A Constrained Variable Bit Rate (CVBR) Algorithm for VVenC, an Open VVC Encoder Implementation

Christian R. Helmrich, Christian Bartnik, Jens Brandenburg, Adam Wieckowski, Benjamin Bross, and Detlev Marpe

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

***Abstract*—Rate control (RC) schemes allow audio and video encoders to produce bitstreams according to specific overall bitrate constraints. However, when no rate capping is enforced, the instantaneous bitrate may vary strongly and may exceed the target rate by an order of magnitude, potentially causing playback stutter especially in video streaming scenarios. This paper introduces a rate capping extension for the two RC modes in VVenC, an open Versatile Video Coding (VVC) compliant encoder implementation. After a revisit of VVenC's two-pass RC approach, the algorithmic details of the rate capping model are described. The paper concludes with an objective evaluation of the performance of the RC extension in a random-access configuration.**

***Keywords*—QoE, rate control, video coding, VoD, VQA, VVC**

I. Introduction

In virtually all audio and video coding scenarios, more or less strict constraints are imposed by the encoder operator in terms of (**a**) average, or *target*, bitrate R_{trgt} across a (relatively long) interval of the media content such as a film or a podcast episode, and (**b**) maximum instantaneous bitrate $R_{\max}$, usually determined across a (relatively short) interval such as one or two seconds around the frame f being encoded. In traditional broadcasting applications, these long-term and short-term bitrate constraints must be enforced rigorously, in order to avoid signal dropout or stutter at the receiver side, i. e., degradation in quality of service (QoS) on the consumer devices. Nevertheless, Web based streaming and teleconferencing solutions benefit from a relatively narrow value range between R_{trgt} and $R_{\max}$ as well, as such a configuration minimizes the likelihood of rebuffering and/or resolution reduction during playback.

The choice for R_{trgt} and $R_{\max}$, with the latter given relative to the former in most cases (e. g., $R_{\max} = 2 \cdot R_{\mathrm{trgt}}$), is, therefore, a question of quality of experience (QoE) stability, and empirical studies indicate that the employed values differ strongly among applications [1]. For this reason, most audio and video encoders support some variant of constrained variable bitrate (CVBR) encoding. For example, exhale, an open Extended HE-AAC [2] encoder implementation [3], provides, like other recent HE-AAC encoders [4], bitrate presets which guarantee created bitstreams to exhibit rates within the respective associated range $\{0.5 \cdot R_{\mathrm{trgt}}, 1.5 \cdot R_{\mathrm{trgt}}\}$, i. e., with $R_{\max} \approx 1.5 \cdot R_{\mathrm{trgt}}$ [5]. VVenC, an open VVC [6] encoder implementation [7], is one of the few video encoders with, at the time of this writing, a *bitrate* option (R_{trgt}, for RC operation) but no *maxrate* option.

A. Related Work

Regarding rate capping in video streaming and videoconferencing scenarios, where strict buffer requirements usually do not apply, only few academic studies appear to have been published. Dagher *et al.* [8] present a *leaky bucket* based RC method for constrained scalable Motion JPEG2000 encoding. Owing to the inherently Intra-only (no motion compensation) encoding paradigm in their experiments, no measures for rate allocation or restriction across a group of pictures (GOP) are investigated. In random-access (RA) VVC streams, however, GOPs as large as 32 hierarchically arranged—and, thus, interdependent—pictures are utilized, thus rendering a relatively simple approach as that of [8] impractical in the VVC context.

Bao *et al.* [9] and Kim *et al.* [10] describe approaches for improving the QoE at the client side in DASH applications, via online dynamic video bitrate selection or power consumption capping algorithms, respectively. In other words, a low bitrate variation already during *encoding* is not a topic of these DASH architecture specific, media codec agnostic studies. One of the publications most relevant to the use case at hand is the paper by Blestel *et al.* [11], where a constant quality control (CQC) algorithm is proposed. However, that work controls an HEVC encoding run by enforcing GOP-wise average and maximum *distortion* conditions, upon which R_{trgt} and $R_{\max}$ then depend. Hence, direct limitation to $R_{\max}$ is not the scope of that study, although, arguably, customers are, likely, much more familiar with the usage of rate values than of distortion or quantization parameter (QP) values when configuring typical video coders.

Lin *et al.* [12] present a RC scheme for VVC, particularly for 360-degree video, but the description of the applied Intra-frame rate capping remains vague and indirect: the "proposed scheme constrains the frame-level QP of each Intra frame". A similar, also vaguely described approach is pursued by Menon *et al.* [13], capping each GOP-level QP (denoted constant rate factor, CRF) to some $c_{\max}$ during the second pass in a CQC-like setting. VVenC's RC, configured directly by a target bitrate to facilitate its usage, has been described by Helmrich *et al.* in [14 – 16]. It would be beneficial for users to, in addition, be able to customize $R_{\max}$ when configuring the encoder and to enforce $R_{\max}$ with acceptable accuracy during encoding, similarly to the CQC and capped rate/distortion designs in [11–13].

B. Contribution

In this paper, a rate cap extension to VVenC's rate control, making use of an additional *maxrate* parameter as a means to adjust $R_{\max}$, is proposed. Contrary to some of the prior work discussed above, the extension uses motion error and rate statistics already calculated by the encoder. Thus, it requires virtually no additional computational complexity. In order to limit $R_{\max}$ to a reasonable value range, an upper bound of $R_{\max} = 3 \cdot R_{\mathrm{trgt}}$ was realized in both the sequence-wise two-pass RC [14] as well as the GOP-wise, look-ahead based RC [15] with, as will be demonstrated, almost no reduction of the encoding efficiency. Users may then specify $R_{\max}$ freely within the range $\{1.5 \cdot R_{\mathrm{trgt}}, 3 \cdot R_{\mathrm{trgt}}\}$, covering the vast majority of use cases [1].

C. Paper Outline

The remainder of this paper is organized as follows. Sec. II revisits the *R-QP* model applied in VVenC's RC, along with rate matching specific algorithmic details of the second (i. e., final) encoding pass. Sec. III then outlines the additional steps proposed to realize rate capping functionality in the final RC pass and describes how existing frame statistics are leveraged for this purpose. The preparation and outcome of evaluation experiments conducted to assess the effect of the proposal on VVenC's coding efficiency (in terms of BD-rate) are outlined in Sec. IV, and Sec. V summarizes and concludes the paper.

II. Revisit of VVenC's Rate Control Model

To produce encodings averaging at a user specified target rate R_{trgt}, VVenC's two-pass RC method employs a two-step *R-QP* model [14]. Based on preliminary frame QP, q_f, and bit consumption, r_f, statistics collected in a (fast) rate-distortion (RD) optimized first encoding pass governed by fixed overall quantization parameter QP_{base}, the (full featured) second RD optimized pass is performed with frame-wise final QP values

$$q''_f = \left\lfloor q'_f + c_{\mathrm{high}} \cdot \max\left(0;\ QP_{\mathrm{start}} - q'_f\right) + o_l + \tfrac{1}{2} \right\rfloor \quad (1)$$

with

$$q'_f = q_f - c_{\mathrm{low}} \cdot \sqrt{\max\left(1;\ q_f\right)} \cdot \log_2\left(\frac{r''_f}{r_f}\right), \quad (2)$$

where r''_f denotes the frame-wise second-pass target bit count. Constants $c_{\mathrm{low}} \approx 0.82$, $c_{\mathrm{high}} = 0.5$ were chosen empirically, and QP_{start} depends on the video size and GOP-wise updated noise statistics [16, 17]. Details thereon and on the video resolution, Intra-frame period, and R_{trgt} dependent specification of QP_{base} shall be omitted here for brevity. The relationships between QP_{base} and q_f, as well as between q_f and the corresponding Lagrange parameter λ_f, for RD optimized encoding are adopted from VTM, the JVET reference software implementation for VVC [18]. Figure **1** depicts the nonlinear behavior of the two-step *R-QP* model for different values of c_{high}, with data points emphasized at power-of-two multiples of 1000 bit/s. It further demonstrates how, at moderate second-pass q''_f values (center of figure), a reduction of said frame QP by approximately 4.5 doubles the resulting frame bit count, denoted r^{res}_f hereafter.

A. Corrective Adjustments During Second RC Pass

Note how, in (1), a corrective, temporal level l dependent offset $-12 \le o_l \le 12$ is added before the obligatory rounding to integer. Such a QP correction is required since VVenC's *R-QP* model is only a relatively simple approximation, resulting in second-pass q''_f being off by (typically) one or two QP values from the ideal choice for some frames. Each o_l is, therefore, constantly updated during the second RC pass based on past per-f pairs of allocated, r''_f, and resulting, r^{res}_f, bit count data:

$$o_l = \max\left(-12;\ \min\left(12;\ \alpha \cdot \log_2\left(\frac{\sum_{j\in B} r^{\mathrm{res}}_j}{\sum_{j\in B} r''_j}\right)\right)\right) \quad (3)$$

with

$$\alpha = c_{\mathrm{low}} \cdot \sqrt{QP_{\mathrm{avg}}},\ \ B = \text{set of all past frames } f \text{ at level } l, \quad (4)$$

where QP_{avg} is the mean of all q''_f in the last Intra period, as in [15]. $\alpha \cdot \log_2()$ approximates the lower limit (at $c_{\mathrm{high}} = 0$) of the

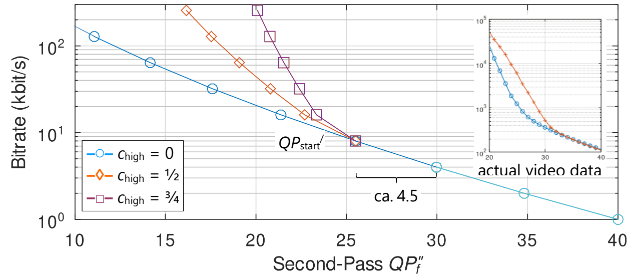

Figure 1. Example of two-step *R-QP* function (1, 2) used in VVenC's RC. An arbitrary 1 kbit/s at $q''_f = 40$ was taken. Varying QP_{start} moves the orange and violet curve horizontally along the blue $c_{\mathrm{high}} = 0$ curve.

R-QP model; *cf.* Fig. 1. In this way, the rate matching accuracy of (1, 2) can be improved over time. For best performance, the sums in (3) are additionally zeroed out at scene cuts ($B = \emptyset$), and $o_l = 0$ is assumed when the denominator sum equals zero. The last parameter in (2, 3) not yet introduced, the final-pass target bit count r''_f, depends on the RC encoding mode in use:

- With *GOP-wise, look-ahead based RC*, a full GOP (here, 32 frames) of new picture data is encoded in the first and, then, the final pass, with parallel processing in both passes in case of multithreading. Since, consequently, each pass does not process the entire video sequence at once (for use in *on-the-fly* applications), all r''_f for an Intra-frame period I, to be defined as an integer multiple $I_G = I/G$ of GOP size G herein, are determined based on the first-pass encoding results for the new (i. e., look-ahead) and the last I_G GOPs:

$$r''_f = \left\lfloor \max\left(1;\ r'_f + \left(\tfrac{F_C \cdot R_{\mathrm{trgt}}}{fps} - \textstyle\sum_{j\in C} r^{\mathrm{res}}_j\right) \cdot d \cdot \frac{r'_f}{g'_f}\right)\right\rfloor \quad (5)$$

with

$$r'_f = \left\lfloor r_f \cdot \frac{R_{\mathrm{trgt}} \cdot I}{fps \cdot a_f} + \frac{1}{2} \right\rfloor,\ \ a_f = \sum_{l=0}^{l_{\max}} \frac{I_G \cdot \mathrm{mean}_l(r_{f\in A})}{\min(1;\ 2^{2-l})}, \quad (6)$$

where A is the set of all frames f in either the last I_G GOPs or the new GOP, i. e., in the analysis window [15]. C is the set of all f already encoded in both passes, and constant d is set to 1 for all f in the last encoded GOP, else to 0.5 [14]. F_C counts the already final-pass encoded frames in C, while mean_l denotes averaging of data in frames having level l, with $l_{\max} = 6$ here as $G = 32$, and $l > 0$ for non-Intra frames.
- With *sequence-wise, file based RC*, the entire video is RD encoded in the first pass, after which the second-pass bit counts r''_f are determined. Since the overall bit consumption resulting from the first-pass encoding is known prior to starting the second pass, the r''_f can be calculated easily:

$$r''_f = \left\lfloor \max\left(1;\ r'_f + \left(\textstyle\sum_{j\in C} r'_j - r^{\mathrm{res}}_j\right) \cdot d \cdot \frac{r'_f}{g'_f}\right) + \frac{1}{2} \right\rfloor \quad (7)$$

with

$$r'_f = \left\lfloor r_f \cdot \frac{R_{\mathrm{trgt}} \cdot F}{fps \cdot \sum_f r_f} + \frac{1}{2} \right\rfloor,\ \ F = \text{total frame count}. \quad (8)$$

Thus, to obtain the second-pass bit counts r'', (8) averages the first-pass bit counts across all frames, while (6) applies the averaging only across frames in temporal window A.

In both modes, g'_f is the sum of all budget agnostic (uncorrected) target bit counts r' in the GOP f is associated with, giving *frame-to-GOP* ratio r'_f/g'_f, whereas *fps* is the frame rate in Hz. Note that C is a superset of all sets of level-wise B, i. e., $B \subseteq C$.

III. CONSTRAINED VBR CODING WITH VVENC

The corrective measures employed in VVenC's RC modes during the second encoding pass ensure that, on average, the finished bitstream exhibits R_{trgt} as closely as possible. Specifically, the QP correction of (3, 4) serves to improve the fidelity of the *R-QP* model (1, 2) as the final-pass encoding progresses while (5, 7), governing said model via (2), ensure that any bit rate excessively, or only partially, consumed by already final-pass encoded frames is accounted for during the rate allocation for following frames to be encoded. In any frame f, bit budget

$$b_f = \sum_{j \in C} r'_j - r_j^{\text{res}} = \sum_{j \in C} r'_j - \sum_{j \in C} r_j^{\text{res}},\ j < f, \quad (9)$$

representing the difference between estimated and actual (i.e., resulting) frame bit consumption accumulated in the final RC pass, can be determined prior to encoding frame f. Positive b_f values indicate that additional bit budget is available for spending in f, whereas negative b_f imply that bits must be saved in f. In (5) and (7), b_f is being adopted with additional *frame-to-GOP* scaling so as to maintain, during the final pass, the first-pass rate distribution among the different frames in each GOP and, thereby, a high coding efficiency especially in RA cases.

This design, while yielding good *overall* rate matching and subjective (visual) as well as objective (RD efficiency) performance [14–16], does not consider *instantaneous* rate behavior at any point in the generated bitstreams. This instantaneous R_{i} which, for simplicity, shall be defined hereafter as an average across the sliding analysis window A introduced in Sec. II.A,

$$R_{\text{i}} = \frac{fps}{I} \cdot \sum_{f \in (A \cap C)} r_f^{\text{res}}, \quad (10)$$

may, in particular, greatly exceed the average overall R_{trgt}, i. e., $R_{\text{i}} \gg R_{\text{trgt}}$. In fact, on the publicly available high-quality UHD sequences *TearsOfSteel* [19] and *SolLevante* [20], the authors noticed rate differences of up to a factor of thousand between individual scenes (e. g., strong and irregular motion vs. movie credits) during fixed-QP encoding and, thereby, $R_{\text{i}} > 10 \cdot R_{\text{trgt}}$ in some scenes. To reduce the risk of playback issues as noted in Sec. I, modifications to (5–8) are presented in the following, effectively allowing for rate capping such that $R_{\text{i}} \le R_{\max}$ in any GOP of the resulting bitstream, with $R_{\max}$ selected by the user.

Before illuminating algorithmic details, it is worth noting that bitstreams resulting from rate capped encoding runs may be analyzed in various ways to assess their instantaneous rate behavior, especially regarding the length of the temporal interval across which the rate behavior is being measured. Due to the hierarchical GOP structuring in RA coding, where most of the rate is allocated to the low temporal levels l, short-interval measurements typically result in high instantaneous rate fluctuation during third-party analysis. Thus, $R_{\max}$ may seem to be exceeded in a GOP, starting at f_{g}, despite $R_{\text{i}} \le R_{\max}$ having been enforced during encoding. Such measurement fluctuations are exacerbated by the fact that GOPs containing Intra-only coded ($l = 0$) key frames, named *I-GOPs* hereafter, usually consume a notably larger share of the available bits than GOPs without I-frames, i. e., *non-I-GOPs* using non-Intra ($l = 1$) key frames.

Since, in I-GOPs, the frame-to-GOP ratio r'_{f0}/g'_{f0} for the I frame (at frame index f_0) indicates the inter-frame bit distribution within each Intra period quite well, and an instantaneous rate measurement interval of a few GOPs may be assumed, the maximum allowed GOP bit count may be defined as follows:

$$g_{\max} = \frac{R_{\max}}{fps} \cdot \frac{G \cdot I}{I + m_0 \cdot G} \cdot \begin{cases} 1 + m_0 & \text{for I-GOPs} \\ 1 & \text{otherwise,} \end{cases} \quad (11)$$

with $m_0 = r'_{f0}/g'_{f0}$. For high I-frame-to-GOP ratios $m_0 \approx 1$ with e. g. very little and regular motion, (11) restricts the maximum bit count for I-GOPs to twice the bit count for non-I-GOPs in each Intra period. In scenes with strong, irregular motion, on the other hand, $m_0 \approx 1/G$, which results in I-GOPs and non-I-GOPs having almost the same maximum bit count. For both of these extreme cases, (11) was found to maintain sufficient visual quality and $R_{\max}$ adherence during user measurements.

A. Constrained VBR for GOP-Wise, Look-Ahead RC

To realize flexible VBR coding with VVenC's look-ahead based RC, with or without GOP-wise rate capping using $g_{\max}$ of (11), four aspects specific to VVenC need to be addressed.

First, the term $F_C \cdot R_{\text{trgt}}/fps$ in (5), serving as an estimate of the consumed second-pass bits, was found to be inappropriate with RA encoding since it assumes identical bit consumption in each frame. As mentioned earlier, the bit distribution varies between I and non-I GOPs and between different l within each GOP. Considering that budget-uncorrected target bit counts r'_f are readily available in each frame and that all r' were derived, in (6), from R_{trgt} as well as an estimate a_f/I of the instantaneous first-pass rate in the vicinity of f (window A), it is proposed to apply (7), used in sequence-wise RC, also with GOP-wise RC.

Second, it is worth noting that (6) results in R_{i} approaching R_{trgt} in every Intra period of the generated bitstreams. To allow for more variability, especially for temporarily increased R_{i} in hard-to-compress scenes to boost encoding efficiency, a solution was devised which (**a**) saves the frame-average minimum motion estimation error, $MMEE_f$, resulting from temporal prefiltering analysis, of all filtered f [16, 21], (**b**) finds the maximum $MMEE_f$ value in each GOP at f_{g}, assuming value 0 for all unfiltered f, (**c**) stores the maxima of the last eight GOPs in a circularly updated buffer, (**d**) obtains the average μ of all non-zero maxima in the buffer before final-pass encoding each f_{g}.

Third, when the maximal $MMEE$ of (b) exceeds μ by at least

$$T_\mu = \frac{2^{BD-6} \cdot R_{\text{trgt}}}{\min(2 \cdot R_{\text{trgt}}; R_{\max})}, \text{ where } BD = \text{coding bit-depth}, \quad (12)$$

at the start of a GOP (i. e., at f_{g}), b_f of (9) is relaxed by setting

$$b_f = \max(0;\ b_f) \Leftrightarrow \sum_{j \in C} r'_j = \max\left(\sum_{j \in C} r_j^{\text{res}};\ \sum_{j \in C} r'_j\right), \quad (13)$$

thus clearing negative bit budget states. In addition, r'_f of (6) is scaled by the ratio of maximum $MMEE$ and $\mu + T_\mu$, which is larger than 1, to allocate proportionally more bits to the selected hard-to-compress GOPs without causing new negative b_f. Note that motion error and rate statistics are readily available, so the above *rate boosting* adds no computational complexity.

Fourth, rate capping, via (5, 7, 11), to $g''_f \le g_{\max}$ is applied in each f of each GOP, to enforce $R_{\text{i}} \le R_{\max}$ in the second pass:

$$\bar{r}''_f = \min\left(g_{\max} \cdot \frac{r'_f}{g'_f};\ r''_f\right), \text{ i.e., } g''_f \approx r''_f \cdot \frac{g'_f}{r'_f}. \quad (14)$$

Then, $\bar{r}''_f$ is used instead of r''_f in (2). Here, g''_f denotes the yet unknown actual GOP bit count, hence the indirect but accurate approximation using the first-pass frame-to-GOP ratio r'_f/g'_f.

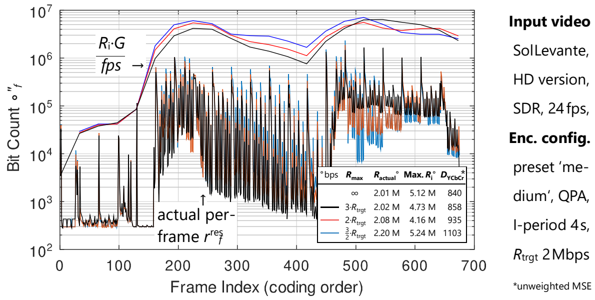

Figure 2. Effect of rate capping, using different R_{max}, when encoding the first 21 GOPs of *SolLevante* [20] with VVenC's sequence-wise RC.

B. Rate Capping for Sequence-Wise, File Based RC

The rate boosting of Sec. III.A can only increase r'_f temporarily, until μ "catches up" with increased motion activity (i. e., *MMEE* maxima). The sequence-wise RC first-pass encodes a full video at once, so a simpler approach can be used here that

- identifies all I-GOPs, determines m_0 and g_{max} for each, and applies (14) but on the r'_f instead of (not yet calculated) r''_f,
- when GOP x was rate capped in the previous step, flags x and finds the difference between its initial and capped rate,
- sums up these rate differences across all flagged GOPs and redistributes the sum evenly among all non-flagged GOPs by adding the rate share to r'_f and applying (14) on it again,

before calculating (7, 14), and using $\bar{r}''_f$ in (2), in the final pass. Figure **2** shows, on the *SolLevante* intro, how this form of rate capping flattens peaks in the R_i curve when R_{max} is decreased.

IV. Experimental Evaluation On Known Test Set

To more thoroughly quantify their effects on better-known sets of videos, the extensions of Sec. III were evaluated in RA configuration, via Bjøntegaard delta-rate (BD-rate) measurements [23]. The required changes to VVenC's code base were implemented on top of GitHub commit *ec61375* (June 2023), serving as BD-rate reference [7]. Additionally, d in (5, 7) was scaled by $1 + m_0$ in I-GOPs to maximize efficiency. As in prior RC related publications [14–16], speed preset *fast*, GOP size 32, multithreading, MCTPF [24], and XPSNR based QPA for perceptual optimization [25] were employed. All experiments were conducted according to JVET's common test conditions (CTC) for SDR video [26], with the class-A UHD sequences extended to 10 s duration and Fraunhofer HHI's public *Berlin* sequences [27] added for more content diversity. The four R_{trgt} values for each video were obtained via fixed-QP coding with $QP_{base} = 22, 27, 32, 37$ and calculation of the resulting bitrates. The RC rate matching accuracy, *BitErr*, is measured as in [28].

XPSNR based BD-rate results for different R_{max} as well as for the "baseline" RC condition without rate capping, all 6:1:1 averaged across the Y, C_b, C_r components [23] and video class, are listed in Tables **I** and **II** for the GOP-wise and sequence-wise RC modes, respectively. They indicate, in particular, that

- the efficiency of the GOP-wise RC benefits from the four modifications of Sec. III.A when $R_{max} \geq 2 \cdot R_{trgt}$, especially on sequences with scene cuts like *MarketPlace* (HD, ±½–¾ dB XPSNR), where subjective quality improves as well,
- the *BitErr* numbers increase with GOP-wise RC coding as R_{max} increases, which may be expected since more bits can be spent in GOPs with suddenly increased motion activity,

Table I. XPSNR [22] BD-rate and *BitErr* results for GOP-wise RC.

Resolution	**no cap, $R_{max} = \infty$**		**cap $R_{max} = 2 \cdot R_{trgt}$**		**cap $R_{max} = \frac{3}{2} \cdot R_{trgt}$**	
Class	BD-rate	*BitErr*	BD-rate	*BitErr*	BD-rate	*BitErr*
UHD A½	–2.65%	0.91%	–2.65%	0.91%	–2.68%	0.91%
UHD HHI	–2.32%	2.80%	–2.34%	2.81%	–2.23%	2.56%
HD B	–2.18%	4.81%	–2.73%	3.59%	–2.58%	3.39%
HD HHI	–0.75%	3.14%	–0.79%	3.52%	–0.94%	4.90%
SD C	–1.61%	1.98%	–1.62%	1.97%	–1.66%	2.11%
Overall	**–1.87%**	**2.74%**	**–1.97%**	**2.64%**	**–1.97%**	**2.92%**

Table II. XPSNR BD-rate and *BitErr* results for sequence-wise RC.

Resolution	**no cap, $R_{max} = \infty$**		**cap $R_{max} = 2 \cdot R_{trgt}$**		**cap $R_{max} = \frac{3}{2} \cdot R_{trgt}$**	
Class	BD-rate	*BitErr*	BD-rate	*BitErr*	BD-rate	*BitErr*
UHD A½	–0.33%	0.43%	–0.34%	0.53%	2.39%	2.54%
UHD HHI	–0.54%	0.44%	0.32%	0.60%	5.53%	1.58%
HD B	–0.39%	0.89%	0.16%	1.44%	4.13%	2.53%
HD HHI	–0.67%	1.70%	1.06%	2.18%	4.35%	4.27%
SD C	–0.12%	0.94%	–0.06%	1.05%	2.06%	4.09%
Overall	**–0.46%**	**0.90%**	**0.31%**	**1.19%**	**3.95%**	**2.94%**

- the results for the sequence-wise RC with $R_{max} = 2 \cdot R_{trgt}$ and the noncapped baseline RC are almost identical, with HHI sequence *Quadriga* (±4% BD-rate) as the main exception,
- the RC accuracy increases in sequence-wise RC coding as R_{max} increases, which may also be expected since fewer bit budget related corrections must be applied during the final encoding pass (the likelihood of rate clippings decreases).

A deeper analysis of the performance difference on UHD and HD sequence *Quadriga* reveals that this very easy-to-encode scene (low noise and motion activity and, thus, output rates at the CTC QP_{base} values with fixed-QP encoding) requires allocating most of the bit budget in the I-frames. Since, however, Sec. III proposed to limit the maximum bit budget for I-GOPs to twice the budget for non-I-GOPs in a given Intra period, the value range of the hierarchical frame-QP cascade on such input is, effectively, reduced (or compressed) by the RC, thereby causing a slight loss in coding efficiency. Overall, though, the usage of $R_{max} = 2 \cdot R_{trgt}$ seems to represent an efficient tradeoff between target and maximum rate for Web based applications. Also, with the exception of case $R_{max} = \infty$ in GOP-wise RC or case $R_{max} \leq 1.5 \cdot R_{trgt}$ in both two-pass RCs, where precise rate matching in RA becomes difficult, RC accuracy remains high.

V. Summary and Conclusion

This paper outlined a constrained variable bitrate (CVBR) mode for VVenC, realized via low-complexity estimation and limitation of the instantaneous bitrate (called rate capping) via option R_{max} during two-pass rate control (RC) encoding. With GOP-wise, look-ahead based RC and high R_{max}, the proposal improves VVenC's efficiency, in terms of BD-rate and visual quality, especially on videos with scene changes (at the cost of a slightly reduced rate accuracy). Users of the sequence-wise, file based RC, on the other hand, benefit, via R_{max}, from more control over the tradeoff between bitrate variance and coding efficiency across a video sequence (without having to worry about resulting bitstreams reaching R_{trgt} at least for reasonably high R_{max}). Future studies will focus on enforcing R_{max} during fixed-QP encoding for CRF-like functionality as in, e. g., [13].

References

[1] J. Ozer, "Constrained VBR Levels of the Rich and Famous," *Streaming Learning Center*, Apr. 1, 2022. https://streaminglearningcenter.com/codecs/constrained-vbr-levels-of-the-rich-and-famous.html.

[2] S. Quackenbush, "MPEG Unified Speech and Audio Coding," *IEEE Multimedia*, vol. 20, no. 2, pp. 72–78, Apr. 2013.

[3] C. R. Helmrich, project ecodis, "exhale: ecodis extended high-efficiency and low-complexity encoder," version 1.2, *Gitlab repository*, Dec. 2022. https://gitlab.com/ecodis/exhale.

[4] HydrogenAudio, "AAC encoders," *HydrogenAudio Knowledgebase*, Jan. 2017. https://wiki.hydrogenaud.io/index.php?title=AAC_encoders.

[5] C. R. Helmrich, project ecodis, "exhale Wiki: Frequently Asked Questions (FAQ)," *Gitlab Wiki*, Dec. 2022. https://gitlab.com/ecodis/exhale/-/wikis/faq.

[6] 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 Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[7] Fraunhofer HHI, "Fraunhofer Versatile Video Encoder (VVenC)," *Github repository*, June 11, 2022. https://github.com/fraunhoferhhi/vvenc.

[8] J. C. Dagher, A. Bilgin, and M. W. Marcellin, "Resource-Constrained Rate Control for Motion JPEG2000," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1522–1529, Dec. 2003.

[9] Y. Bao, L. Zhang, W. Wang, X. Gong, and X. Que, "Optimizing Playback Quality of HTTP-Based Dynamic Adaptive Streaming on Smartphones," in *Proc. IEEE ICSPCC*, Ningbo, Sep. 2015.

[10] G. Kim, D. Lee, and M. Song, "Design and Implementation of Bitrate Adaptation Schemes for Power Capping in Wi-Fi Video Streaming," in *IEEE Access*, vol. 8, pp. 22581–22591, Jan. 2020.

[11] M. Blestel, J. Le Tanou, and M. Ropert, "Constant Quality Control Based on Temporal Distortion Backpropagation in HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Athens, pp. 3279–3283, Oct. 2018.

[12] Y-H. Lin, C.-Y. Chen, and C.-W. Tang, "VVC Based Rate Control Using SKIP CTU Predictor," in *Proc. IEEE ICCE-Asia*, Yeosu, Oct. 2022.

[13] V. V Menon, H. Amirpour, M. Ghanbari, and C. Timmerer, "ETPS: Efficient Two-Pass Encoding Scheme for Adaptive Live Streaming," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Bordeaux, pp. 1516–1520, Oct. 2022.

[14] 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 Commun. Image Process. (VCIP)*, Munich, Dec. 2021.

[15] 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 Picture Coding Sympos. (PCS)*, San Jose, pp. 241*ff.*, Dec. 2022. www.ecodis.de/ratecontrol.htm.

[16] C. R. Helmrich, A. Henkel, T. Hinz, A. Wieckowski, B. Bross, and D. Marpe, "Finalization of VVenC's Screen Content Detector and Two-Pass Rate Control Using Pre-Filtering Statistics," accepted into *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Kuala Lumpur, Oct. 2023.

[17] 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.

[18] JVET and Fraunhofer HHI, "VVCSoftware_VTM," *Gitlab repository*, Apr. 2023. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.

[19] Blender Foundation, "Tears of Steel," open movie, 2.35:1 aspect ratio, Creative Commons Attrib., 2012. https://mango.blender.org/download. SDR version from http://forum.doom9.org/showthread.php?t=175776.

[20] H. Miyagawa and K. Peña, "Bringing 4K and HDR to Anime at Netflix with Sol Levante," *Netflix,* 2020. https://netflixtechblog.com/bringing-4k-and-hdr-to-anime-at-netflix-with-sol-levante-fa68105067cd, source http://download.opencontent.netflix.com/?prefix=SolLevante.

[21] 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.

[22] 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.

[23] 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

[24] 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 Sympos. (PCS)*, San Jose, pp. 247*ff.*, Dec. 2022.

[25] 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.

[26] F. Bossen, X. Li, V. Seregin, K. Sharman, and K. Sühring, "VTM and HM common test conditions and software reference configurations for SDR 4:2:0 10-bit video," ITU / ISO / IEC doc. JVET-Y2010, Feb. 2022.

[27] B. Bross, H. Kirchhoffer, C. Bartnik, M. Palkow, and D. Marpe, "AHG4 Multiformat Berlin Test Sequences," ITU doc. JVET-Q0791, Jan. 2020. https://jvet-experts.org All meetings, https://hhi.fraunhofer.de/8kberlin.

[28] Z. Wang, A. Rehman, K. Zheng, J. Wang, and Z. Wang, "SSIM-Motivated Two-Pass VBR Coding for HEVC," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 27, no. 10, pp. 2189–2203, Oct. 2017.