Residual Coding for Transform Skip Mode in Versatile Video Coding

T. Nguyen*, B. Bross*, H. Schwarz*†, D. Marpe*, and T. Wiegand*‡

*Fraunhofer HHI
Video Coding & Analytics
Berlin, Germany

†Freie Universität Berlin
Inst. of Computer Science
Berlin, Germany

‡Technische Universität Berlin
Media Technology
Berlin, Germany

firstname.lastname@hhi.fraunhofer.de

Abstract

The support for screen content coding has received more attention with the latest development in video compression, the upcoming Versatile Video Coding (VVC) standard. Among the dedicated screen content coding tools, the transform skip mode (TSM) represents a promising approach for improving the coding efficiency at a low impact on implementation complexity. In this work, we present a dedicated residual coding for transform blocks coded in TSM. Due to the lack of the energy compaction of the transform, the quantization indexes for blocks coded in TSM have different statistical properties, which can be exploited in the entropy coding. Our coding experiments with screen content sequences yielded bit-rate savings of 3.9% for intra-only coding and 2.8% for typical random access configurations.

Introduction

The upcoming Versatile Video Coding (VVC) standard [1], the successor of High Efficiency Video Coding (HEVC) [2], supports the coding of screen content in an extended manner. Being developed by the Joint Video Experts Team (JVET), the expert group founded jointly by ITU-T VCEG and ISO/IEC MPEG, VVC includes dedicated screen content coding tools that are not available in version 1 of HEVC. One of the screen content coding tools, the transform skip mode (TSM) [3], is a technique that was already supported in HEVC version 1, but only for 4×4 transform blocks. Noteworthy is the fact that the JVET common test conditions (CTC) [4] specify the usage of TSM for all sequences of the JVET test set. In contrast to that, the other dedicated screen content coding tools, such as intra block copy (IBC) [5] or palette mode coding (PLT) [6], are enabled only for specific screen content video sequences of the JVET test set. An encoder conforming to the Versatile Working Draft 4 (WD4) can use the TSM for transform block sizes up to 32×32, but only for the luma component, which is similar to the specification for the HEVC Range Extensions (RExt) [7]. While the HEVC RExt modifies the regular residual coding (RRC) for transform skip blocks by using a single dedicated context model during the significance map coding and a rotation of the residual signal, the WD4 does not specify a new residual coding path for the transform skip mode. Note that due to the bypassing of the transform in the TSM, the blocks themselves consist of spatial residuals. Nevertheless, this paper uses the term transform block to denote blocks of quantization indexes (also called transform coefficient levels), regardless of whether they are obtained by transform and quantization or quantization only.

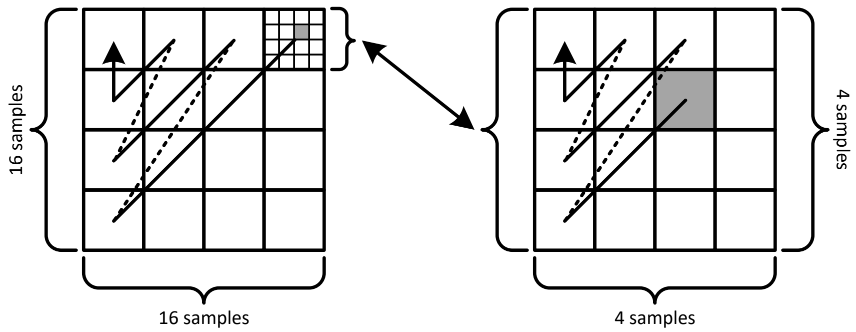

Figure 1: An example of the partitioning of a transform block into 4×4 sub-blocks and the corresponding processing order given by the reverse diagonal scan pattern. The gray shaded sample position represents the last significant scan position.

Instead of modifying the regular residual coding (RRC) as in HEVC RExt, we present a dedicated residual coding design for the TSM, which is also referred to as transform skip residual coding (TSRC). Derived from the RRC design, the TSRC scheme keeps important concepts of the RRC and changes only aspects that lead to an improvement in compression efficiency. Such an approach minimizes the implementation overhead, especially for hardware architectures, since the developers can reuse existing building blocks. The JVET adopted the initial TSRC design, as presented in this paper, into the 5-th Working Draft (WD5) for the VVC standard.

A detailed description of the TSRC is presented in the remainder of the paper. Section 1 briefly reviews the regular residual coding (RRC) specified in the current VVC draft. Section 2 describes the TSRC scheme that we propose for coding the quantization indexes of transform skip blocks. Experimental results are presented in Section 3. A conclusion is given in Section 4.

1 Residual Coding in VVC

The design of the regular residual coding (RRC) of VVC is the result of a development that considered several constraints, such as implementation complexity, throughput, and more, besides its primary goal, the improvement of compression efficiency. The basic approach has several commonalities with the residual coding of HEVC. Transform blocks larger than 4×4 are divided into disjunct 4×4 sub-blocks [8], which are processed using a reverse diagonal scan pattern. Figure 1 depicts an example for a 16×16 transform block, partitioned into 16 sub-blocks. The reverse diagonal scan pattern is used for processing the sub-blocks of a transform block as well as for processing the frequency positions within each sub-block.

At the beginning of a transform block, a syntax element s_{cbf} is coded. It is also referred to as coded block pattern and indicates whether the transform block contains any non-zero transform coefficient levels. If s_{cbf} is equal to 1, the position of the first non-zero level in scan order (also referred to as last significant scan position) is coded. Starting with this position, the transform coefficient levels are transmitted on the basis of sub-blocks as will be described in the following.

1.1 Last Significant Scan Position

Due to the energy compaction property of the transforms used in video coding, the quantization indexes at higher frequency positions tend to become zero. The higher the quantization step size, the more likely the last significant scan position is closer to the DC frequency position, i.e., the top-left corner of the transform block. Consequently, in VVC, the last significant scan position is coded as x- and y-coordinates, relative to the top-left corner of the transform block. For the example in Fig. 1, the last significant scan position is located at $(14, 1)$ relative to the top-left corner of the transform block. Thus, the bitstream includes the syntax elements $last_x = 14$ and $last_y = 1$ for specifying the last significant scan position.

1.2 Coded Sub-Block Flags

Similar to the s_{cbf} for the transform block, the syntax includes a s_{cbf} for each sub-block indicating the existence of any non-zero level inside the sub-block. However, two exceptions exist: One for the sub-block including the last significant scan position and another for the sub-block including the DC location. Since the last significant scan position already indicates that the corresponding level is non-zero, the s_{cbf} for this sub-block is inferred to be equal to 1. For the sub-block covering the DC location, it is highly likely that one of the lower frequency positions is significant, again due to the energy compaction property of transforms. Therefore, the parsing at the decoder side always processes the sub-block covering the DC location.

The s_{cbf} is the first syntax element that is transmitted for a sub-block. If it indicates that the sub-block contains any non-zero levels, the transform coefficient levels are coded as described in the following. For the last scan position inside a sub-block, the significance flag that indicates whether the level is non-zero (see below) is not transmitted if all preceding levels of the same sub-block are equal to 0, since in this case, it can be inferred that the level at the last position inside a sub-block is non-zero. Exceptions are the two cases desribed above.

1.3 Coding of Transform Coefficient Levels

Within each sub-block, the absolute values of the transform coefficient levels and, for absolute values greater than 0, the signs of the levels are coded in several loops over the scan positions of a sub-block. In the first loop, the binary syntax elements s_{sig}, s_{gt1}, s_{par}, and s_{gt3} are coded for the scan positions inside the sub-block, where s_{sig} indicates whether the absolute value is larger than 0, s_{gt1} indicates whether the absolute value is greater than 1, s_{par} indicates the parity of the absolute value, and s_{gt3} indicates whether the absolute value is greater than 3. The flags s_{gt1} and s_{par} are only coded if s_{sig} indicates that the corresponding level is not equal to 0. The greater-than-3-flag s_{gt3} is only coded if $s_{gt1} = 1$, i.e., the absolute value is greater than 1. Furthermore, the first loop may be terminated within an sub-block when the number of context-coded bins (CCB) exceeds a pre-defined threshold. Note that HEVC achieves a similar CCB limitation using a bit-plane wise transmission where up to 8 s_{gt1} and up to 1 s_{gt2} syntax elements are transmitted.

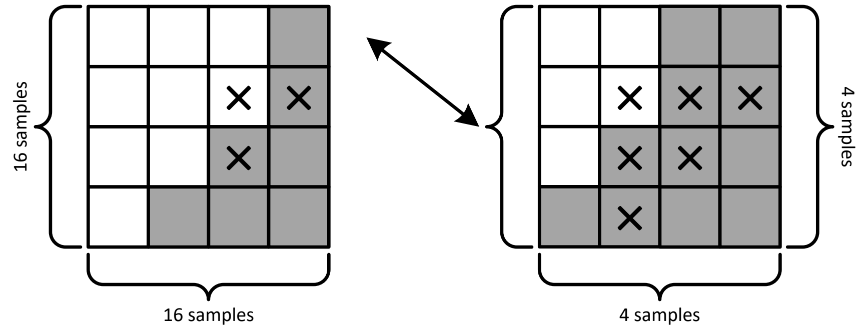

Figure 2: These figures illustrate the local template for context modelling using either neighboring sub-blocks or neighboring frequency positions.

The partially reconstructed absolute value $|\ell|^*$ for a transform coefficient level ℓ after the first pass is given by $|\ell|^* = s_{sig} + s_{gt1} + s_{par} + 2 \cdot s_{gt3}$. In the second loop, the remainders rem for the absolute levels are transmitted using a combination of Rice and Exp-Golomb coding as in HEVC. It is referred to as remainder because, for some scan positions, there exist partial information on the absolute value $|\ell|$ coded in the first loop (in which case $rem = (|\ell| - |\ell|^*)/2$ is transmitted), whereas the remainder represents the entire absolute level ($rem = |\ell|$) for scan positions not covered in the first pass. The Rice parameter derivation itself in VVC employs the same local template as for the significance flag s_{sig} (see below). After summing up the absolute values of all neighboring levels covered by the local template (right side of Fig. 2), a lookup table maps the final absolute sum to the Rice parameter. In the last scan pass, the signs for all non-zero levels are transmitted.

1.4 Context Modeling and Dependent Quantization

The main improvement of the residual coding in VVC relative to HEVC is the advanced context modeling, which exploits additional statistical dependencies between neighboring quantization indexes inside a transform block. The left side of Fig. 2 shows the template used for selecting the context for the sub-block s_{cbf} flag. The right side of Fig. 2 illustrates the template used for context model derivation for the flags s_{sig}, s_{gt1}, s_{par}, and s_{gt3}. It includes five already coded scan positions in the local neighborhood. As noted above, the same template is also used for deriving the Rice parameter for coding the remainder. Depending on the sum of absolute levels inside the template, the context model and the Rice parameter are selected. A more detailed description of the context modeling can be found in [9].

Since VVC supports trellis-coded quantization (TCQ) [10], the context model selection for the significance flag s_{sig} additionally depends on the current state of a finite state machine (FSM). Depending on the internal state of the FSM, the processing selects a different context model set, whereas the derivation of the context model index inside the context set remains the same. The state derivation depends on the parities of preceding absolute levels in scanning order. For avoiding a frequent switching between context-coded and bypass-coded bins, the VVC syntax includes the dedicated parity flag s_{par} in the first coding pass.

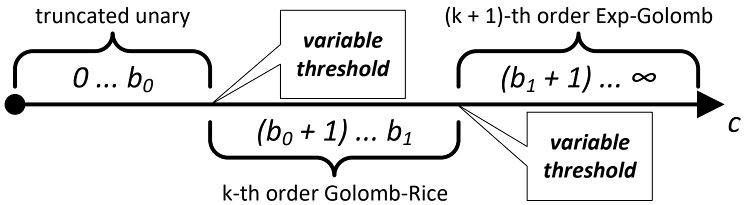

Figure 3: Binarization of absolute transform coefficient levels in both HEVC and VVC. The binarization represents a concatenation of three different variable-length codes.

1.5 Binarization

The binarization of transform coefficient levels in VVC is very similar to that of HEVC. Fig. 3 illustrates the binarization of transform coefficient levels in HEVC; it shows that the binarization process is a concatenation of three different variable-length codes: A truncated unary code, a Rice code, and an exponential Golomb code (Exp-Golomb). Note that the bounds defining the transitions among the different variable-length codes are variable. Since all bins of the truncated unary part are coded using adaptive context models, each of these bins represents a dedicated syntax element, such as s_{sig} and s_{gtX} with $X \in \{1, 2, 3, \ldots\}$. The first variable bound decreases when the CCB budget is exhausted, reflecting the transition into the bypass mode during encoding and decoding. A difference in the binarization of VVC compared to HEVC is that VVC includes a dedicated parity flag s_{par} for improving the entropy coding of quantization indexes for trellis-coded quantization.

2 Transform Skip Residual Coding

Modifying the residual coding for the transform skip mode can improve the compression efficiency, as has been shown by the range extensions (RExt) of HEVC. Specifically, HEVC RExt includes two simple changes for transform skip residual coding: A rotation of the residual signal and the usage of a single dedicated context model for coding the significance flags s_{sig}. To further improve the compression efficiency, we developed a new transform skip residual coding (TSRC). Beside exploiting similar effects as HEVC RExt, it includes additional aspects that improve the coding efficiency for quantization indexes obtained in transform skip mode. The developed TSRC reuses many aspects of the regular residual coding, so that decoder implementations can reuse many building blocks. Since our investigations showed that trellis-coded quantization does typically not improve the coding efficiency for transform skip residuals, it is disabled in transform skip mode.

2.1 Coding Order and Scanning

For TSRC, instead of transmitting the last significant scan position, the quantization indexes of all scan positions of a transform block are coded. It also means that the encoder transmits the s_{cbf} for all sub-blocks. An exception is the last sub-block in processing order: When all levels in all other sub-blocks are equal to zero, the s_{cbf} for the last subblock is inferred to be equal to 1. The second difference to the RRC is a reversion of the scan order. The diagonal scan is used in a forward manner (and not in reverse order as in the RRC). The reason is that the intra prediction itself becomes less efficient for sample positions that are further away from the employed reference samples. In other words, the local signal variance becomes larger with greater distance to the reference samples, resulting in larger residual values at the right-bottom corner of the transform block. When reversing the scanning order, the probability that a quantization index is non-zero increases in scan order, similar as in RRC. It should be noted that reversing the scanning order is very similar to a residual rotation, as used in HEVC RExt. Although the benefit for inter-predicted blocks is not significant, changing the direction of the scan pattern does not harm inter-predicted blocks.

2.2 Coding of Residual Levels

Similar to the RRC design, the coding of quantization indexes proceeds in several scan passes. The first pass includes context-coded syntax elements, which are the s_{sig}, s_{sign}, s_{gt1}, and s_{par} flags. Note that, in contrast to the RRC, the first pass does not include the s_{gt3} syntax element, but the sign flag s_{sign}. While the overall probability of s_{sign} remains roughly equal to 0.5, a data analysis showed that locally the signs are biased towards one direction. By using a single adaptive context model, this property is utilized for improving the coding efficiency. Moreover, the data analysis reveals that increasing the variable bound between the truncated unary code and the Rice code further improves the compression efficiency for screen content. Therefore, the second pass of TSRC includes the syntax elements s_{gt3}, s_{gt5}, s_{gt7}, and s_{gt9}. Finally, the remainder values are bypass-coded in a third pass, similar as in the RRC.

2.3 Context Modeling and Context Coded Bins Limit

Similar to the context modelling in RRC, the TSRC employs a local template, but with only two neighbors. Specifically, the processing evaluates the top and the left neighbor relative to the current scan position. For the significance flag s_{sig}, the context model index is given by the number of non-zero neighbors. For each of the remaining syntax elements coded in the first pass (s_{sign}, s_{gt1}, and s_{par}), a single dedicated context model is used. The syntax elements of the second pass (s_{gt3}, s_{gt5}, s_{gt7}, and s_{gt9}) use the same context model as the greater-than-one flag s_{gt1}.

In the RRC of VVC, the number of context-coded bins is limited to 1.75 bins per transform coefficient. As noted above, our data analysis reveals that residuals in transform skip mode require a higher number of context-coded bins than in the RRC case. However, transform skip blocks with a large number of non-zero quantization indexes do not occur very often, and it turned out that specifying a maximum number of 3 context-coded bins per residual sample retains most of the coding efficiency improvements. In the presented TSRC design, the encoder guarantees that this limit is not violated. This is achieved by not using the transform skip mode if its coding would results in a larger number of context-coded bins.

Table 1: The compression efficiency improvements, in terms of BD-rate, of the presented transform skip residual coding (TSRC) scheme without a CCB limit. The intra-block copy (IBC) mode is disabled in both the anchor and the TSRC configuration.

classes	**Y**	**Cb**	**Cr**	**enc. time**	**dec. time**
all intra (AI)					
F	-5.43%	-3.87%	-3.80%	102%	97%
TGM	-13.98%	-10.48%	-10.76%	101%	83%
F, TGM	**-9.70%**	**-7.17%**	**-7.28%**	**101%**	**95%**
A-C, E	**-0.17%**	**-0.11%**	**-0.07%**	**100%**	**99%**
random access (RA)					
F	-3.79%	-2.67%	-2.44%	100%	100%
TGM	-7.49%	-5.60%	-5.70%	99%	98%
F, TGM	**-5.64%**	**-4.14%**	**-4.07%**	**99%**	**99%**
A-C, E	**-0.05%**	**-0.03%**	**-0.06%**	**99%**	**101%**

3 Experimental Results

We implemented the presented TSRC into version 3 (VTM3) of the reference software for VVC. The coding efficiency was evaluated according to the JVET common test conditions (CTC) [4]. For each tested configuration, four bitstreams for the base QP values of 22, 27, 32, and 37 are generated. The differences in coding efficiency between two configurations are measured using the Bjøntegaard-Delta rate (BD-rate) metric [11]. Negative BD-rate values specify bit-rate savings for the same reconstruction quality and, thus, indicate improvements in compression efficiency.

3.1 Test Set

The JVET common test conditions specify different test sequences, which are grouped into classes. Each class contains video sequences with similar characteristics, such as a similar spatial resolution and a similar type of video content. Class F consists of four video sequences with screen content. The test set includes an additional class labeled as Text and Graphics with Motion (TGM), which contains video sequences with a similar type of content. Both class F and class TGM are used for testing the effectiveness of screen content coding tools in the VVC development [12].

According to the JVET common test conditions, the two screen content classes F and TGM and the class D (which includes low resolution video sequences) are not considered in calculating the overall average results. The reason is that the main envisioned application of VVC is the coding of high-resolution material with natural video content.

Table 2: The compression efficiency improvement, in terms of BD-rate, of the presented transform skip residual coding (TSRC) scheme with a CCB limit of 3 bin per sample. The intra-block copy (IBC) mode is disabled in both the anchor and the TSRC configuration.

classes	Y	Cb	Cr	enc. time	dec. time
all intra (AI)					
F	-5.07%	-3.56%	-3.39%	103%	96%
TGM	-12.95%	-9.59%	-9.83%	104%	92%
F, TGM	**-9.01%**	**-6.85%**	**-6.61%**	**103%**	**96%**
A-C, E	**-0.17%**	**-0.10%**	**-0.07%**	**104%**	**92%**
random access (RA)					
F	-3.30%	-2.17%	-2.07%	100%	99%
TGM	-6.78%	-5.50%	-5.06%	99%	97%
F, TGM	**-5.04%**	**-3.59%**	**-3.57%**	**100%**	**98%**
A-C, E	**-0.05%**	**-0.06%**	**-0.06%**	**99%**	**101%**

3.2 Experimental Setup

The JVET CTC describes different encoder configurations that reflect typical applications scenarios. We present results for the following two configuration: The all-intra (AI) and the random access (RA) configuration. In the all-intra configuration, each video picture is coded independently of all other video pictures. Hence, inter-picture prediction cannot be used in this setup. The random access configuration represents a typical setup used in applications like video streaming and broadcasting. The generated bitstreams provide random access in intervals of about 1.1 seconds. For the random access configuration a coding structure with hierarchical B pictures is used. Random access is enabled by inserting intra pictures in regular intervals and restricting the inter-picture prediction accordingly.

For the screen content sequences, i.e., the classes F and TGM, the CTC specifies the usage of IBC, which already results in a significant improvement in compression efficiency for screen content. For the remaining encoder parameters, the CTC specifies default values so that the outcome is a balanced trade-off between compression efficiency and encoding/decoding complexity.

3.3 TSRC and CCB Limit

IBC is an efficient but also complex coding tool for screen content. In order to show the interaction between TSRC and IBC, we first tested the proposed TSRC without IBC. That means, IBC has been disabled in both the reference and the test configuration. Table 1 summarizes the BD-rate values for the presented TSRC scheme without a CCB limit, i.e., up to 8 CCB per residual sample may be coded on average for each transform block. Notably, the improvement achieved by adding the TSRC is almost 10% on average for the screen content sequences; for camera captured content, an average bit-rate saving of 0.17% has been measured. Moreover, it should be noted

Table 3: The compression efficiency improvement, in terms of BD-rate, of the presented transform skip residual coding (TSRC) scheme with a CCB limit of 3 bin per sample. For this test, the intra-block copy (IBC) mode was enabled in both the anchor and the tested TSRC configuration.

classes	Y	Cb	Cr	enc. time	dec. time
all intra (AI)					
F	-2.85%	-1.44%	-1.39%	100%	98%
TGM	-4.88%	-2.68%	-2.80%	100%	95%
F, TGM	**-3.86%**	**-2.06%**	**-2.10%**	**100%**	**96%**
A-C, E	**-0.14%**	**0.01%**	**0.00%**	**100%**	**100%**
random access (RA)					
F	-1.89%	-1.16%	-0.92%	99%	98%
TGM	-3.79%	-2.33%	-2.35%	98%	99%
F, TGM	**-2.84%**	**-1.75%**	**-1.63%**	**99%**	**99%**
A-C, E	**-0.05%**	**-0.07%**	**0.08%**	**99%**	**101%**

that the encoding time is not increased relative to the anchor. The slightly decreased decoding time indicates either that the generated bit rates are lower than that of the anchor or the reconstruction is faster due to the skipping of the inverse transform.

Table 2 summarizes the BD-rate values for the case when the CCB limit is set equal to 3. The compression efficiency improvements are slightly lower, with a higher impact on the sequences of the TGM class. A CCB limit of 3 bin per residual sample seems to be a reasonable choice for balancing the coding efficiency and the worst-case number of context-coded bins.

3.4 TSRC with IBC enabled

When IBC is enabled, the obtained coding efficiency improvements are lower as shown in Table 3. The improvement for the screen content is, however, still 3.86% on average. The results also show that the combination of IBC and TSRC leads to further improvements in compression efficiency. Note that this is not always possible; for example, IBC and PLT cannot be used together for the same block.

4 Conclusion

In this paper, we presented a dedicated residual coding scheme for blocks coded in transform skip mode (TSM). By re-using the architecture of the regular residual coding and modifying the coding order and the context modeling, an encoder can achieve a significant improvement in compression efficiency for screen content. The compression efficiency improvements, in terms of BD-rate, are roughly 9% in an all-intra (AI) configuration, and about 5% in a typical random access (RA) configuration for the screen content sequences of the JVET test set. When intra-block copy (IBC) is enabled, the corresponding compression efficiency improvements are about 4% and 3%

for AI and RA, respectively, which also shows that both screen content coding tools, TSM and IBC, can be efficiently combined. Even though the new residual coding mode introduces some additional complexity, the overhead is very low, and the encoding and decoding times for software implementations are not affected. Compared to other screen content coding tools, the TSM with the proposed residual coding provides a promising trade-off between coding efficiency and complexity.

References

[1] G. Sullivan, "VVC – The Next-Generation Video Standard of the Joint Video Experts Team," Online: `http://mile-high.video/files/mhv2018/pdf/day1/1_05_Sullivan.pdf`, July 2018.

[2] G. Sullivan, J. R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec 2012.

[3] M. Mrak and J. Xu, "Improving Screen Content Coding in HEVC by Transform Skipping," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug 2012, pp. 1209–1213.

[4] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "JVET Common Test Conditions and Software Reference Configurations for SDR Video," in *JVET-L1010*, Macau, MO, Oct 2018.

[5] X. Xu, S. Liu, T. Chuang, Y. Huang, S. Lei, K. Rapaka, C. Pang, V. Seregin, Y. Wang, and M. Karczewicz, "Intra Block Copy in HEVC Screen Content Coding Extensions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, Dec 2016.

[6] W. Pu, M. Karczewicz, R. Joshi, V. Seregin, F. Zou, J. Sole, Y. Sun, T. Chuang, P. Lai, S. Liu, S. Hsiang, J. Ye, and Y. Huang, "Palette Mode Coding in HEVC Screen Content Coding Extension," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 420–432, Dec 2016.

[7] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, "Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 4–19, Jan 2016.

[8] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Transform Coding Techniques in HEVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 978–989, Dec 2013.

[9] H. Schwarz, T. Nguyen, D. Marpe, T. Wiegand, M. Karczewicz, M. Coban, and J. Dong, "Improved Quantization and Transform Coefficient Coding for the Emerging Versatile Video Coding (VVC) Standard," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 1183–1187.

[10] H. Schwarz, T. Nguyen, D. Marpe, and T. Wiegand, "Hybrid Video Coding with Trellis-Coded Quantization," in *2019 Data Compression Conference (DCC)*, March 2019, pp. 182–191.

[11] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD Curves," in *VCEG-M33*, Austin TX, USA, Apr 2001.

[12] X. Xu, Y.-C. Chao, Y.-C. Sun, and J. Xu, "Description of Core Experiment 8 (CE8): Screen Content Coding Tools," in *JVET-L1028*, Macau, MO, Oct 2018.