SCREEN CONTENT COMPRESSION: A BRAINSTORMING REPORT

Fanyi Duanmu

Video Lab, New York University, Brooklyn, NY 11201, USA

Abstract: In this technical report, some brainstorming ideas are presented to improve screen content coding (SCC) efficiency. Firstly, a Layer-based Screen Content Coding (L-SCC) scheme is proposed, which decomposes the screen contents into multiple layers of sub-regions and compresses each sub-region respectively. Secondly, a Decoder-side Motion/Block Vector Derivation (DMBVD) framework is proposed, which derives additional motion vector candidates at both encoder and decoder, in order to reduce motion vector difference (MVD) signalling. Thirdly, a Block/Motion Vector Relay (BMVR) scheme is proposed to relay in between intra block copy (IBC) block vector (BV) and/or inter-frame motion vector (MV), in order to better predict the motion and reduce MVD signaling. Fourthly, a Pattern-based Palette Mode (P-PLT) framework is proposed, which is intrinsically a mixture of intra block copy mode and palette (PLT) mode, to particularly improve text region compression.

1. Introduction

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

To develop more efficient screen content compression solutions, the ISO/IEC Moving Picture Expert Group and the ITU-T Video Coding Experts Group, also known as "Joint Collaborative Team on Video Coding" (JCTVC), has standardized the High Efficiency Video Coding (HEVC) [1] SCC extension [2] in 2016. The official JCTVC Screen Content Model software (SCM) [3] is reported to provide >50% BD-Rate saving over the HEVC Range Extension (RExt) [2] for computer-generated contents. Four major coding tools were 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.

The first half of my Ph.D. dissertation [15] focuses on optimizing screen content encoding and transcoding by reducing system complexity, leveraging machine learning and statistical studies. In our previous work [10], a neural network (NN) based coding unit (CU) fast partition decision framework is proposed, using CU low-level features (such as sub-CU luminance consistency, CU variance, CU color diversity, CU gradient kurtosis, etc.) as NN inputs to regress a block partition soft-decision. Based on the confidence of the soft-decision, the CU is either directly partitioned or fast terminated. Compared with HEVC Screen Content Model software (SCM), a 37% complexity reduction is achieved with only 3% BD-Rate loss for Intra-frame coding. In our previous work [11], a refined decision tree (DT) based framework is proposed to make fast CU decisions for both mode and partition and achieves a higher complexity reduction (40%) with only 1.46% BD-Rate penalty. In our previous work [12], the first HEVC-SCC fast transcoding framework in the world is proposed for bandwidth reduction application. Leveraging CU statistical features to make fast mode and partition mapping decisions, the proposed framework achieves 48% re-encoding complexity reductions with 2.1% negligible BD-Rate loss. In our previous works [13] and [14], the first SCC-HEVC transcoding solution in the world is presented based on statistical mode mapping techniques. The proposed system demonstrates a remarkable SCV transcoding speedup (i.e., >80% complexity reduction for low-delay encoding configuration) and supports HEVC-SCC bitstream decoding compatibility over the legacy HEVC hardware. The proposed framework can be easily integrated into cloud infrastructure to support single-input-multiple-output (SIMO) transcoding for adaptive streaming applications, such as cloud gaming, remote desktop interfacing, etc.

Collaborated with and inspired by my Ph.D. adviser Prof. Yao Wang, my industrial collaborators Dr. Haoping Yu, Dr. Zhan Ma, Dr. Xiaozhong Xu, Dr. Yuwen He, Dr. Yan Ye, Dr. Xiaoyu Xiu, Dr. Meng Xu and many JCTVC and JVET researchers, I have been continuously brainstorming new ideas to further improve screen content compression efficiency. In this report, some SCC brainstorming ideas are presented (unfortunately I did not get a chance to implement and validate each of them but I think it worth archiving for future investigation and extension). The sequel of the report is structured as follows. Section 2 briefly reviews SCM coding structure and new coding tools. Section 3 to 6 present the novel coding solutions, including Layer-based Screen Content Coding (L-SCC), Decoder-side Motion/Block Vector Derivation (DMBVD), Block/Motion Vector Relay (BMVR) and Pattern-based Palette Mode (P-PLT), respectively. Finally, this report concludes in Section 7.

2. HEVC Screen Content Model (SCM): A Quick Review

HEVC Screen Content Model (SCM) is the JCTVC official test model software for SCC extension. This software is developed based on HEVC-RExt and supports YUV-4:4:4, YUV-4:2:0 and RGB-4:4:4 sampling formats. Beyond HEVC, new coding modes (e.g.: Intra Block Copy, Palette Mode, etc.) are introduced to improve the coding efficiency.

SCM shares the same recursive quad-tree block partitioning scheme as HEVC, which enables the use of Coding Units (CU), Prediction Units (PU) and Transform Units (TU) to adapt to diverse picture contents. CU is the basic unit for mode decision and is always in square shape. The Coding Tree Unit (CTU) is the largest CU (64x64 pixels). At the encoder, pictures are divided into non-overlapping CTUs and each CTU can be further divided into four equal-sized smaller CUs recursively, until the maximum hierarchical depth is reached. At each CU depth, to determine the optimal encoding parameters (e.g., partition decision, mode decision, etc.), an exhaustive search method is currently employed by comparing RD costs using different coding modes and comparing the minimum RD cost at the current CU depth against the sum of RD costs from its sub-CUs (each using best mode and partition). Beyond conventional HEVC modes, four novel encoding tools are introduced into SCM, as summarized below.

Intra Block Copy (IBC) [4] [5] is an Intra-frame version of the motion estimation and compensation scheme. To compress the current PU, the encoder will search over the previously-coded areas (either in restricted neighborhood or globally) in the same frame to 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 (PLT) [6] compresses the current CU as a combination of a color table and the corresponding index map. Color table stores the representative color "triplets" of RGB or YUV. Accordingly, 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. The index map coding consumes the majority of the bits. During JCTVC standardization, many index map coding solutions have been proposed and discussed, such as Run-based encoding, Line-based encoding, Index Map Recursive Splitting [17] [18] [19] encoding, 2D Index Map encoding [20], etc. To trade off the encoding/decoding complexity and coding efficiency, the run-length based encoding (either "copy left" or "copy above") in traversal scan is finally adopted for PLT mode index map coding.

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

Adaptive Motion Compensation Precision (AMCP) [8] [9] analyzes Inter-frame signal characteristics and classifies the current frame as 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.

3. Layer-based Screen Content Coding (L-SCC)

Screen contents have intrinsic layer structures among texts, graphics, terminals, windows and natural images. Each layer is typically homogeneous in signal representation. For example, natural image region is usually smooth and can be efficiently coded using intra modes for intra-frame coding. For inter-frame coding, natural image regions usually require sub-pixel motion estimation (ME) precision, whereas for text or graphics regions, screen content modes (e.g., IBC or PLT) are more efficient and adaptive color transform into YCoCg color space is often chosen. Besides, many screen content videos contain a large proportion of static regions and frames, as illustrated in Figure 1. Therefore, frame or region level signaling is more efficient than block-level signaling.

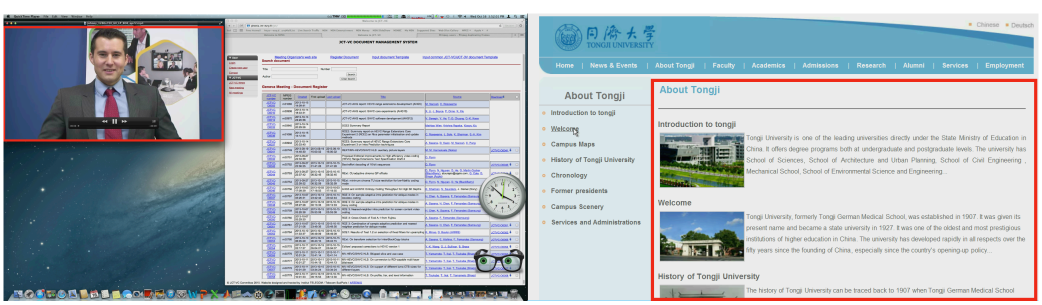

Figure 1 Sample Screen Content Frame (red bounding box enclosing non-static regions)

The signaling can be designed as “top-to-bottom”. For example, a single SPS bit ***IsStatic*** can be used to indicate whether the current frame is static or dynamic. For each dynamic frame, number of dynamic regions (e.g., in rectangular shape) ***NumOfDynamicRegions*** is signaled (e.g., using range encoding). For each dynamic region, geometry information is signaled to reflect the bounding box (e.g., ***x***, ***y***, ***width***, ***height***) information. Finally, over each dynamic region, standard coding solutions (e.g., HEVC, HEVC-SCC, AV1, etc.) can be applied.

L-SCC directly supports adaptive bitrate-allocation and saliency encoding. For example, different frame elements can be organized into a layer-representation. The background layer can be maintained as a long-term reference frame. The floating consoles, windows, etc., can be maintained as long-term reference patch for future reappearances. Saliency layers (such as the console enclosing the mouse, the foremost command window, etc.) can be encoded with more bits. Besides, special-effect signaling (e.g., alpha channel, saturation, watermark, etc.) over sub-regions can be easily and independently processed using additional meta data.

4. Decoder-Side Motion/Block Vector Derivation (DMBVD)

In HEVC, neighboring blocks can provide motion reference for compressing the current CU. In Merge or Skip mode, the motion vector is directly inherited from the merge candidates. In AMVP [16] mode, the candidate motion vector is used as initial search point and then progressively refined within a local search window using a diamond or square search pattern and the updated best vector becomes the new search center until convergence. The derived motion vector difference (MVD) is signaled in the bitstream, which is mostly smaller than the spatial offset between the current block and the best matching block and therefore improves the coding efficiency.

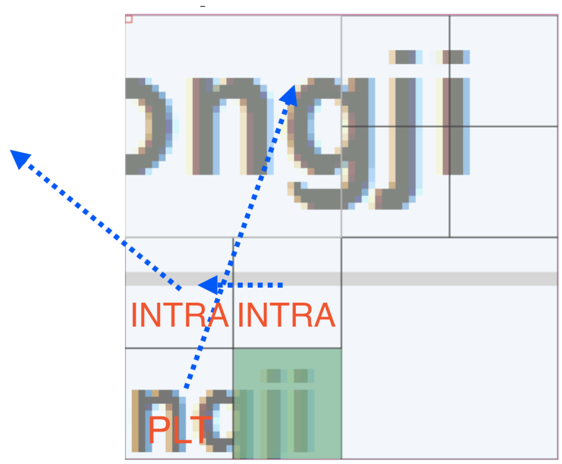

Figure 2 Decoder-Side Block/Motion Vector Derivation (green block indicating current CU to be encoded. Neighboring Intra or PLT block can provide additional block vector or motion vector)

For screen content coding, due to the repetitive signal characteristics, Decoder-Side Motion/Block Vector Derivation (DMBVD) can be applied, as illustrated in Figure 2. At both encoder and decoder, motion estimation can be applied over Intra block and PLT block to derive “extra” block vector (for IBC case) or motion vector (for Inter case). The derived block vector or motion vector may potentially increase the motion estimation accuracy and reduce the MVD or block vector difference (BVD) to be signalled and therefore improve the coding efficiency. In this example, the current block (marked in green) to be encoded/decoded can examine the derived MV or BV from neighboring Intra or PLT blocks. In this particular example, the current block can fully reuse the BV from the left PLT block without sending MVD or residual (i.e., encoded as an IBC-Skip block).

5. Block/Motion Vector Relay (BMVR)

As presented in [13] and [14], BV-MV relay technique has been applied for SCC transcoding applications, in order to efficiently map an IBC block into a corresponding Inter block, as illustrated in Figure 3.

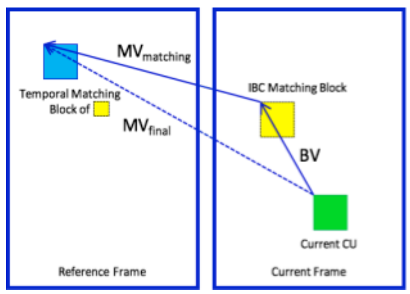

Figure 3 Block/Motion Vector Relay for IBC-Inter Transcoding

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.

In this report, an extended Block/Motion Vector Relay (BMVR) is proposed for encoding applications. In BMVR, beyond the conventional merge and AMVP candidates from Inter mode and IBC mode, extended motion vectors or block vectors can be derived by relaying in between block vector(s) and motion vector(s), as illustrated in Figure 4.

There are totally four scenarios considered, as summarized below:

- Scenario 1: BV relay, i.e., the current block B0’s neighbor is IBC Block B1 with block vector v1. B1’s matching block is B2. B2 is inside another IBC-coded block R1 with block vector v2 pointing to region R2. The relayed BV = v1(bv) + v2(bv).
- Scenario 2: MV relay, i.e., the current block B0’s neighbor is Inter Block B1 with motion vector v1. B1’s matching block is B2. B2 is inside another Inter block R1 with motion vector v2 pointing to region R2. The relayed MV = v1(mv) + v2(mv). In this case, B0 and R2 reside in different frames.
- Scenario 3: BV-MV relay, i.e., the current block B0’s neighbor is IBC Block B1 with block vector v1. B1’s matching block is B2. B2 is inside another Inter block R1 with motion vector v2 pointing to region R2 in a different reference frame. The relayed MV = v1(bv) + v2(mv). In this case, B0 and R2 reside in different frames.
- Scenario 4: MV-BV relay, i.e., the current block B0’s neighbor is Inter Block B1 in a reference frame ***F*** with motion vector v1. B1’s matching block is B2. B2 is

inside another IBC block R1 with block vector v2 pointing to region R2 in ***F***. The relayed MV = v1(bv) + v2(mv). In this case, B0 and R2 reside in different frames.

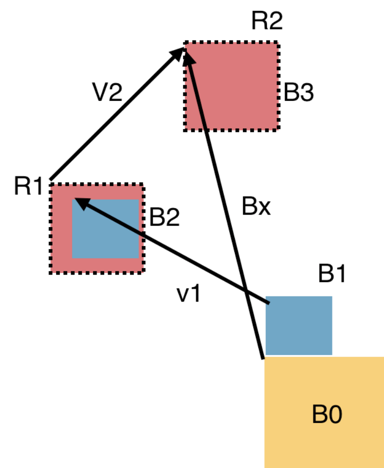

Figure 4 Block/Motion Vector Relay for Screen Content Encoding (B0: current block; B1: neighboring block; B2: B1's matching block; R1: an Inter or IBC block containing B2; R2: Final Reference block Location. v1 and v2 are BV or MV.)

6. Pattern-Based Palette Mode (P-PLT)

Screen contents contain repetitive patterns (e.g., text, graph, curve, icon). These patterns can be stored in the cache for predictive coding. In Pattern-based Palette Mode (P-PLT), the encoder and decoder dynamically maintain the same pattern lookup table storing the historical or pre-defined patterns encoded/decoded, as illustrated in Figure 5.

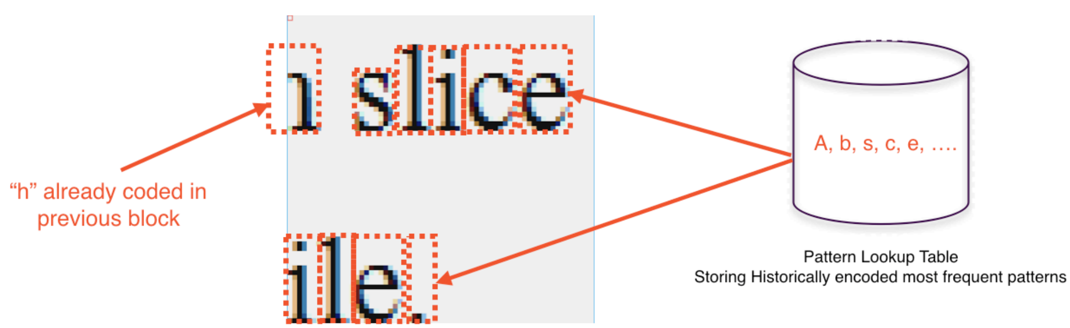

Figure 5 Block/Motion Vector Relay for Screen Content Encoding

After encoding/decoding a block, pattern recognition (e.g., foreground separation, optical character recognition) techniques are applied over encoded/decoded block to extract pattern candidates (e.g., the bounded text block in Figure 5). For a new block to be encoded, if the same pattern (e.g., the letter "e" in the pattern lookup table) is found, the encoder can directly signal the spatial offset (e.g., in x and y or scan index) and the pattern lookup index in the table (to copy from) and afterward mark the area (covered by the pattern block) as "coded". If no previous pattern is found (e.g., a new pattern), conventional IBC or PLT mode are used for compression.

The "coded" areas behave like "holes" in the coding unit block and can be inpainted based on block and neighboring pixels. The inpainted block contains less foreground elements (e.g., fewer letters) and can be compressed more efficiently (because PLT mode index map scanning run-length gets increased and IBC mode can find the matching block more efficiently over a sub-region than the entire region).

Please note that the pattern lookup and pattern matching can extend beyond block boundary, for example, the partial letter "h" from the left block in Figure 5, which is already coded and marked during the encoding of the block on the left.

The lookup table can be customized based on applications and memory constraints. For example, in remote education and online document sharing applications, the alphabets (e.g., Latin letters in different colors and fonts) can be pre-defined at both encoder and decoder as default lookup patterns for efficient text coding. With memory constraints, the lookup table can be designed to sort and archive only the most frequent patterns that appeared historically (e.g., up to N).

7. Conclusion

In this report, some of my brainstorming ideas (beyond my Ph.D. dissertation topic) for improving screen content compression have been presented. These tentative ideas are based on existing IBC and PLT mode designs but further extended leveraging computer vision techniques. Some of the ideas (I think) are potential (though requiring tunings) for next generation of screen content coding (e.g., VVC-SCC, AV2-SCC).

Acknowledgement

If you would like to further pursue, utilize or optimize some the ideas presented in this report, feel free to contact me via fanyi.duanmu@nyu.edu and I may be able to share some initial experimental results and/or code implementation. Please cite this report and our relevant SCC contributions from our lab website or my personal website:

https://wp.nyu.edu/videolab/publications/journals-and-conference-papers/

https://sites.google.com/site/duanmufanyi/publications

References

[1] 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.

[2] 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.

[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 CU partition decision using machine learning for screen content compression", in IEEE International Conference on Image Processing (ICIP), pp. 4972 - 4976, 2015.
[11] 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.
[12] 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. of IEEE International Conference of Image Processing (ICIP), pp. 4205-4209, Phoenix, Arizona, USA, 2016.
[13] F. Duanmu, M. Xu, Y. Wang, and Z. Ma, "HEVC-Compliant Screen Content Transcoding Based on Mode Mapping and Fast Termination," in Proc. of IEEE Visual Communications and Image Processing Conference (VCIP), pp. 1-4, St. Petersburg, Florida, USA, 2017.
[14] F. Duanmu, Z. Ma, M. Xu, and Y. Wang, "An HEVC-Compliant Fast Screen Content Transcoding Framework Based on Mode Mapping," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2018.
[15] F. Duanmu, "Fast Screen Content Coding and Two Tier 360 Degree Video Streaming", Ph.D. dissertation, New York University, Tandon School of Engineering, 2018.
[16] J.-L. Lin, Y.-P. Tsai, Y.-W. Huang, and S. Lei, "Improved Advanced Motion Vector Prediction", Doc. JCTVC-D125, January 2011.
[17] M. Xu, W. Wang, F. Duanmu, H. Yu, "Non-CE1: Context model unification for palette run type flags in HEVC SCC", Doc. JCTVC-U0148, June 2015.
[18] M. Xu, W. Wang, F. Duanmu, H. Yu, "Non-CE1: Index Map Splitting (IMS) Mode for Palette Coding in HEVC SCC," Doc. JCTVC-U0149, June 2015.
[19] H. Yu, M. Xu, W. Wang, F. Duanmu, S. Minaee, "Advanced coding techniques for high efficiency video coding (HEVC) screen content coding (SCC) extensions", US Patent: 10148977B2, 2018.
[20] Y. Xu, W. Huang, W. Wang, F. Duanmu, and Z. Ma, "2-D Index Map Coding for HEVC Screen Content Compression", in Proc. of IEEE Data Compression Conference (DCC), pp. 263-272, Snowbird, Utah, USA, 2015.