RATE CONTROL FOR HEVC INTRA-CODING BASED ON PIECEWISE LINEAR APPROXIMATIONS

Victor Sanchez

Dept. of Computer Science, University of Warwick, Coventry, UK (vsanchez@dcs.warwick.ac.uk)

ABSTRACT

This paper proposes a rate control (RC) algorithm for intra-coded sequences (I-frames) within the context of block-based predictive transform coding (PTC) that employs piecewise linear approximations of the rate-distortion (RD) curve of each frame. Specifically, it employs information about the rate (R) and distortion (D) of already compressed blocks within the current frame to linearly approximate the slope of the corresponding RD curve. The proposed algorithm is implemented in the High-Efficiency Video Coding (HEVC) standard and compared with the current HEVC RC algorithm, which is based on a trained rate-lambda (R-λ) model. Evaluations on a variety of intra-coded sequences show that the proposed RC algorithm not only attains the overall target bit rate more accurately than the current RC algorithm but is also capable of encoding each I-frame at a more constant bit rate according to the overall bit budget, thus avoiding high bit rate fluctuations across the sequence.

Index Terms**—** HEVC, rate control, intra-coding, R-λ model.

1. INTRODUCTION

Rate Control (RC) is an important tool that allows meeting any channel bandwidth, end-to-end delay or storage requirements by controlling the bit rate of the compressed media. This is achieved by appropriately allocating a bit budget during the encoding process so that a target bit rate is attained with minimum distortion. The HEVC reference software includes two algorithms for RC. One is based on a quadratic model and the mean absolute difference between the original and the re-constructed signals [1, 2]. The other one is based on an R-λ model, which approximates the slope of the rate-distortion (RD) curve of the sequence to be compressed and considers the hierarchical coding structure by distinguishing between intra (I)-frames and inter (P, B)-frames [3]. This R-λ-based RC algorithm has been shown to provide a better performance than the one based on a quadratic model [3]. Other proposals for RC in HEVC include an algorithm based on textured and non-textured rate models [4], methods that allow compressing different regions of a frame using distinct R-λ models [5, 6], and an algorithm that allows for lossless region-of-interest coding with lossy coding of the remaining regions [7].

Despite its good performance, the RC algorithm based on an R-λ model heavily depends on accurately approximating λ, i.e., the slope of the RD curve. If this approximation is accurate, a target bit rate is accurately attained. To this end, this R-λ based RC algorithm includes a mechanism to adaptively update the parameters that approximate λ during the encoding process, starting from a set of parameters that are obtained by using training data. Although this adaptation mechanism avoids computing the value of λ for each sequence *a priori*, a poor adaptation to the true RD characteristics of the sequence may result in inaccurately attaining the target bit rate [6-7]. This issue may be very severe for intra-coded sequences, i.e., the all intra (AI) profile in HEVC [8], as this RC algorithm poorly adapts the model's parameters for I-frames [6,7]. Although recent proposals aim to optimally allocate a target bit rate by exploiting intrinsic characteristics of the encoding method, such as the dependencies among frames induced by motion compensation [9], little work has been done for I-frames.

In this paper, we propose an RC algorithm for intra-coded sequences within the context of block-based PTC. Instead of relying on a trained model that approximates the RD curve, our algorithm approximates the slope of this curve by a sequence of piecewise linear segments, whose computational complexity is very low. These piecewise linear approximations are computed by using actual bit rate and distortion values of previously compressed blocks within the same I-frame, and thus, provide a very accurate representation of the RD curve of each frame. The proposed RC algorithm is implemented in HEVC and evaluated on a variety of sequences using the AI profile at different bit rates. Evaluation results show that the proposed algorithm attains the overall target bit rate more accurately than the current R-λ based RC algorithm of HEVC. More importantly, the proposed RC algorithm avoids high bit rate fluctuations at the frame level across the sequence.

The rest of the paper is organized as follows. Section 2 briefly reviews the current HEVC RC algorithm based on an R-λ model when used for intra-coded sequences. We describe our RC algorithm in Section 3. Section 4 presents the performance evaluations and Section 5 concludes this paper.

2. RATE CONTROL IN HEVC

RC in HEVC aims at determining the quantization parameter (QP) for each coding unit (CU) according to a target bit rate, R_{max}, so that the incurred distortion is minimum. To determine the best set of QPs, an R-λ model is commonly used to approximate the slope of the sequence's RD curve, λ:

$$\lambda = -\frac{\partial D}{\partial R} = \alpha R^{\beta}, \tag{1}$$

where ∂denotes a partial derivative; and α and β are the model's parameters [3]. For intra-coded sequences, the algorithm works at the frame and CU levels. At the frame level, each frame is assigned T_f bits, as follows:

$$T_f = \frac{T_{max} - \hat{T}_{max}}{p}, \tag{2}$$

where T_{max} and $\hat{T}_{max}$ are the overall bit budget and the number of bits spent on the already encoded frames, respectively, and p is the number of remaining frames to be compressed [3]. At the CU level, T_{CU} bits are assigned to each CU, as follows:

$$T_{CU} = \frac{HAD_{CU}}{HAD_f - \widehat{HAD}_f}\omega, \tag{3}$$

where HAD_{CU} denotes the coding cost of the current CU, which is

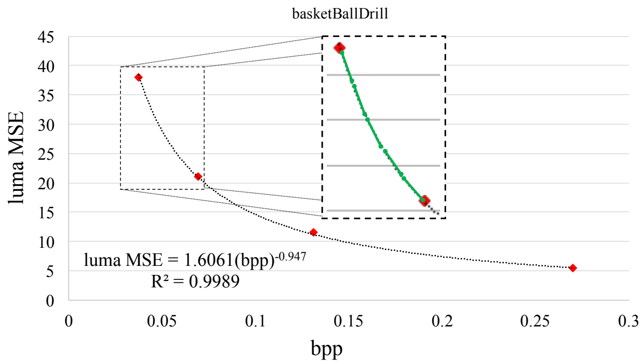


Fig. 1. RD curve of the Y component of sequence *basketBallDrill* approximated by a hyperbolic model with correlation coefficient, R^2 [3]. A section of this curve is also approximated by piecewise linear segments.

calculated by deriving the corresponding sum of absolute Hadamard transformed differences (SATD) [10]; HAD_f and $\widehat{HAD}_f$ denote the coding cost of all the CUs and those already compressed in the current frame, respectively; and ω is the number of bits left in T_f weighted according to the number of CUs to be encoded and the number of bits already spent [3]. The λ value for each CU, denoted by λ_{CU}, is then computed based on $C_{CU} = HAD_{CU}/N_{CU}$, i.e., the cost per pixel of the CU [5]:

$$\lambda_{CU} = \alpha(C_{CU}/R_{CU})^{\beta}, \quad (4)$$

where $R_{CU} = T_{CU}/N_{CU}$ is the target bit rate of the CU and N_{CU} is the number of pixels in the CU. It is important to note that for I-frames, λ_{CU} depends on model's parameters α and β, which remain constant for the entire frame and are only updated after compressing a complete frame, thus potentially adapting poorly to the RD characteristics of the current I-frame. To maintain a low computational complexity, the QP for each CU, denoted by QP_{CU}, is determined linearly by using λ_{CU} [11]:

$$QP_{CU} = 4.2005 ln\lambda_{CU} + 13.7122 \quad (5)$$

A consistent quality is then achieved by clipping all λ_{CU} and QP_{CU} values in a narrow range [3].

3. PROPOSED RC ALGORITHM

As shown in [3], the RD curve of most camera-captured sequences can be approximated by a hyperbolic model; the R-λ model for RC in HEVC is based on this premise. Fig. 1 shows the curve representing the RD relationship for the luma (Y) component of the *basketBallDrill* sequence. Note that although a hyperbolic model can indeed approximate the curve very accurately, small linear piecewise segments can also be used to approximate it. Our proposed RC algorithm uses this observation to approximate λ at the CU level. The algorithm works at the frame and CU levels. At the frame level, T_f is computed according to Eq. (2). At the CU level, after computing T_{CU} according to Eq. (4), the algorithm computes λ_{CU} using piecewise linear approximations. Let us denote the coding cost and target bit rate of the i^{th} CU by HAD_{CU_i} and R_{CU_i}, respectively. Our algorithm aims to find, within the current frame, two already compressed CUs, a and b, so that:

$$(1-\rho)HAD_{CU_i} \le HAD_{CU_a} \le (1+\rho)HAD_{CU_i}, \quad (6)$$

$$(1-\sigma)R_{CU_i} \le \hat{R}_{CU_a} \le R_{CU_i}, \quad (7)$$

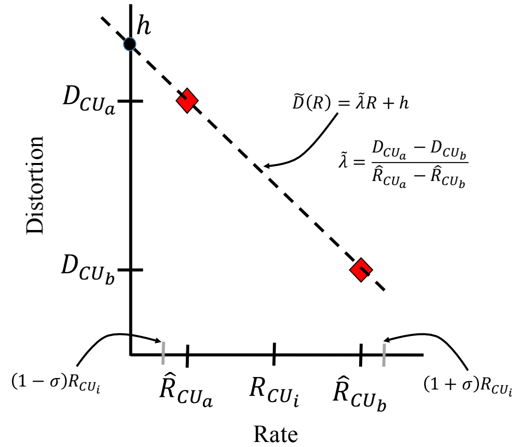


Fig. 2. Approximation of $\tilde{D}(R)$ over the linear segment defined by the actual rates and distortions of the two already compressed CUs, a and b.

$$(1-\rho)HAD_{CU_i} \le HAD_{CU_b} \le (1+\rho)HAD_{CU_i}, \quad (8)$$

$$(1+\sigma)R_{CU_i} \ge \hat{R}_{CU_b} \ge R_{CU_i}, \quad (9)$$

where $\rho \ll 1$ and $\sigma \ll 1$ are small constants, and $\hat{R}_{CU_n}$ denotes the actual bit rate of the n^{th} already compressed CU. In other words, our algorithm aims to find an already compressed CU, i.e., CU_a, whose coding cost is very similar to that of the current CU, and whose actual bit rate is lower than (but very similar to) the target bit rate, R_{CU_i} [see Eq. (6) and (7)]. Our algorithm also aims to find another already compressed CU, i.e., CU_b, whose coding cost is also very similar to that of the current CU, and whose actual bit rate is higher than (but very similar to) the target bit rate, R_{CU_i} [see Eq. (8) and (9)]. Moreover, the distortion of CU_a and CU_b, denoted by $D_{CU_a}$and D_{CU_b}, respectively, must satisfy:

$$D_{CU_a} > D_{CU_b}. \quad (10)$$

The condition in Eq. (10) guarantees that CU_a, which is encoded at a bit rate lower than that of CU_b, is indeed reconstructed at a lower quality than that of CU_b. This is expected given the conditions in Eq. (6)-(9) and the fact that both CUs, a and b, have similar coding costs. These ideas are illustrated in Fig 2, where one can see that if σ in Eq. (7) and (9) is constrained to be very small (i.e., $\sigma \ll 1$), the actual bit rates and distortions of the two already compressed CUs, a and b, form a linear segment defined in the interval $[\hat{R}_{CU_a}, \hat{R}_{CU_b}]$. Based on this constraint, we can then employ a piecewise linear approximation to approximate a distortion function $\tilde{D}(R)$ over this interval, as follows:

$$\tilde{D}(R) = \tilde{\lambda}R + h, \quad (11)$$

where $\tilde{\lambda}$ is the slope of the linear interval and h is the *distortion* intercept (i.e., where the line crosses the *distortion* axis). Slope $\tilde{\lambda}$ can then be easily computed as:

$$\tilde{\lambda} = \frac{D_{CU_a} - D_{CU_b}}{\hat{R}_{CU_a} - \hat{R}_{CU_b}}. \quad (12)$$

Since the target bit rate for the i^{th} CU is within the interval $[\hat{R}_{CU_a}, \hat{R}_{CU_b}]$ [see Eq. (7) and (9)], slope $\tilde{\lambda}$ in Eq. (12) also represents the slope of the RD curve of the i^{th} CU, which is denoted by $\tilde{\lambda}_{CU_i}$. After computing $\tilde{\lambda}_{CU_i}$, QP_{CU_i} is computed using Eq. (5). Quality consistency is achieved by clipping all QPs in a narrow range. Specifically, QP_{CU_i} is guaranteed that $QP_{min} \le QP_{CU_i} \le$

Table 1. Characteristics of the test sequences and BRE (%) values attained by each RC algorithm.

Sequence	Frame size (pixels)	fps	Color format	Current			Proposed		
				|Average|	Min	Max	|Average|	Min	Max
class Screen Content									
flyingGraphics	1920×1080	60	RGB	0.716	-4.157	0.027	**0.004**	**-0.002**	**0.017**
map	1280×720	60	RGB	0.020	0.004	0.049	**0.002**	**0.000**	**0.004**
missionControlClip3	1920×1080	60	4:2:0	2.130	-12.840	**0.007**	**0.149**	**-1.848**	**0.007**
class F									
slideShow	1280×720	20	4:2:0	1.012	-1.568	2.695	**0.398**	**-0.725**	**0.178**
slideEditing	1280×720	30	4:2:0	0.027	-0.006	0.086	**0.025**	**0.001**	**0.070**
chinaSpeed	1024×768	30	4:2:0	0.040	0.005	0.112	**0.013**	**0.001**	**0.044**
class B									
kimono	1920×1080	24	4:2:0	0.018	0.003	0.049	**0.002**	**0.000**	**0.004**
parkScene	1920×1080	24	4:2:0	0.005	0.002	0.009	**0.002**	**-0.001**	**0.007**

QP_{max}, where QP_{min} and QP_{max} are the minima and maximum QPs allowed by the encoder, respectively. Additionally, QP_{CU_i} is clipped in a narrow range determined by the QPs of the previously coded spatially adjacent CUs:

$$QP_{avg} - \phi \leq QP_{CU_i} \leq QP_{avg} + \phi, \quad (13)$$

where QP_{avg} is the average QP of the three CUs spatially adjacent to the i^{th} CU, located above, to the left, and above and to the left; and $\phi = 4$ is a constant that allows accommodating for coding cost differences between the i^{th} CU and the spatially adjacent CUs.

After compressing the current frame, the actual bit rate, the actual distortion, and the coding cost values of the initial CUs of the current frame are used to help to encode the initial CUs of the following frame in cases where not enough previously coded CUs are available in the following frame that satisfy Eq. (6)-(10). It is important to mention that the proposed RC algorithm requires simple operations to approximate $\tilde{\lambda}_{CU_i}$ [see Eq. (12)], which is advantageous to keep encoding times short. Moreover, it does not require any RD models to be trained, as the RD curve of each frame is approximated as CUs are sequentially encoded.

4. PERFORMANCE EVALUATION

The proposed RC algorithm is implemented in the reference software HM16.8 [12] and compared to the current RC algorithm (based on an R-λ model), using the AI profile with a largest CU (LCU) of 64×64 samples. Table I tabulates the characteristics of the 8-bit precision test sequences [13], each of which is encoded at 15 different target bit rates ranging from 5 to 500 Mbps. For the Screen Content (SC) sequences, the Screen Content Coding (SCC) tools are used with the AI-SCC profile [14, 15].

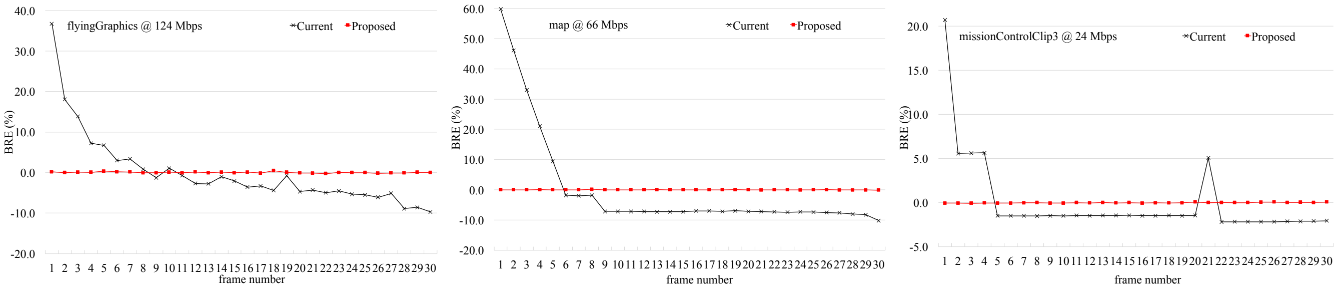
The accuracy of both algorithms is evaluated in terms of the bit rate error (BRE - %). The BRE indicates how accurately the target bit rate is attained; negative numbers indicate underspending, while positive numbers indicate overspending the bit budget. Average absolute BRE values of both algorithms over all evaluated target bit rates are tabulated in Table I, along with the maximum and minimum BRE values attained. Note that our proposed RC algorithm attains the target bit rate more accurately than the current RC algorithm. This is particularly evident for SC sequences, for which the current RC algorithm underspends the overall bit budget by as much as 12.84% (*missionControlClip3*). Note that for *class B* sequences, which are camera-captured sequences, the current RC algorithm is also very accurate. This accuracy level is expected, as the model's parameters currently used by HEVC are computed based on a large set of training camera-captured sequences.

The advantages of our algorithm are more evident in Fig. 3, which plots BREs on a per-frame basis after encoding the first 30 frames of some of the test sequences at high (Fig. 3a-f) and low compression ratios (Fig. 3g-l). These plotted bit rates are the ones at which both algorithms attained similar overall BREs. Let us recall that both algorithms tend to assign an equal bit budget to all I-frames, according to Eq. (2). Therefore, a constant bit rate is expected across all frames. Note that the proposed RC algorithm indeed maintains a very constant bit rate across all frames (i.e., BREs close to 0%), avoiding the high bit rate fluctuations and inaccuracies (i.e., high/low BREs) observed in the results attained by the current RC algorithm. Let us take as an example sequence *flyingGraphics* encoded at 398 Mbps (see Fig. 3g), which shows important differences in the way the two algorithms distribute the bit budget across the frames. The current RC algorithm underspends the bit budget of the initial frames, thus encoding these frames at a lower bit rate than the target one. This algorithm then compensates any underspending by overspending the bit budget of the remaining frames, thus encoding these later frames at a higher bit rate than the target one. The proposed algorithm, on the other hand, accurately encodes all frames at the target bit rate, thus minimizing any overspending or underspending. Fig. 4 shows a reconstructed region of the green (G) component of frame 1 of this sequence. Note the lower reconstruction quality attained by the current RC algorithm due to underspending the bit budget. A similar behavior is observed for the *missionControlClip3* sequence encoded at 298 Mbps (see Fig 3i), while the opposite behavior is observed in Fig. 3a-3f, where the current RC algorithm overspends the bit budget for the initial frames. Note that for the *parkScence* (10 and 150 Mbps) and *chinaSpeed* (35 Mbps) sequences, the current RC algorithm is also very accurate (see Fig. 3f, 3k and 3l). This suggests that the current model's parameters in HEVC accurately model the RD characteristics of these sequences.

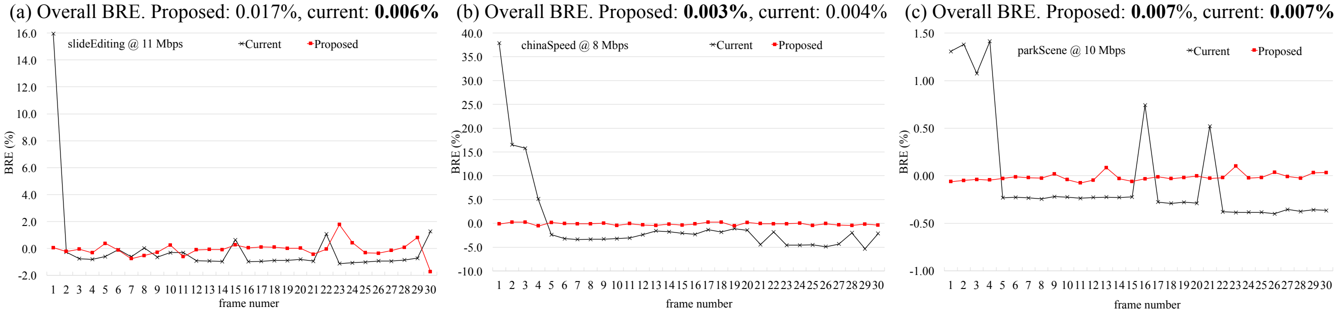
The average increase in encoding time incurred by our RC algorithm for the test sequences is only 0.05% and is mainly due to the search for previously coded CUs that satisfy Eq. (6)-(10).

5. CONCLUSIONS

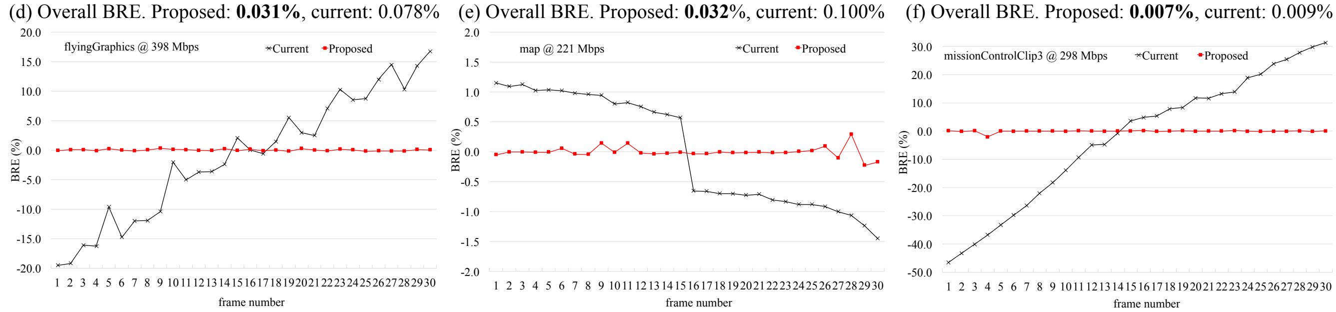
This paper presented an RC algorithm for intra-coded sequences within the context of block-based PTC. The algorithm approximates the RD characteristics of each frame by using piecewise linear approximations that consider the actual rate and distortion values of previously coded blocks within the same frame. Therefore, the proposed algorithm does not require trained models to approximate these RD characteristics. Performance evaluations in HEVC over various sequences encoded at different target bit rates using the AI and AI-SCC profiles confirm the higher accuracy of the proposed RC algorithm, and the more constant bit rate achieved, on a per-frame basis, compared to the current HEVC RC algorithm based on an *R*-λ model.



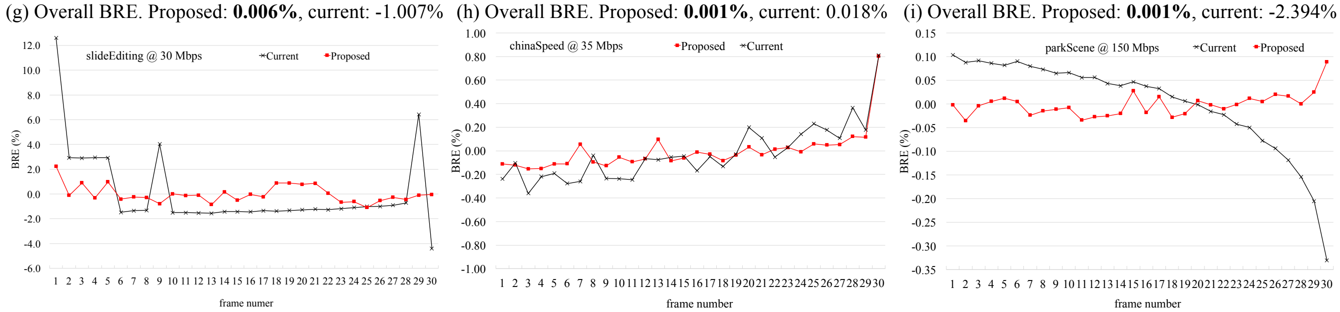
(a) Overall BRE. Proposed: 0.017%, current: **0.006%** (b) Overall BRE. Proposed: **0.003%**, current: 0.004% (c) Overall BRE. Proposed: **0.007%**, current: **0.007%**



(d) Overall BRE. Proposed: **0.031%**, current: 0.078% (e) Overall BRE. Proposed: **0.032%**, current: 0.100% (f) Overall BRE. Proposed: **0.007%**, current: 0.009%



(g) Overall BRE. Proposed: **0.006%**, current: -1.007% (h) Overall BRE. Proposed: **0.001%**, current: 0.018% (i) Overall BRE. Proposed: **0.001%**, current: -2.394%



(j) Overall BRE. Proposed: **0.018%**, current: 0.086% (k) Overall BRE. Proposed: **0.002%**, current: 0.021% (l) Overall BRE. Proposed: **0.001%**, current: 0.005%

Fig. 3. Per-frame BREs for the proposed (red) and current (black) RC algorithms using the AI profile at (a)-(f) high and (g)-(l) low compression ratios.

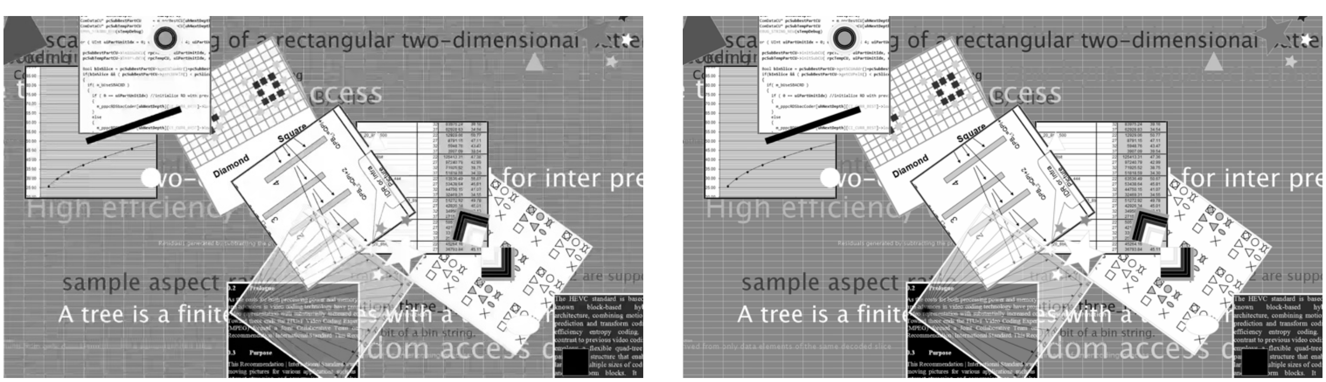


Fig. 4. Reconstructed G component of frame1of *flyingGraphics* at a 398 Mbps target bit rate. PSNR: 31.82 dB (left - current) and 42.52 dB (right - proposed).

6. REFERENCES

[1] M. Naccari and F. Pereira, "Quadratic modeling rate control in the emerging HEVC standard," in *Proc. Picture Coding Symp. (PCS)*, pp. 401-404, 2012.

[2] S. Ma, J. Si, and S. Wang, "A study on the rate distortion modeling for high efficiency video coding," in *Proc. IEEE Int. Conf. on Image Poecessings (ICIP)*, September 2012, pp. 181-184.

[3] B. Li, H. Li, and L. Li, "Lambda-domain rate control algorithm for High Efficiency Video Coding," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3841-3854, 2014.

[4] B. Lee, M. Kim, and T. Nguyen, "A frame-level rate control scheme based on texture and non-texture rate models for High-Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1-14, 2013.

[5] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu, "Region-of-interest based rate control scheme for high-efficiency video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing* (*ICASSP*), pp. 7338-7342, May 2014.

[6] K. Perez-Daniel, and V. Sanchez, "Luma-aware multi-model rate-control for HDR content in HEVC," in *Proc. IEEE Int. Conf. on Image Poecessings (ICIP)*, 2017, in presss.

[7] V. Sanchez, F. Aulí-Llinàs, R. Vanam and J. Bartrina-Rapesta, "Rate control for lossless region of interest coding in HEVC intra-coding with applications to digital pathology images," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing (ICASSP)*, pp. 1250-1254, 2015.

[8] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[9] A. Fiengo, G. Chierchia, M. Cagnazzo and B. Pesquet-Popescu, "Rate allocation in predictive video coding using a convex optimization framework," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 479-489, Jan. 2017.

[10] Y. Kim, D. Jun, S. Jung, and J. Choi, "A fast intra prediction method using Hadamard transform in high-efficiency video coding," in *Proc. Visual Information Processing and Communication III*, pp. 270-280, February 2012.

[11] B. Li, D. Zhang, H. Li, and J. Xu, "QP determination by lambda value," in *JCT-VC I0426, 9th meeting of Joint Collaborative Team on Video Coding of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, Geneva, CH, 2012

[12] HM116.8 software. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.8/

[13] H. Yu, R. Cohen, K. Rapaka, and J. Xu, *Common Test Conditions for Screen Content Coding*, document JCTVC-T1015, Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Feb. 2015.

[14] V. Sanchez, F. Aulí-Llinàs, and J. Serra-Sagristà, "DPCM-Based *Edge Prediction for Lossless Screen Content Coding in HEVC*," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 497-507, Dec. 2016.

[15] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50–62, Jan. 2016.