Finalization of VVenC's Screen Content Detector and Two-Pass Rate Control Using Pre-Filtering Statistics

Christian R. Helmrich, Anastasia Henkel, Tobias Hinz, Adam Wieckowski, Benjamin Bross, and Detlev Marpe

Video Communication and Applications Group, Fraunhofer Heinrich Hertz Institute (HHI), Einsteinufer 37, 10587 Berlin, Germany

ABSTRACT

For improved performance, practical video encoders integrate algorithms for screen content detection and rate control. This paper outlines recently implemented optimizations to both the screen content classifier (SCC) and two-pass rate control (RC) of VVenC, an open Versatile Video Coding (VVC) compliant encoder. The improvements, confirmed by evaluation experiments in random-access configurations using an extended test set of videos, are mainly achieved by leveraging motion error statistics acquired during motion compensated temporal prefiltering (MCTPF), carried out in VVenC's pre-analysis stage. All three aspects – pre-analysis stage, SCC, and RC – are revisited herein, and the exploitation of MCTPF data is described.

***Index Terms*—**QoE, rate control, SCC, video coding, VoD, VVC

1. INTRODUCTION

With the rapidly increased share of compressed video content in global IP traffic especially since the COVID pandemic, via video-on-demand (VoD) streaming, social media, or teleconferencing apps, efficient video coding solutions have attracted further interest. To address this need, an open encoder generating Versatile Video Coding (VVC) compliant bit-streams [1], named VVenC [2], was published in 2020 while the VVC specification [3] was being finalized, in order to serve as an early but realistic demonstration of VVC's capabilities. Since then, VVenC has been equipped with a screen content classification (SCC) method, optimizing VVenC's encoding process for fast and efficient operation on screen sharing and online gaming input, as well as a two-pass rate control (RC) algorithm [4, 6], allowing the user to specify a target rate R_{target}, in bps, instead of a quantization parameter QP_{base}, in the range $0 \ldots QP_{\max}$. The automatic encoding optimization for screen vs. camera captured video signals and more intuitive control via R_{target} makes VVenC, or any other encoder, much more user friendly. However, though operating satisfactorily, these SCC and RC components of VVenC were found to perform suboptimally on a specific class of video material, as described in the following.

1.1. SCC and RC Shortcomings

Based on the SCC decision – *no*, *weak*, or *strong* screen-like content – made for each frame f in a video during pre-analysis, VVenC adjusts some rate-distortion (R-D) speed-ups, such as block partitioning, motion estimation, and merging of motion prediction candidates. Moreover, some coding tools are being activated or disabled depending on the frame-wise SCC type:

- Block based differential pulse code modulation (BDPCM) and residual transform skip (TS) are used with *weak* SCC,
- Intra-picture block copy (IBC) and fast motion estimation with diamond-region search are enabled with *strong* SCC, in addition to the BDPCM and residual TS encoding tools,
- luma mapping and chroma scaling (LMCS), MCTPF, and SCC specific fast merging are disabled with *strong* SCC.

Details on these coding tools and optimizations are published in [1, 5]. Deactivating the MCTPF on computer generated input greatly speeds up the encoding process, but it also makes VVenC's efficiency quite sensitive to suboptimal *strong* SCC decisions. In fact, the authors observed that some camera captured movie scenes with homogenous picture areas, showing little variation in texture and luminance but notable levels of film grain or camera sensor noise, may be classified as *strong* SCC. These scenes are then encoded without MCTPF and, as a result, the peak signal-to-noise ratio (PSNR) drops by a few percent at the same R_{target}, relative to encoding using MCTPF.

VVenC's two-pass RC design can operate either in *offline* sequence-wise mode for use in file based workflows [4] or in *on-the-fly* GOP-wise mode for use in stream based scenarios where the entire video input is not available a priori [6]. GOPs are groups of consecutive (in *display order*) pictures enabling efficient temporally hierarchical coding especially in random access (RA) configurations. Since the input pictures in a GOP are reordered before encoding (i. e., in *coding order*) based on their individual assignment to temporal level $l_f \geq 0$, GOPs may be considered the smallest reasonable video duration for two-pass RC methods such as the lookahead based one in VVenC.

Although this two-pass RC approach is already somewhat *noise aware* and was proven to be very efficient [4, 6–8], two issues regarding its performance have recently been observed:

- The particular realization of the noise level dependent QP limiter described in [6] results in R_{target} not being reached on some video sequences due to inefficient bit allocation,
- compared with fixed-QP encodings, VVenC's RC exhibits a somewhat inferior performance (in terms of BD-rate [9]) especially on input with visible film grain or sensor noise.

In other words, both RC issues as well as the SCC suboptimality occur with content having above-average levels of noise. An assessment of the "noisiness" of each scene – or at least of each GOP – in a video would be desirable to improve the RC and SCC models. At the same time, any additional collection of picture statistics should consume only little computational overhead, in order not to slow down VVenC's *fast*(*er*) presets.

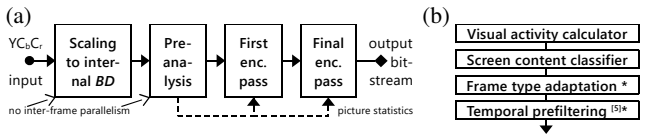

Figure 1. Processing stages in VVenC. (a) Location of pre-analysis stage in signal path, (b) components of pre-analysis stage in order of input signal processing, * possible restriction to low temporal levels.

Fortunately, with the availability of MCTPF functionality as part of VVenC's pre-analysis stage (operated after SCC, as noted), a method collecting variation statistics between input pictures in a motion compensated fashion – i. e., some kind of noisiness information – already exists, and this data is readily accessible in every f without requiring additional complexity.

1.2. Outline of This Paper

The remainder of this paper is organized as follows. The pre-analysis stage as currently integrated in VVenC is introduced in Sec. 2, along with brief descriptions of SCC and MCTPF. Sec. 3 presents three improvements to the SCC, one of which makes use of minimal motion compensated picture difference statistics collected within the MCTPF process. Sec. 4 outlines three enhancements to the two-pass RC scheme, one of which also reuses the MCTPF motion error information. The results of individual evaluation experiments conducted to assess the effect of the SCC and RC modifications are summarized and discussed in Sec. 5, and Sec. 6 concludes the publication.

2. VVENC'S PRE-ANALYSIS STAGE

The location of the pre-analysis stage in VVenC's encoding process is shown in Fig. 1(a) for the case of two-pass RC operation. As illustrated, pre-analysis is carried out directly on the bit-depth normalized input pictures P_f in each frame f, prior to any R-D optimization in the first or second RC encoding pass. Some of the pre-analysis steps, listed in Fig. 1(b), are executed *intra-frame parallelized*, i. e., multithreaded in separate block lines. The purpose of the four pre-analysis steps is as follows:

1. Calculation of spatial and temporal visual activity per f for use with XPSNR [10] based perceptual QP adaptation [11] in each coding block (CTU), scene cut detection [4], and frame type adaptation [6]. The latter two algorithms make use of a picture-wise average luma-channel visual activity
$$\hat{a}_f = \max\left(a_{\min}^2;\ \left(\tfrac{1}{4WH}\textstyle\sum_{[x,y]\in P_f}|h_\mathrm{s}[x,y]| + 2|h_\mathrm{t}[x,y]|\right)^2\right) \quad (1)$$
with W and H being the picture width and height, respectively, $a_{\min}=2^{BD-6}$ representing a lower activity limit where BD is the internal image bit-depth as in Fig. 1(a), and h_s, h_t denoting spatial and temporal high-pass output, as in [10].
2. SCC by segmenting the luma channel of each P_f into 8×8 subblocks Y, deriving the mean $\mu_{0..3}$ and standard deviation $\sigma_{0..3}$ of the samples of each of the 4×4 quadrants 0..3 of Y, rounding all $\sigma_{0..3}$ values to integer, and counting, across all Y associated with each of the four quadrants $Q_{0..3}$ of P_f, the number of cases where within a $Y{\in}Q$ vertically or horizontally neighboring values of σ (i. e., σ_0 and σ_2, σ_1 and σ_3, σ_0 and σ_1, or σ_2 and σ_3) are equal. Details on how the equality statistics are used in the SCC decision are given in Sec. 3.
3. Frame type adaptation (FTA), i.e., a signal adaptive choice between frame types 'I' (Intra-only) and 'non-I' (temporal motion prediction allowed) in each key frame during hierarchical RA encoding. In other words, the goal of FTA is to convert traditional 'non-I'-type key frames, identifiable by their temporal level $l_f = 0$, into 'I' frames in case these are located at, or directly after, a scene change, in order to improve the coding efficiency [6]. The detection is based on comparing frame visual activities $\hat{a}_f$ derived from (1) in successive key frames. Details shall be omitted for brevity.
4. MCTPF, consisting of a motion estimation (ME) *analysis* and a bilateral *filtering* part [12, 13], intends to attenuate random picture components in a motion aware fashion, in order to further improve the coding efficiency (temporally uncorrelated input exhibits high entropy and is, therefore, hard to compress using hybrid codecs like VVC). Both the ME and the filtering are operated block line parallelized, with the block size ranging from 128×128 samples in P_f resolution (first "coarse" level in the motion search, applied on 1:4 downsampled P_f samples) down to 16×16 samples (final "fine" level for fractional motion prediction, carried out on 16:1 upsampled P_f samples). To speed up especially VVenC's *fast*(*er*) presets, MCTPF is applied to fewer l_f at at low rates (high QP_base) than at high rates (low QP_base) [5].

The MCTPF is the last process in the pre-analysis stage prior to the *inter-frame parallelized* (i. e., multithreaded across the frames of each l_f as well as CTU block lines) first and second R-D optimized encoding passes. It denoises the input samples in P_f associated with low l_f, as noted above, especially those l_f being referenced the most in RA coding. The filter is applied three-dimensionally, in both spatial and temporal direction, in blocks of 16×16 P_f samples, separately for the luma and, when available, chroma channels. To isolate and attenuate noise in picture P_f with sufficient accuracy, neighboring (in display order) pictures utilized in the temporal filtering are motion compensated for each 16×16 block $B_k \in P_f$. In other words, for each B_k, the co-located samples of the neighboring pictures P_{f-N}, … P_{f-1}, P_{f+1}, …, P_{f+N}, with $N \le 2$ for the *fast*(*er*) and $N \le 4$ for all other VVenC speed presets, are motion aligned relative to the current-picture samples of B_k. Basically, this alignment represents the motion compensation of each block co-located to B_k that results in a minimum (across the ME search space) mean inter-picture sample difference, abbreviated *minimum motion estimation error* (MMEE) hereafter. Such block-wise $MMEE_k$ values may be considered a rough estimate of residual quasi-random picture content in B_k not related to motion, texture, or structure – i. e., a "noise measure" estimate for block index k.

Using large enough search spaces and fractional-sample ME, as in VVenC, to reduce the risk of residual edges or texture in the motion difference affecting the MCTPF performance, one can argue that the higher $MMEE_k$, the more noise present in B_k.

3. IMPROVED SCREEN CONTENT DETECTION

The SCC introduced in Sec. 2, using low-order statistics $\mu_{0..3}$ and $\sigma_{0..3}$ to obtain quadrant-wise σ-equality figures, classifies a frame P_f as *weak* screen content when the sum (across Q) of all four σ-equality counts exceeds 25% of all 4×4-sample analysis blocks in P_f, and as *strong* screen content when the above holds and, in *each* of the four Q, the σ-equality count exceeds 25% of all 4×4 blocks in that Q. This simple approach reliably identifies screen sharing and some gaming content as at least

Figure 2. Examples of video content triggering false SCC in VVenC up to version 1.6.1. (left) Low lighting, saturation in JVET sequence *Campfire* [15], (right) black bars in 1080p movie *Tears of Steel* [16].

weak computer generated input. Unfortunately, however, high σ-equality counts may also occur in case of very dark camera captured input, saturating towards the minimum luma sample value μ_{min} (2^{BD-4} for BT.709 [14] input), as well as widescreen movies beyond 16:9 aspect ratio saved in full-HD or 4K resolution, thus having boundary black bars, as depicted in Fig. 2.

Since, as indicated in Fig. 2, black homogenous regions of P_f do not allow for clear distinction between camera captured and computer generated material, VVenC versions since 1.6.1 [17] exclude from σ-equality counting those 4×4 blocks with $\mu = \mu_{\text{min}}$ and $\sigma < 1$ before rounding σ to integer. This exclusion is aborted within a Q when 20% or more of all 4×4 blocks in that Q have already been excluded, to avoid deteriorating the SCC accuracy on actual screen sharing input with dark areas. In other words, once the threshold of 20% is reached in a Q, dark low-σ blocks are again included in the equality check. It was found that the use of this method renders dedicated black bar detection as in, e. g., the aomenc encoder [18] obsolete.

To reduce false-negative *strong* SCC of real screen content (classified as *weak* at most), the Q with the highest σ-equality count is identified and, when that count is at least 54% of the number of 4×4 blocks in that Q, *strong* SCC is enforced in P_f.

Regarding false-positive *strong* SCC on noisy camera captured videos, VVenC versions since 1.8.0 leverage the MMEE information outlined in Sec. 2 as follows. With $MMEE_k$ being a block-wise mean absolute difference between the samples in $B_k \in P_f$ and those of one neighboring motion compensated picture, there are actually $2N$ different $MMEE_k$ values associated with each B_k. To obtain a single "noisiness measure" for a B_k, $MMEE_k$ shall, hereafter, be defined as the minimum of all $2N$ MMEE values associated with B_k. The use of minimum statistics here is motivated by the usefulness of such an approach in noise level estimation for, e. g., SNR calculations in speech coding [19]. To derive a picture-wise overall $MMEE_f$ value, it is then reasonable to simply average the $MMEE_k$ results of all $B_k \in P_f$. Here, the average was found to be more robust against outliers (e. g., black B_k) than searching, again, for a minimum.

In each *strong*-SCC frame for which $MMEE_f$ exceeds a BD normalized threshold T, the MCTPF filtering step can then be reactivated. In VVenC, $T = 27 \cdot 2^{BD-12}$ was chosen empirically. Regarding this *noise thresholding*, two aspects shall be noted:

- One could extend the MMEE approach by, e. g., searching in each GOP g for the minimum $MMEE_f$ value and use the resulting $MMEE_g$ instead of $MMEE_f$. However, due to code parallelization and the need to, then, run MCTPF analysis in every f and l_f, this approach was not pursued in VVenC.
- Using $MMEE_f$, not $MMEE_g$, data allows to restrict the noise thresholding based SCC improvement to low temporal levels. In fact, the best tradeoff between SCC accuracy and complexity was found for limit $l_f < 1$, i. e., the restriction to key frames.

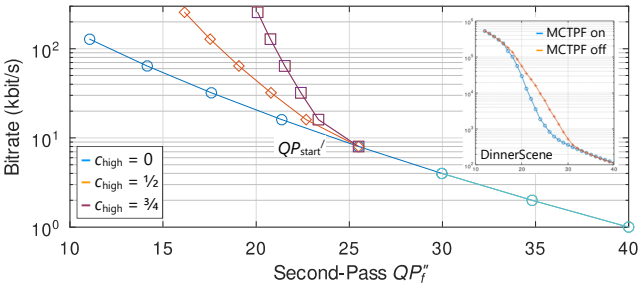

Figure 3. Example of *R-QP* function used in VVenC's RC. An arbitrary 1 kbit/s at $QP''_f = 40$ was chosen. (top right) actual video data [20].

4. IMPROVED TWO-PASS RATE CONTROL

VVenC's two-pass RC method, as described in [4, 6], obtains a first-pass overall QP_{base} from R_{target} and the video dimensions and, in an RA setting, derives therefrom a GOP-wise cascade of frame-wise QP values QP_f for use in the first encoding pass; *cf.* Fig. 1(a). The frame-wise bit consumption r_f resulting from this first pass is then used, along with QP_f and the targeted bit count r'_f, to calculate the QP''_f values for use in the second pass:

$$QP'_f = QP_f - c_{\text{low}} \cdot \sqrt{\max(1;\ QP_f)} \cdot \log_2\left(\frac{r'_f}{r_f}\right), \qquad (2)$$

where $c_{\text{low}} \approx 0.82$ is a constant for the low-rate end of the function and QP'_f is a preliminary second-pass QP value refined to

$$QP''_f = \text{round}\left(QP'_f + c_{\text{high}} \cdot \max(0;\ QP_{\text{start}} - QP'_f)\right), \quad (3)$$

with $0 < c_{\text{high}} < 1$ being a constant for the higher-rate end of the function and QP_{start} denoting a starting QP value below which corrective step (3) kicks in. Usage of the two-step *R-QP* model of (2, 3) is accompanied by an estimation of the second-pass overall QP''_{base}, details of which are omitted here for brevity.

In [4, 6], the *R-QP* model, illustrated in Fig. 3, is used with a fixed $QP_{\text{start}} = 24$ and a varying, video height dependent c_{high} (e. g., 0.5 for UHD and 0.25 for SD content). It was, however, observed recently that, for videos other than those in the CTC [15], a slightly better statistical fit of the model can be reached by using a fixed $c_{\text{high}} = 0.5$ and varying QP_{start} instead. Further, it was found that, by adjusting QP_{start} in each GOP g, based on a picture-wise "noise measure" determined for that GOP as in Sec. 3, an even better model fit can be achieved on some very low-noise as well as some very noisy video sequences. Thus,

- $MMEE_g$ is obtained once per g during the first RC pass, by finding the minimum of all available $MMEE_f$ values in g (as in Sec. 3, MCTPF analysis may not be run for each l_f),
- a noise level equivalent QP value QP^*_g is determined from $MMEE_g$, $QP^*_g = 24 + 0.5 \cdot (6 \log_2 MMEE_g + i - 24)$, and refined, $QP_{\text{start}} = QP^*_g + \log_2\left(\frac{W \cdot H}{3840 \cdot 2160}\right)$ with W and H as in Sec. 2, (4) in order to correct for bias due to more high-frequency signal energy, caused by sharper edges, at low video resolutions; in case no MMEE data are available, $QP^*_g = 24$ is used,
- the block-wise $MMEE_k$ are averaged across each CTU area and the results are used to improve the QP noise-limiter [6].

Constant i in (4) compensates for differences in BD and serves as a normalization factor (in linear domain) between quantizer step-size in VVC [21] and estimated noise level. $i = -1$ is used in this work. The $\log_2(\div)$ bias corrector was found experimentally. As can be seen in Fig. 3, with proper choices for QP_{start} and c_{high}, the *R-QP* model in VVenC's RC follows actual high-resolution video statistics quite closely below about 90 Mbps.

The use of CTU-wise averaged $MMEE_k$ values in the noise level based QP limiter – specifically, as an additional input to the block-luminance dependent *minimum statistics* estimator (i.e., on top of the block's spatial and temporal visual activity) [6] – significantly improves the noise estimation accuracy. As a result, lower noise levels are often calculated and fewer RC encoding cases occur in which R_{target} is not reached. On some single-scene videos like *DaylightRoad* [15], though, high-rate RC encodings still end up a few percent below the target rate.

The reason for this behavior is a suboptimal reallocation of bit budget saved during QP noise-limiting in the second RC pass: when, in a frame f, a CTU-level QP_k is increased in order not to end up below a noise level equivalent QP limit for that CTU, no effort is made to redistribute the change in QP_k value (and, thereby, rate savings) *within* f by reducing the QP_k of the CTUs which haven't been subjected to the QP-limiting yet. To conclude this study, a "rate recovery" method was, therefore, integrated into VVenC which, after QP-limiting, successively reduces by 1 the highest-value QP_k in f not already reduced or limited, until the mean of all QP_k in f equals the original mean QP before any limiting or all QP_k have been reduced or limited.

5. EXPERIMENTAL EVALUATION

The SCC and two-pass RC improvements described in Sec. 3 and Sec. 4, respectively, were implemented into a slightly improved variant of VVenC 1.7 (commit [22]), which served as a reference release for BD-rate evaluation [9] in RA configuration. To reflect a realistic use case, preset *fast* with activated multithreading, MCTPF, and FTA on top of a 4s Intra period was used. Other encoding options were configured according to JVET's common test conditions [15] or, when not specified there, as in [4, 6]. In particular, JVET CTC sequences shorter than 10s were extended to 10s from the original source videos, and Fraunhofer HHI's 10s-length *Berlin* set [23] was added.

5.1. Performance of SCC Modifications

The results of the evaluation of the improved SCC, determined on fixed-QP encodings without any perceptual QPA, are listed in Tab. 1, for both camera captured (CTC, HHI) and computer generated (TGM[1]) input. The BD-rates, tabulated separately for luma and chroma as well as a 6:1:1 YUV average, indicate substantial improvement in encoding performance for classes

- A1, thanks to the reactivation of MCTPF, which had been disabled in most frames, on *Campfire* (–1.8% BD-rate$_{\text{YUV}}$),
- TGM, due to *strong*, instead of *weak*, SCC in all but the 9 first and last frames of *ChineseEditing* (–15% BD-rate$_{\text{YUV}}$).

Large efficiency gains, i.e., several percent BD-rate reduction, could also be observed on some movie scenes with boundary black bars and picture noise, which are not part of this test set.

[1] Text and Graphics with Motion, 4:2:0 versions of a JVET set, see [24]

Table 1. PSNR based BD-rate results for SCC changes as in Sec. 3.

Resolution	**Luma**	**Chroma**		**Average**	Runtime	**Outlier Sequence**	
Class	Y	U	V	YUV	Ratio	max. |BD-rate$_{\text{YUV}}$|	
UHD A1	–0.4	–1.9	–1.7	**–0.6%**	101%	*Campfire*	–1.79%
UHD A2	0.00	0.00	0.00	**0.00%**	100%		
UHD HHI	0.00	0.00	0.00	**0.00%**	100%		
HD B	0.03	0.01	0.00	**0.02%**	101%	*RitualDance*	0.10%
HD HHI	0.03	0.06	0.00	**0.03%**	100%	*ReichstagTr.*	0.26%
HD TGM	–3.6	–2.4	–2.6	**–3.3%**	101%	*ChineseEd.*	–15.5%
SD C	0.00	0.00	0.00	**0.00%**	100%		

Table 2. XPSNR based BD-rate results for RC changes as in Sec. 4.

Resolution	**Fixed-QP vs. Ref.**		**Fixed-QP vs. Test**		RC **Ref. vs.** RC **Test**	
Class	YUV	Runtime	YUV	Runtime	YUV	Runtime
UHD A1	2.88%	114%	**3.09%**	115%	0.18%	101%
UHD A2	2.19%	116%	**2.47%**	117%	0.25%	101%
UHD HHI	5.14%	112%	**4.09%**	113%	–0.98%	101%
HD B	2.59%	110%	**2.70%**	112%	0.09%	102%
HD HHI	5.37%	104%	**3.01%**	108%	–2.14%	104%
SD C	1.82%	103%	**1.80%**	100%	–0.01%	96%
Overall	3.86%	110%	**3.04%**	111%	–0.75%	101%

5.2. Performance of RC Modifications

The BD-rates for the second experiment, obtained from YUV averaged XPSNR data as VVenC was operated in GOP-wise two-pass RC configuration with perceptual QPA [10, 11], are provided in Tab. 2. The sequence-wise target rate R_{target} for the RC encoding was obtained from fixed-QP CTC-like encoding runs (again with QPA and an Intra period of 4s) using VVenC 1.7.0, similar to [6]. The resulting per-class BD-rate averages, tabulated both relative to the baseline two-pass RC implementation and relative to the baseline fixed-QP RA setting, reveal notable gains in encoding efficiency without reductions in rate matching accuracy or increases in runtime. More specifically,

- sequence *DaylightRoad* in class A2 closely matches R_{target} due to the improved rate recovery, at a similar BD-rate$_{\text{YUV}}$,
- UHD sequences *Oberbaum* and *Quadriga* in class HHI are RC encoded with BD-rate$_{\text{YUV}}$ reductions around 3 to 4%,
- HD sequences *BerlinCrossroads*, *Oberbaum*, and *Quadriga* in class HHI show BD-rate$_{\text{YUV}}$ reductions of up to 10%.

The BD-rate performance in the conventional CTC classes A, B, and C changes only negligibly, which is desirable since a low level of picture noise can be found on the CTC sequences and the RC encoding configuration already achieves a performance very close to that of the fixed-QP encoding reference.

6. SUMMARY AND CONCLUSION

This paper completes recent studies towards the improvement of VVenC's screen content detector and two-pass rate control algorithm. The modifications, which focus on leveraging prefiltering statistics available from a MCTPF pre-analysis stage and whose benefits were confirmed experimentally on an extended set of test material, increase both the performance and stability in videoconferencing and online gaming applications as well as rate control based encodings of VoD movie content. Using this work in very fast RC [25] is a topic for future study.

REFERENCES

[1] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the Versatile Video Coding (VVC) Standard and Its Applications," *IEEE Trans. Circuits & Systems for Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[2] A. Wieckowski, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. R. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, I. Zupancic, B. Bross, and D. Marpe, "VVenC: An Open and Optimized VVC Encoder Implementation," in *Proc. IEEE ICME*, virtual/online, July 2021.

[3] ITU-T H.266 and ISO/IEC 23090-3, "Versatile Video Coding," Aug. 2020 (and subsequent editions). https://www.itu.int/rec/T-REC-H.266.

[4] C. R. Helmrich, I. Zupancic, J. Brandenburg, V. George, A. Wieckowski, and B. Bross, "Visually Optimized Two-Pass Rate Control for Video Coding Using the Low-Complexity XPSNR Model," in *Proc. IEEE Int. Conf. Visual Communic. & Image Process. (VCIP)*, Munich, Dec. 2021.

[5] A. Wieckowski, T. Hinz, C. R. Helmrich, B. Bross, and D. Marpe, "An Optimized Temporal Filter Implementation for Practical Applications," in *Proc. IEEE Picture Coding Symposium (PCS)*, San Jose, Dec. 2022.

[6] C. R. Helmrich, C. Bartnik, J. Brandenburg, V. George, T. Hinz, C. Lehmann, I. Zupancic, A. Wieckowski, B. Bross, and D. Marpe, "A Scene Change and Noise Aware Rate Control Method for VVenC, an Open VVC Encoder Implementation," in *Proc. IEEE PCS*, San Jose, Dec. '22.

[7] M. Wien and V. Baroncini, "Report on subjective performance evaluation of the ECM," document *JVET-AB0270*, version 2, Mainz, Oct. 2022.

[8] M. Wien and V. Baroncini, "Training Methods in Visual Assessment: Potential Improvements for Expert Viewing Tests," document *JVET-AC0267*, teleconf., Jan. 2023. https://jvet-experts.org/ → All meetings.

[9] ITU-T HSTP-VID-WPOM and ISO/IEC TR 23002-8, "Working practices using objective metrics for evaluation of video coding efficiency experiments," 2021. https://www.itu.int/pub/T-TUT-ASC-2020-HSTP1

[10] C. R. Helmrich, S. Bosse, H. Schwarz, D. Marpe, and T. Wiegand, "A Study of the Extended Perceptually Weighted Peak Signal-to-Noise Ratio (XPSNR) for Video Compression with Different Resolutions and Bit Depths," *ITU Journal: ICT Discoveries*, vol. 3, no. 1, May 2020. http://handle.itu.int/11.1002/pub/8153d78b-en.

[11] C. R. Helmrich, S. Bosse, M. Siekmann, H. Schwarz, D. Marpe, and T. Wiegand, "Perceptually Optimized Bit-Allocation and Associated Distortion Measure for Block-Based Image or Video Coding," in *Proc. IEEE Data Compress. Conf. (DCC)*, Snowbird, pp. 172–181, Mar. 2019.

[12] J. Enhorn, R. Sjöberg, and P. Wennersten, "A Temporal Pre-Filter For Video Coding Based on Bilateral Filtering," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, pp. 1161–1165, Oct. 2020.

[13] J. Enhorn, C. Hollmann, R. Sjöberg, K. Andersson, and P. Wennersten, "Improvements to a Temporal Filter for Video Coding," in *Proc. IEEE Int. Symp. Circuits & Systems (ISCAS)*, Austin, pp. 834–838, May 2022.

[14] Int. Telecommunic. Union, "Parameter values for the HDTV standards for production and internal programme exchange," recommendation ITU-R BT.709-6, July 2015. https://www.itu.int/rec/R-REC-BT.709/en.

[15] F. Bossen, X. Li, K. Sharman, V. Seregin, and K. Sühring, "VTM and HM common test conditions and software reference configurations for SDR 4:2:0 10-bit video," document *JVET-AB2010*, teleconf., Jan. 2022.

[16] Blender Foundation, "Tears of Steel", open movie, 2.35:1 aspect ratio, Creative Commons Attrib., 2012. https://mango.blender.org/download.

[17] Fraunhofer HHI, "Fraunhofer Versatile Video Encoder (VVenC)," GitHub repository, v1.6.1, 2022. https://github.com/fraunhoferhhi/vvenc.

[18] Alliance for Open Media, "AV1 Codec Library aomenc," Git repository, src file *firstpass.c*, Feb. 2023. https://aomedia.googlesource.com/aom/.

[19] R. Martin, "An Efficient Algorithm to Estimate the Instantaneous SNR of Speech Signals," in *Proc. EuroSpeech*, Berlin, Germany, Sep. 1993. www.isca-speech.org/archive_v0/eurospeech_1993/e93_1093.html.

[20] P. de Lagrange, "AHG4: Experiments in preparation of film grain visual tests," document *JVET-AC0181*, section 6, figure 5, teleconf., Jan. 2023.

[21] H. Schwarz, M. Coban, M. Karczewicz, T.-D. Chuang, F. Bossen, A. Alshin, J. Lainema, C. R. Helmrich, and T. Wiegand, "Quantization and Entropy Coding in the Versatile Video Coding (VVC) Standard," *IEEE Trans. Circuits & Systems for Video Technol.*, vol. 31, no. 10, pp. 3891–3906, Oct. 2021.

[22] Fraunhofer HHI, "Fraunhofer Versatile Video Encoder (VVenC)," commit https://github.com/fraunhoferhhi/vvenc/commit/a2ec456a3, 2023.

[23] B. Bross, H. Kirchhoffer, C. Bartnik, M. Palkow, and D. Marpe, "AHG4 Multiformat Berlin Test Sequences," document *JVET-Q0791*, Jan. 2020.

[24] H. Yu, R. Cohen, K. Rapaka, and J. Xu, "Common test conditions for screen content coding," document *JCTVC-Z1015*, Geneva, Jan. 2017. http://phenix.it-sudparis.eu/jct/ → All meetings.

[25] V. V Menon, A. Henkel, P. T Rajendran, C. R. Helmrich, A. Wieckowski, B. Bross, C. Timmerer, and D. Marpe, "All-Intra Rate Control Using Low Complexity Video Features for Versatile Video Coding," submitted to *IEEE Int. Conf. Image Process. (ICIP)*, Kuala Lumpur, Feb. 2023.