for All-Intra configuration, beyond the Intra mode reuse (i.e., transcoding SCC Intra-block to HEVC Intra-block), the IBC block vector reuse and PLT mode index map directionality inference contribute to 23% additional complexity reduction and introduce only 0.43% BD-Rate loss on average. Since SCC mode usage is dominant in Intra-frame coding, fast SCC mode mapping proves very effective. For Low-Delay configuration, beyond the Intra mode reuse and Inter motion reuse, the IBC motion relay technique contributes to 6% additional complexity saving and sometimes outperforms HEVC default motion estimation with 1.10% average BD-Rate saving. Compared with the Intra-frame statistics, the Inter-frame complexity reduction is less, because a large proportion of screen content blocks are encoded using Skip mode and the SCC mode percentage is relatively small in Inter-frames.

7. The proposed parallel SIMO SC transcoding significantly reduces the transcoding complexity, calculated using the sum of the re-encoding times for 4 QP settings between the anchor HEVC encoder and our proposed framework. The overall complexity saving in SIMO configuration is 6% less than the SISO case due to the additional mode checking over the partitioned blocks. For the sequences dominated by temporal motion over SC region, e.g., "Desktop", "Console", the proposed framework preserves the coding performance better. For the sequences dominated by temporal motions in the natural video regions, the proposed framework introduces relatively larger but still marginal BD-Rate losses, e.g., "SlideShow", "BasketballScreen", etc. Compared with the basic FDFE transcoding solution (in which the server can also distribute and transcode multiple video tiers in different bitrates in parallel), the proposed SIMO framework achieves up to 76% average complexity reduction (as detailed in Table X). Compared with the other SIMO fast transcoding framework proposed in [50], which uses the block depth side information from other decoded bitstreams to accelerate the current bitstream encoding, the proposed SIMO solution makes fast decisions directly from the high-quality source bitstream without any dependency on other bitstreams. For example, following the transcoding order proposed in [50], the video in intermediate quality requires and has to wait for the coding depth decisions from both high-QP and low-QP bitstreams before being fast transcoded. The video transcoded earlier is less accelerated than the video transcoded later due to the amount of available side information (e.g., coding depths). In contrast, the proposed SIMO framework in this work is entirely paralleled and therefore can significantly reduce the end-to-end processing delay.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2874475, IEEE Transactions on Circuits and Systems for Video Technology

TABLE X
CODING EFFICIENCY AND COMPLEXITY REDUCTION FOR SINGLE-INPUT-MULTIPLE-OUTPUT (SIMO) SCC-HEVC TRANSCODING

Index	Sequence	QP	HM-16.4 (AI)			Proposed (AI)					HM-16.4 (LD)			Proposed (LD)				
			Rate	*PSNR*	*Time*	*Rate*	*PSNR*	*Time*	△*R*	△*T*	*Rate*	*PSNR*	*Time*	*Rate*	*PSNR*	*Time*	△*R*	△*T*
1	Desktop 1080p, YUV444 150-frame	27	153523	44.57	5690	154108	44.52	3119	+0.64%	-43%	3482	45.67	5449	2858	45.66	1245	-18.13%	-78%
		32	124748	39.48	5348	125202	39.43	2956			3084	40.65	5328	2512	40.67	1207		
		37	90918	34.31	4955	91230	34.24	2754			2561	35.25	5179	2100	35.28	1165		
2	Console 1080p, YUV444 150-frame	27	75239	45.64	4906	75692	45.60	2484	+1.08%	-50%	7942	45.37	8128	6992	44.97	2594	-10.23%	-72%
		32	59814	40.56	4648	60212	40.49	2313			6265	39.63	7605	5499	39.23	2338		
		37	44349	34.51	4303	44641	34.20	2143			4593	34.25	6877	4016	33.81	2090		
3	ChineseEditing 1080p, YUV444 150-frame	27	201496	41.36	6000	201966	41.33	3440	+0.53%	-43%	5805	41.08	5574	5405	40.86	1332	-3.66%	-78%
		32	151489	36.78	5538	151825	36.73	3234			3871	36.30	5547	3728	35.89	1234		
		37	99577	31.96	5042	99969	31.89	2907			2407	31.50	5308	2311	31.01	1129		
4	WebBrowsing 720p, YUV444 300-frame	27	24637	45.18	4344	24669	45.11	2089	+0.55%	-51%	590	44.59	4273	303	44.69	764	-47.90%	-83%
		32	17684	41.15	3969	17714	41.10	1955			409	39.72	4195	212	39.83	732		
		37	9516	35.75	3639	9556	35.72	1793			240	34.68	4089	138	34.52	710		
5	Map 720p, YUV444 300-frame	27	45064	40.88	4397	45311	40.88	2530	+1.03%	-44%	1589	40.06	5271	1593	39.94	1559	+1.99%	-74%
		32	28397	36.42	4038	28770	36.40	2260			953	35.54	4921	964	35.41	1440		
		37	17608	32.79	3530	18076	32.71	2066			565	31.97	4607	578	31.79	1279		
6	Programming 720p, YUV444 300-frame	27	43065	43.73	4426	43255	43.67	2329	+1.00%	-48%	3800	42.85	7142	3830	42.73	2326	+2.69%	-72%
		32	29224	39.11	4050	29419	39.06	2160			1846	38.22	6172	1894	38.18	2025		
		37	19921	34.76	3762	20157	34.72	2012			873	33.98	5379	907	33.89	1687		
7	SlideShow 720p, YUV444 300-frame	27	2915	48.18	3174	2936	48.15	1095	+1.26%	-67%	504	46.11	5919	513	45.95	1869	+3.07%	-73%
		32	2025	44.19	3116	2052	44.15	1024			293	41.85	5459	298	41.70	1689		
		37	1376	40.07	3053	1409	40.04	978			179	37.77	5097	189	37.52	1510		
8	BasketballScreen 1440p, YUV444 150-frame	27	151330	43.20	4421	151976	43.17	2363	+0.68%	-47%	4059	43.34	4969	4095	43.29	1330	+3.34%	-76%
		32	104209	38.94	4068	104736	38.91	2192			2405	39.22	4551	2420	39.17	1181		
		37	66348	34.82	3778	66979	34.79	2021			1452	35.13	4307	1467	35.10	1058		
9	MissionControlClip2 1440p, YUV444 150-frame	27	165079	43.76	4423	165562	43.72	2376	+0.60%	-48%	2583	44.08	4208	2615	44.04	852	+1.93%	-81%
		32	115740	39.34	4112	116261	39.33	2211			1728	39.91	4045	1765	39.89	800		
		37	72000	34.91	3829	72567	34.89	2056			1074	35.71	3958	1110	35.67	765		
10	MissionControlClip3 1080p, YUV444 150-frame	27	124103	43.02	5059	124638	43.02	2866	+0.43%	-45%	2156	42.93	5229	1959	42.82	1293	-7.09%	-78%
		32	87902	38.43	4679	88358	38.43	2650			1393	38.50	4843	1300	38.33	1159		
		37	55861	33.92	4298	56320	33.95	2450			906	33.99	4582	841	33.76	1058		
							Average		+0.78%	-49%					Average		-7.40%	-76%

Rate: in kbps; ***PSNR***: Y-component PSNR in dB; ***Time***: Re-encoding time in Second. ***△R***: BD-Rate Increment in Percentage. ***△T***: Encoding Time Reduction in Percentage
Note: QP=22 HEVC bitstreams are directly transcoded from QP=22 SCC bitstreams. Therefore, the QP=22 coding results are identical to SIMO case in Table **IX**.

TABLE XI
CODING EFFICIENCY AND COMPLEXITY REDUCTION COMPARED WITH DIRECT INTRA MODE REUSE AND INTER MOTION REUSE

Index	Sequence	QP	HEVC mode reuse (AI)			HEVC mode reuse and SCC mode mapping (AI)					HEVC mode reuse (LD)			HEVC mode reuse and SCC mode mapping (LD)				
			Rate	*PSNR*	*Time*	*Rate*	*PSNR*	*Time*	△*R*	△*T*	*Rate*	*PSNR*	*Time*	*Rate*	*PSNR*	*Time*	△*R*	△*T*
1	Desktop 1080p, YUV444 Text & Graphics 150-frame	22	187885	49.40	4186	188545	49.35	3331	+0.56%	-21%	3285	50.33	1134	3274	50.31	1048	-1.04%	-6%
		27	153494	44.70	3998	154063	44.63	3150			2903	45.89	1111	2869	45.81	1027		
		32	124275	39.55	3812	124647	39.51	3034			2546	40.84	1096	2516	40.80	1013		
		37	90026	34.30	3559	90135	34.24	2816			2131	35.28	1067	2095	35.31	992		
2	Console 1080p, YUV444 Text & Graphics 150-frame	22	92094	50.58	3826	92559	50.54	2603	+0.60%	-32%	8785	50.40	1901	9005	50.36	1747	+0.81%	-11%
		27	75399	45.66	3660	75652	45.62	2474			7057	45.46	1768	7143	45.42	1668		
		32	60320	40.56	3499	60503	40.53	2359			5550	39.93	1639	5575	39.99	1565		
		37	44341	34.86	3203	44502	34.74	2144			4040	34.73	1497	4028	34.79	1431		
3	ChineseEditing 1080p, YUV444 Text & Graphics 150-frame	22	260808	46.02	4572	261015	46.00	3599	+0.28%	-22%	7194	45.78	1317	7290	45.82	1166	+1.11%	-7%
		27	201896	41.52	4340	202194	41.51	3400			5379	41.23	1237	5491	41.29	1094		
		32	151550	36.96	4075	151779	36.94	3169			3603	36.56	1138	3708	36.61	1026		
		37	97849	32.35	3638	98015	32.33	2876			2245	32.07	1031	2166	32.12	941		
4	WebBrowsing 720p, YUV444 Text & Graphics 300-frame	22	31373	50.72	2969	31394	50.72	2141	+0.23%	-28%	429	50.32	755	396	50.44	694	-10.76%	-8%
		27	24607	46.13	2892	24626	46.13	2046			343	45.42	745	313	45.66	678		
		32	17612	42.18	2761	17632	42.19	1936			249	40.70	736	226	41.03	668		
		37	9392	36.77	2423	9405	36.70	1775			158	35.53	713	144	35.90	660		
5	Map 720p, YUV444 Text & Graphics 300-frame	22	68282	46.78	3007	68324	46.78	2662	+0.35%	-17%	2502	45.78	1078	2505	45.78	1019	-0.13%	-4%
		27	45289	40.88	2735	44797	42.43	2336			1564	41.63	967	1564	41.62	923		
		32	28753	36.40	2457	27811	38.92	1968			924	38.30	848	921	38.30	837		
		37	18074	32.75	2243	17196	35.90	1650			536	35.36	770	536	35.43	762		
6	Programming 720p, YUV444 Text & Graphics 300-frame	22	62245	48.69	3065	62379	48.66	2383	+0.63%	-26%	6767	48.03	1639	6793	48.03	1487	-0.26%	-7%
		27	43172	43.70	2876	42454	44.58	2162			3613	43.82	1337	3618	43.84	1230		
		32	29403	39.11	2714	28706	40.01	1950			1716	39.65	1074	1713	39.67	1018		
		37	20149	34.73	2569	19713	35.74	1784			817	35.56	894	817	35.58	866		
7	SlideShow 720p, YUV444 Text & Graphics 300-frame	22	4399	54.54	1383	4412	54.50	1050	+0.87%	-24%	856	51.66	1253	863	51.70	1158	-0.27%	-8%
		27	2911	50.45	1261	2928	50.43	971			497	47.71	1147	502	47.79	1047		
		32	2015	46.22	1189	2025	46.19	902			292	43.60	1053	294	43.69	956		
		37	1382	41.91	1145	1389	41.78	838			176	39.45	950	177	39.59	869		
8	BasketballScreen 1440p, YUV444 Mixed-Content 150-frame	22	212751	48.87	3061	212897	48.86	2325	+0.30%	-26%	7332	48.60	996	7339	48.61	955	+0.22%	-3%
		27	150450	44.85	2800	150586	44.83	2084			3928	45.24	885	3930	45.25	853		
		32	102874	40.72	2611	103023	40.70	1913			2337	41.35	793	2338	41.31	796		
		37	64177	36.55	2369	64275	36.52	1705			1424	36.92	767	1427	36.90	747		
9	MissionControlClip2 1440p, YUV444 Mixed-Content 150-frame	22	221924	50.19	2873	222303	50.09	2241	+0.29%	-23%	3818	50.24	806	3828	50.11	761	+0.72%	-5%
		27	164749	45.23	2689	164903	45.21	2053			2548	45.69	740	2550	45.68	716		
		32	115044	40.68	2479	115189	40.67	1901			1704	41.54	723	1707	41.46	681		
		37	70534	36.11	2209	70635	36.10	1720			1049	37.00	683	1052	36.96	660		
10	MissionControlClip3 1080p, YUV444 Mixed-Content 150-frame	22	166027	49.08	3260	166182	49.05	2768	+0.22%	-17%	2968	48.34	1079	2955	48.33	1029	-1.43%	-5%
		27	123943	44.30	3133	124060	44.30	2598			1891	44.19	970	1876	44.24	916		
		32	87669	39.67	2929	87762	39.66	2426			1230	39.70	881	1219	39.79	843		
		37	55473	35.26	2679	55560	35.24	2214			786	35.15	830	779	35.25	783		
							Average		+0.43%	-23%					Average		-1.10%	-6%

Rate: in kbps; ***PSNR***: Y-component PSNR in dB; ***Time***: Re-encoding time in Second. ***△R***: BD-Rate Increment in Percentage. ***△T***: Encoding Time Reduction in Percentage.

VII. Conclusion and Future Work

In this paper, a novel HEVC-SCC fast transcoding solution is presented to efficiently bridge the state-of-art HEVC standard and its screen content coding (SCC) extension. Based on extensive statistical studies and mode mapping techniques utilizing side information extracted from the decoded SCC bitstream, the proposed transcoding framework can efficiently determine the corresponding HEVC mode and partition. Compared with the direct transcoding solution reusing Intra mode and Inter motion, the proposed mode mapping scheme introduces additional 23% and 6% complexity reductions for All-Intra and Low-Delay configurations with 0.43% BD-Rate loss and 1.10% BD-Rate saving, respectively. Our transcoding framework achieves 51% and 81% re-encoding complexity reductions under All-Intra and Low-Delay configurations, compared with the direct "Full-Decoding-Full-Encoding" solution. We also extend the proposed solution to support the single-input-multiple-output (SIMO) transcoding and achieve 49% and 76% complexity reductions under All-Intra and Low-Delay configurations, respectively. The future studies include SIMO extensions to support spatial and temporal transcoding and other standards (e.g., H.264/AVC, VP9, etc.).

References

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

[2] G. -J. Sullivan, J.-R. 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. 1648–1667, Dec. 2012.

[3] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma, Y. Ye, "Screen Content Coding Test Model 4 Encoder Description (SCM 4)," Doc. JCTVC-T1014, February 2015.

[4] C. Pang, J. Sole, L. Guo, M. Karczewicz, and R. Joshi, "Non-RCE3: Intra Motion Compensation with 2-D MVs," Doc. JCTVC-N0256, July 2013.

[5] J. Chen, Y. Chen, T. Hsieh, R. Joshi, M. Karczewicz, W.-S. Kim, X. Li, C. Pang, W. Pu, K. Rapaka, J. Sole, L. Zhang, and F. Zou, "Description of screen content coding technology proposal by Qualcomm," Doc. JCTVC-Q0031, April 2014.

[6] L. Guo, M. Karczewicz, and J. Sole, "RCE3: Results of Test 3.1 on Palette Mode for Screen Content Coding," Doc. JCTVC-N0247, July 2013.

[7] L. Zhang, J. Chen, J. Sole, M. Karczewicz, X. Xiu, Y. He, Y. Ye, "SCCE5 Test 3.2.1: In-loop Color-Space Transform," Doc. JCTVC-R0147, June 2014.

[8] X. Li, J. Sole, M. Karczewicz, "Adaptive MV precision for Screen Content Coding," Doc. JCTVC-P0283, January 2014.

[9] B. Li, J. Xu, G. J. Sullivan, Y. Zhou, B. Lin, "Adaptive motion vector resolution for screen content," Doc. JCTVC-S0085, October 2014.

[10] 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 Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), 2016 Aug; Vol: PP, Issue 99, Page:1-15.

[11] Z. Liang, Z. Li, M. Siwei, and Z. Debin, "Fast mode decision algorithm for intra prediction in HEVC," IEEE Visual Communications and Image Processing (VCIP), pp. 1–4, Tainan, 2011.

[12] W. Jiang, H. Ma, and Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012.

[13] Y. Piao, J. Min, and J. Chen, "Encoder Improvement of Unified Intra Prediction", Doc. JCTVC-C207, January 2013.

[14] M. Zhang, J. Qu, and H. Bai, "Entropy-Based fast Largest Coding Unit Partition Algorithm in High-Efficiency Video Coding," Entropy 2013, 15, 2277-2287.

[15] X. Shen, and Y. Lu, "CU splitting early termination based on weighted SVM," EURASIP Journal on Image and Video Processing (2013): 1-11.

[16] J. Hou, D. Li, Z. Li, X. Jiang, "Fast CU size decision based on texture complexity for HEVC intra coding," Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on, pp. 1096-1099, 2013.

[17] H. Zhang and Z. Ma, "Early termination schemes for fast intra prediction in high-efficiency video coding," in Proc. IEEE ISCAS, pp. 45–48, May 2013.

[18] H. Zhang and Z. Ma, "Fast Intra Mode Decision for High Efficiency Video Coding (HEVC)," IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no.4, pp. 660–668, 2014.

[19] R. Li, B. Zeng, M.L. Liu, "A New Three Step Search Algorithm for Block Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology,vol.4, no.4, pp. 438–442, 1994.

[20] L.M. Po, W.C. Ma, "A Novel Four-step Search Algorithm for Fast Block Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol.6, no.3, pp. 313–317, 1996.

[21] S. Zhu, K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," IEEE Transactions on Image Processing, vol. 9, pp. 287–290, 2000.

[22] C. H. Cheung, L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 12, pp. 1168–1177, 2002.

[23] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern forfast block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 5, pp. 349–355, 2002.

[24] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," Proc. SPIE Vis. Communication Image Processing (VCIP), vol. 4671, pp. 1069–1079, Jan. 2002.

[25] B. Li, J. Xu, "A Fast Algorithm for Adaptive Motion Compensation Precision in Screen Content Coding", in Proc. DCC, pp. 243-252, March, 2015.

[26] D. -K. Kwon and M. Budagavi, "Fast Intra Block Copy (IntraBC) Search for HEVC Screen Content Coding", in Proc. IEEE ISCAS, pp. 9-12, June, 2014.

[27] B. Li, J. Xu, and F. Wu, "A Unified Framework of Hash-based Matching for Screen Content Coding", IEEE Visual Communications and Image Processing (VCIP), pp. 530-533, 2014.

[28] K. Rapaka, and J. Xu, "Software for SCM with hash based motion search," Doc. JCTVC-Q0248, March 2014.

[29] B. Li, J. Xu, "Hash-based motion search," Doc. JCTVC-Q0245, March 2014.

[30] S. -H. Tsang, Y.-L. Chan and W. -C. Siu, "Fast and Efficient Intra Coding Techniques for Smooth Region in Screen Content Coding Based on Boundary Prediction Samples", in Proc. ICASSP, pp.1409 - 1413, 2015.

[31] D. Lee, S. Yang, H. Shim, and B. Jeon, "Fast Transform Skip Mode Decision for HEVC Screen Content Coding", in Proc. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1-4, 2015.

[32] M. Zhang, Y, Guo, and H. Bai, "Fast Intra Partition Algorithm for HEVC Screen Content Coding", in Proc. VCIP, pp. 390 - 393, 2014.

[33] 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," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 1377-1381.

[34] F. Duanmu, Z. Ma, and Y. Wang, "Fast CU partition decision using machine learning for screen content compression", in IEEE International Conference on Image Processing (ICIP), pp. 4972 - 4976, 2015.

[35] A. Vetro, C. Christopoulos, H.Sun, "Video Transcoding Architectures and Techniques: An Overview", IEEE Signal Processing Magazine, vol. 20, no. 2, pp.18-29. March 2003.

[36] E. Peixoto, B. Macchiavello, R. Queiroz, E.M. Hung, "Fast H.264/AVC to HEVC Transcoding Based on Machine Learning," International Telecommunications Symposium (ITS), pp. 1-4, 2014.

[37] E. Peixoto, B. Macchiavello, E.M. Hung, A. Zaghetto, T. Shanableh, and E. Lzquierdo, "An H.264/AVC to HEVC video transcoder based on mode mapping," IEEE International Conference on Image Processing (ICIP), pp. 1972-1976, 2013.

[38] E. Peixoto, B. Macchiavello, E.M. Hung, and R. Queiroz, "A fast HEVC transcoder based on content modeling and early termination," IEEE International Conference on Image Processing (ICIP), pp. 2532-2536, 2014.

[39] A. J. Diaz-Honrubia, J. L. Martinez, J. M. Puerta, J. A. Gamez, J. De Cock, and P. Cuenca, "Fast Quadtree Level Decision Algorithm for

H.264/HEVC Transcoder," in IEEE International Conference on Image Processing (ICIP), pp. 2497–2501, 2014.

[40] A. J. Diaz-Honrubia, J. L. Martinez, J. M. Puerta, J. A. Gamez, J. De Cock, and P. Cuenca, "A Data-Driven Probabilistic CTU Splitting Algorithm for Fast H.264/HEVC Video Transcoding," in Data Compression Conference (DCC), pp. 449, April 2015.

[41] F. Zheng, Z. Shi, X. Zhang and Z. Gao, "Effective H.264/AVC to HEVC transcoder based on Prediction Homogeneity," in IEEE Visual Communications and Image Processing Conference (VCIP), pp. 233-236, 2014.

[42] F. Zheng, Z. Shi, X. Zhang, and Z. Gao, "Fast H.264/AVC To HEVC Transcoding Based on Residual Homogeneity", in IEEE International Conference on Audio, Language and Image Processing (ICALIP), pp. 765-770, 2014.

[43] A. Nagaraghatta, Y. Zhao, G. Maxwell, and S. Kannangara, "Fast H.264/AVC to HEVC transcoding using mode merging and mode mapping", IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp. 165-169, 2015.

[44] P. Xinga, Y. Tian, X. Zhang, Y. Wang, and T. Huang, "A Coding Unit Classification based AVC to HEVC transcoding with background modeling for surveillance videos," in IEEE International Conference of Visual Communication and Image Processing (VCIP), pp. 1-6, 2013.

[45] X. Zhang, T. Huang, Y. Tian, M. Geng, S. Ma, and W. Gao, "Fast and Efficient Transcoding Based on Low-Complexity Background Modeling and Adaptive Block Classification", IEEE Transactions on Multimedia, Vol: 15, Issue: 8, pp: 1769-1785, 2013.

[46] H. Yu, R. Cohen, K. Rapaka, J. Xu, "Common Test Conditions for Screen Content Coding," Doc. JCTVC-T1015, February 2015.

[47] 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 Proc. International Conference of Image Processing (ICIP), pp. 4205-4209, Phoenix, Arizona, September 2016.

[48] R. Pantos, HTTP Live Streaming, Internet Engineering Task Force, June 2013.

[49] Media presentation description and segment formats, Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1, ISO/IEC 23009-1, 2014.

[50] C. Wang, B. Li, J. Wang, H. Zhang, H. Chen, Y. Xu, and Z. Ma, "Single-Input-Multiple-Ouput Transcoding for Video Streaming", in Proc. International Workshop on Multimedia Signal Processing (MMSP), Montreal, Canada, January 2017.

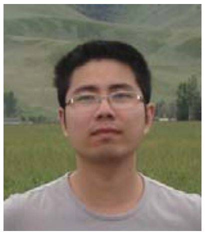
[51] G. Bjontegaard, "Calculation of Average PSNR differences Between RD Curves (VCEG-M33)," in VCEG Meeting (ITU-T SG16 Q.6), Austin, Texas, USA, Apr. 2-4, 2001.

Fanyi Duanmu received the B.S. degree from Beijing Institute of Technology (BIT), China in 2009, and the M.S. and the Ph.D degrees in Electrical & Computer Engineering from New York University, Tandon School of Engineering, Brooklyn, NY, in 2011 and 2018, respectively.

He is currently with Apple Inc. His current research interests include video compression and delivery, screen content coding, 360 degree video processing and streaming, computer vision and machine learning.

Zhan Ma received the B.S. and M.S. from Huazhong University of Science & Technology (HUST), Wuhan, China, in 2004 and 2006 respectively, and the Ph.D. degree from Polytechnic School of Engineering, New York University (formerly Polytechnic University), Brooklyn, New York, in 2011. He is now on the faculty of Electronic Science & Engineering School, Nanjing University, Jiangsu, China.

From 2011 to 2014, he has been with Samsung Research America, Dallas TX, and FutureWei Technologies, Inc., Santa Clara, CA, respectively. His current research focuses on the deep learning based video processing and communication, gigapixel streaming, computational vision models, and multi-spectral signal compression. He has received the 2018 ACM SIGCOMM Student Research Competition Finalist, and also the 2018 Pacific-Rim Conference on Multimedia (PCM) Best Paper Finalist.



Meng Xu received the B.S. degree from Nanjing University, China, in 2006, and the M.S. and the Ph.D. degrees in Electrical Engineering from Polytechnic School of Engineering, New York University, in 2009 and 2014, respectively.

He was with FutureWei Technologies, Santa Clara, CA, Real Communications, Inc., San Jose, CA, and Ubilinx Technology, Inc., San Jose, CA, from 2014 to 2018. He is currently with Tencent America, Palo Alto, CA. His research interests include video compression techniques and video codec design.

Yao Wang received the B.S. and M.S. in Electronic Engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in Electrical and Computer Engineering from University of California at Santa Barbara in 1990. Since 1990, she has been on the faculty of Electrical and Computer Engineering, Tandon School of Engineering of New York University (formerly Polytechnic University, Brooklyn, NY).

Her current research areas include video communications, multimedia signal processing, and medical imaging. She is the leading author of a textbook titled "Video Processing and Communications", and has published over 250 papers in journals and conference proceedings. She has served as an Associate Editor for IEEE Transactions on Multimedia and IEEE Transactions on Circuits and Systems for Video Technology. She received New York City Mayor's Award for Excellence in Science and Technology in the Young Investigator Category in year 2000. She was elected Fellow of the IEEE in 2004 for contributions to video processing and communications. She is also a co-winner of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2004, and a co-winner of the IEEE Communications Society Multimedia Communication Technical Committee Best Paper Award in 2011. She was invited as a keynote speaker at the 2010 International Packet Video Workshop and the 2018 Picture Coding Symposium.