Adaptive Video Encoding for Time-Constrained Compression and Delivery

André Seixas Dias
BBC Research and Development
London, UK

Saverio Blasi
BBC Research and Development
London, UK

Marta Mrak
BBC Research and Development
London, UK

Shenglan Huang
Queen Mary University of London
London, UK

Ebroul Izquierdo
Queen Mary University of London
London, UK

***Abstract*— Some applications require video content to be encoded and uploaded to a remote destination in a fixed time. This paper proposes a novel approach to address this challenge, based on online adaptation of compression parameters to control both encoding and uploading time. In particular, an algorithm to accurately predict the encoding time and bit-rate resulting from using a given quantisation parameter to encode the next frames in the sequence is proposed. This is used to drive decisions during the encoding process, to maintain the cumulative time needed to encode and transmit the video sequence within the constraints. Experimental evaluation shows that when using the proposed method, time constraints can be met with high accuracy under a variety of different target times and bandwidth conditions.**

***Keywords*— HEVC, rate-control, encoding time estimation**

I. INTRODUCTION

Many practical applications require video content to be compressed and transmitted to a remote location within pre-defined time constraints. As an example, journalists in the field often need to contribute high quality video content to a central repository with limited transmission bandwidth. In this case, they want the video content to be not only efficiently compressed, but also uploaded to a central repository, so that it can reach the intended destination in a limited amount of time, while still targeting the maximum possible level of quality.

The state-of-the-art video coding standard, H.265/High Efficiency Video Coding (HEVC) [1], provides notable video compression performance improvements with respect to the predecessor H.264/Advanced Video Coding (AVC) [2][3]. However, since more complex tools are used, higher encoding times are needed. Therefore, in order to meet specific time constraints, encoding time needs to be taken into account along with the uploading time.

Speed up techniques can be used to control the encoder complexity [4]. Conversely, rate-control methods are widely available to control the output bit-rate after compression, which has a direct impact on the uploading time. Unfortunately, using these mechanisms in a decoupled manner may not be sufficient. Controlling only encoding complexity ensures no restrictions on uploading time, which can be significant for high bit-rates. Conversely, fixing a target bit-rate gives no guarantees in terms of encoding complexity. Rate control algorithms typically work by adaptively changing the Quantisation Parameters (QP) used in the encoding process. These QP variations have a non-negligible impact on the encoding time.

A new approach is proposed in this paper to encode video content taking into account time constraints both for encoding and uploading. The approach works by adaptively changing the QP during the encoding process. These changes are driven by accurate estimation algorithms which estimate the time as well as number of bits necessary for encoding a portion of the video sequence with a specific QP. The encoder can then select the lowest QP that satisfies the imposed time constraints, consequently providing the maximum quality within the constraints.

The rest of this paper is organised as follows. A literature review of background work is first presented in Section 2. The proposed approach is then described in Section 3. Section 4 presents a comprehensive experimental evaluation, and Section 5 concludes the paper.

II. BACKGROUND WORK

The core of the framework proposed in this paper is based on accurate bit-rate and encoding time estimation techniques. Rate-control is an essential tool for most practical video coding applications. The reference software of the AVC standard (Joint Model, JM) provides a method [5] based on Mean Absolute Difference (MAD) of residuals of previously encoded blocks. This is used to compute the quantisation step using a quadratic Rate Quantisation (RQ) relationship [6]. The same model was proposed for HEVC [7]. Methods were also proposed based on the relationship between rate and Lagrangian multiplier λ used for rate distortion optimisation. Rate-distortion costs are typically computed as $J = D + \lambda R$, where D is the distortion between original and reconstructed content and R is the corresponding rate [8][9]. λ domain rate control algorithms model a relationship between rate and λ to select the λ value, which is then mapped to a quantisation step [10]. Finally, another group of rate control algorithms rely on the relationship between rate and percentage of non-zero coefficients after quantisation, ρ. An approach based on a quadratic ρ domain rate model was proposed in the context of HEVC [11].

Higher compression performance typically comes at the cost of more encoder complexity [12]. For this reason, many techniques have been studied to reduce the encoding

The work described in this paper has been conducted within the project COGNITUS. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant No 687605.

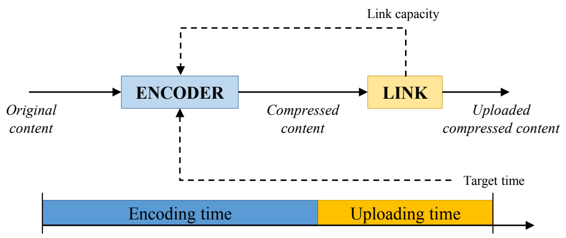

Fig. 1. Scheme of the proposed approach.

complexity, ideally with low impact in rate-distortion performance. An algorithm for rate-distortion-complexity control is proposed in [13] by defining a complexity budget allocation scheme and using adaptive searching algorithms. This algorithm was proposed in the context of a practical implementation of an AVC encoder [14]. The method in [4] proposes a combination of medium and fine granularity encoding time control algorithms to keep the encoding time below a given target time for each group of pictures in an HEVC encoder. In this paper, time restrictions are considered not only to encoding but also to uploading, and therefore the bit-rate also needs to be taken into account. Moreover, differently than previously proposed methods, encoding time variations due to the QP are also taken into consideration.

III. Proposed Approach

Fig. 1 shows the general scheme of the proposed approach. The system is capable of meeting a target time by adapting the QP values while compressing the content so as to control both encoding and uploading time. In [15], it is reported that for both AVC and HEVC reference software implementations, encoding the same sequence with different QPs can lead to an encoding time increase of almost 30%. Even higher impact of the QP on encoding time was obtained when testing a practical HEVC video encoder implementation. Going from a QP of 45 to a QP of 15 resulted in up to 5 times higher encoding times when compressing content with 1920 × 1080 spatial resolution.

The QP in HEVC can be changed during the encoding with values can be changed within a range spanning from 0 to 51. Higher QP values result in higher compression ratios at the cost of lower quality. The proposed algorithm works by varying the base QP on a Structure of Pictures (SOP) basis. Frames in HEVC are typically arranged in SOP layers. The SOP structure in HEVC defines specific parameters for each layer, such as the encoding order (which may be different than their display order, referred to as the Picture Order Count, POC), the reference frames used for inter-prediction, and the QP offsets (which are defined values added to the sequence-level base QP, to better distribute bits within the SOP). Without loss of generality, a fixed SOP structure of 8 frames with 4 SOP layers was used in this paper, in accordance with the Random Access configuration defined in the JCT-VC Common Test Conditions [16]. The proposed algorithm works by varying the base QP for each SOP, so that information extracted from frames belonging to each SOP layer can be used to predict encoding time and bits spent in the next SOP, as explained in the rest of this paper. Since it is undesirable to introduce abrupt variations from SOP to SOP, QPs are limited to a variation of +/-5 between consecutive SOPs.

Denote $\overline{T}_{tot}$ as the total target time, defined as the available time from the start of the encoding process to the moment the content must reach the destination. Assume that uploading happens under a network channel with a fixed bandwidth equal to W_{link}. Assume that the sequence is composed of a total number of N SOPs, where each SOP is referred to as SOP n, with $n \in \{0, \ldots, N-1\}$. The proposed approach operates by assigning a specific target time to each SOP. Before starting encoding the first SOP in the sequence, a uniform distribution of time is assumed among SOPs, so the first SOP, referred to as SOP 0, is assigned a target time $\overline{T}_{tot,0} = \overline{T}_{tot} / N$. A predefined initial QP value, denoted as q_0, is used to encode this SOP. When the encoding process of SOP 0 is complete, a total number of bits B_0 is produced in an encoding time $T_{enc,0}$. During the encoding, relevant information is also collected so that the encoder can perform decisions on the QP to use on the next SOP, as illustrated in the rest of this section. Considering the transmission of the encoded bit-stream, the total time necessary to encode and upload the first SOP, $T_{tot,0}$, is then obtained as:

$$T_{tot,0} = T_{enc,0} + \frac{B_0}{W_{link}} . \quad (1)$$

The target time for the next SOP, $\overline{T}_{tot,1}$, can then be refined by computing how much time is left (with respect to the total target $\overline{T}_{tot}$) using the actual time spent on the first SOP. Based on these estimates and on $\overline{T}_{tot,1}$, the encoder selects a new value q_1, so that the total time for encoding and uploading SOP 1 is as close as possible to the target. Generalising this process:

1. Before encoding SOP n, the SOP target time is computed:

$$\overline{T}_{tot,n} = \frac{\left(\overline{T}_{tot} - \sum_{i=0}^{n-1} T_{tot,i}\right)}{N-n} . \quad (2)$$

2. A set of QP values is considered, namely $\mathbf{Q} = \{q_{n-1} - 5, q_{n-1} - 4, \ldots, q_{n-1} + 5\}$. For each valid q in $\mathbf{Q}$, information extracted from SOP n–1 is used to estimate the encoding time $\tilde{T}_{enc,n}(q)$ and the number of bits $\tilde{B}_n(q)$, needed to encode SOP n with q.
3. A prediction of the total time necessary for encoding and uploading with q is computed as:

$$\tilde{T}_{tot,n}(q) = \tilde{T}_{enc,n}(q) + \frac{\tilde{B}_n(q)}{W_{link}} . \quad (3)$$

4. The QP q_n for SOP n is then selected to maximise quality as:

$$\min\{q : q \in \mathbf{Q}, \tilde{T}_{tot,n}(q) \le \overline{T}_{tot,n}\} . \quad (4)$$

5. SOP n is encoded using q_n as base QP. The actual encoding time $T_{enc,n}$ and number of bits spent on the SOP, B_n, are extracted, and the actual total time is computed as:

$$T_{tot,n} = T_{enc,n} + \frac{B_n}{W_{link}} . \quad (5)$$

Step 2 in this algorithm relies on techniques for estimating the encoding time and bit-rate on a SOP level for a given QP value. In a typical HEVC scheme, the encoder splits the current frame into a number of square blocks, referred to as Coding Units (CUs), and computes a prediction for each CU independently. The prediction is subtracted from the original block to obtain the residual signal. The encoder can then further partition the residual signal into smaller square blocks of samples for luma and chroma components, referred to as Transform Blocks (TBs). Each TB is transformed, quantised and entropy coded with CABAC [17]. Assuming the encoder is given the same inputs, prediction and transform are performed independently from the QP value. The quantisation is then performed using the QP value, but its complexity is the same for any QP value: the actual time spent for quantisation is not affected by the QP because the encoder will still need to process each transformed coefficient. On the contrary, the entropy encoding complexity is drastically affected by the QP value: the more zero-valued coefficients are present in the TB, the less operations are necessary to produce CABAC codes [17]. In the ideal case where all quantised coefficients are zero, no entropy coding needs to be performed, leading to even larger complexity savings, as in this case no inverse transform and inverse quantisation are needed. The entropy encoding process is therefore the main factor for encoding time variations on a given TB when using different QPs, where such variations are directly correlated with the ratio of non-zero levels over the total number of coefficients in the TB. For this reason, the first step for obtaining a reliable estimation of the encoding time for a given QP is to compute the ratio of non-zero coefficients over the total number of coefficients, obtained with a QP value on a TB. This ratio is hereafter denoted as ρ.

The scalar quantisation process in a typical HEVC encoder for a given TB of size $A \times A$ can be described by:

$$v_{i,j} = \mathrm{sgn}(c_{i,j}) \cdot \left\lfloor \frac{|c_{i,j}|}{\delta_q} + d \right\rfloor , \tag{6}$$

where $c_{i,j}$ and $v_{i,j}$ denote the transform coefficient and the resulting coefficient level, respectively, in position (i, j) in the TB, δ_q denotes the quantisation step associated with the QP value q and d is an offset used for rounding. Hence, from (6), a given coefficient $c_{i,j}$ will result in quantized coefficient level $v_{i,j}$ different from zero if it satisfies the following condition:

$$|c_{i,j}| \geq (1-d) \cdot \delta_q . \tag{7}$$

Using (7), the number of non-zero levels in the TB obtained with a given q can be computed as:

$$\Psi(q) = \frac{k_q}{A \times A} , \tag{8}$$

where k_q is the number of coefficients $c_{i,j}$ that satisfy (7) when quantised with q. The expressions (6) - (8) can be used to compute the number of non-zero levels for each TB for all QPs in $\mathbf{Q}$. In practice, the encoder does not need to perform $A \times A$ comparisons for each QP value: if a coefficient $c_{i,j}$ is quantised to 0 for a given QP, it will also be for any higher QP. This significantly reduces the number of comparisons necessary for each TB, reducing the impact of the computation of Ψ in the overall encoding time. Denote then as M the number of TBs tested within the current frame. Denote as $k_{q,m}$ the number of coefficients that satisfy (7) for QP q on TB m of size $A_m \times A_m$. The non-zero level ratio at the frame level, $\rho(q)$, is given by:

$$\rho(q) = \frac{\sum_{m=0}^{M-1} k_{q,m}}{\sum_{m=0}^{M-1} {A_m}^2} . \tag{9}$$

The relation in (9) can be computed for any value of q, regardless of the actual QP used to encode the frame. It is important to note that the resulting $\rho(q)$ is only an estimate of the ratio of non-zero levels that would be obtained if the frame was actually encoded with q. This is due to the fact that when using different QPs, different reconstruction samples become reference samples for subsequent predictions, leading to different prediction signals. Nonetheless, an analysis was performed to confirm the estimation accuracy, especially when q is limited to +/- 5 changes with respect to the actual QP.

Assume then that a given frame at a given SOP layer l, in a given SOP n is being encoded and denote as $q_{n,l}$ the QP value being used to encode this frame (obtained as base QP used in SOP n, q_n, plus the QP offset corresponding to the SOP layer l). Denote as $t_{\mathrm{EC}}(q_{n,l})$ the total time necessary for entropy encoding of quantised transform coefficients in the frame. Similarly, denote as $t_{rem}(q_{n,l})$ the total remaining time necessary for encoding the frame (measured from the instant the frame starts encoding, to the instant the last bit is written in the bit-stream). Denote as $t_{enc}(q_{n,l}) = t_{\mathrm{EC}}(q_{n,l}) + t_{rem}(q_{n,l})$ the total encoding time of the frame. For each QP value q in $\mathbf{Q} = \{q_{n-l} - 5, q_{n-l} - 4, ..., q_{n-l} + 5\}$, the encoder can compute $\rho(q)$ using (9). The encoder also computes the actual average ratio of non-zero levels over the total obtained while encoding with $q_{n,l}$, denoted as $\rho(q_{n,l})$. Assuming that t_{EC} and ρ are linearly correlated, for a given q, the following is computed

$$\tilde{t}_{\mathrm{EC}}(q) = \frac{t_{\mathrm{EC}}(q_{n,l})}{\rho(q_{n,l})} \cdot \rho(q) , \tag{10}$$

where $\tilde{t}_{\mathrm{EC}}(q)$ is the estimated time for entropy encoding the quantised coefficients in the frame obtained with a QP value of q. Finally, the total estimated encoding time for the frame when encoded with a QP value q can be computed as:

$$\tilde{t}_{enc}(q) = \tilde{t}_{\mathrm{EC}}(q) + t_{rem}(q_{n,l}) . \tag{11}$$

In addition to the encoding time estimation of the next SOP for different QPs, another essential element of the proposed algorithm is the estimation of the number of bits necessary to encode SOP n using a QP value of q, denoted as $\tilde{B}_{\mathrm{n}}(q)$. Let $b(q)$ denote the total number of bits needed to encode a given frame with a given QP value of q in SOP n. Again, a linear relationship is assumed between b and ρ, and thus for a given value of q, the following can be computed:

$$\tilde{b}(q) = \frac{b(q_{n,l})}{\rho(q_{n,l})} \cdot \rho(q) . \tag{12}$$

ISBN 978-90-827970-1-5

This process is then used to perform encoding time and bit estimations at a SOP level. Denote as L the number of SOP layers in a SOP. Denote as F_l the number of frames in each layer l. Using (11) and (12), an estimation of the encoding time and bits for each frame in the SOP, for each QP value in **Q** can be computed. Denote as $t_{enc,l}(q)$ the average total estimated encoding time computed for all frames in the SOP belonging to SOP layer l. Finally, the total estimated encoding time for the whole SOP encoded with a QP value of q can be computed as:

$$\widetilde{T}_{enc,n}(q) = \sum_{l=0}^{L-1} F_l \cdot t_{enc,l}(q)\,. \tag{13}$$

Similarly the average number of bits for all frames in a given SOP layer l is computed, denoted as $\widetilde{b}_l(q)$. The estimated number of bits to encode SOP n with a QP value of q is then:

$$\widetilde{B}_n(q) = \sum_{l=0}^{L-1} F_l \cdot \widetilde{b}_l(q)\,. \tag{14}$$

The values obtained using (13) and (14) can be used in (3) to obtain an estimate of the total time necessary for encoding and uploading the SOP with q.

IV. Experimental Evaluation

The method proposed in this paper was tested to evaluate its ability to accurately meet conditions imposed by different time and bandwidth constraints under a variety of test conditions, selected to verify its effectiveness in various possible use cases. The test material includes content with spatial resolutions of 1280 × 720, 1920 × 1080 and 3840 × 2160 at 24, 50 or 60 fps with a duration of 10 seconds. All sequences are either publicly available or belong to the Joint Collaborative Team on Video Coding (JCT VC) Common Test Conditions (CTC) [16]. To simulate challenging conditions, bandwidths of 128, 256 and 512 kbps were considered. Relevant total target times (encoding + uploading) were selected according to the spatial and temporal resolutions of the content being encoded and transmitted.

The proposed method works by balancing encoding and uploading time and, as such, it is critical that a realistic encoding time is achieved by the chosen encoder. This is to avoid completely unbalancing the distribution of time, making the uploading time marginal. Therefore, a practical HEVC software encoder was used as basis for the implementation and experimental evaluation. In particular the Turing codec [18] was selected as base for implementation, as this is an open source HEVC software encoder containing fast encoding presets and software optimisations that are essential in practical video compression applications [19]. The method was implemented on top of the Turing codec (version 1.1). All tests were run using the fast speed preset in single thread mode on Intel Xeon X3450 CPUs (2.67 GHz) with 8 GB of RAM.

First, the accuracy of the estimation steps (both for encoding time and number of bits) was evaluated. This is presented in terms of the relative error between the estimation and the real observed values, computed as:

$$err = \frac{abs(X - X')}{X} \times 100\ [\%], \tag{15}$$

where X and X' denote the real and estimated values, respectively (either in terms of number of bits, or encoding time, in seconds). Table 1 shows these results. The table reports the average error obtained for each SOP. Also, the overall accuracy is reported, obtained by computing the total target encoding time and total target number of bits, and comparing this with the actual encoding time and total number of bits. In terms of average SOP-level accuracy, the method is capable of meeting the target for each SOP with an estimation error of around 9.9 % (for number of bits) and 9.8 % (for encoding time). The error varies slightly depending on the resolution and frame-rate of the sequences, but never exceeds 15 %. On the other hand, the overall estimation errors are much lower. The method is able to compensate online for the errors produced in previous SOPs. The overall error is therefore very low, on average equal to 1.2 % (for number of bits) and 2.2 % (for encoding time).

Table 1 shows that the estimation steps used by the proposed algorithm are capable of accurately meeting the targets and can therefore by used in the overall algorithm illustrated in the previous section. The general performance of the proposed algorithm was therefore tested and is summarised in Table 2. In this table, the ability to accurately meet the total target time (comprising encoding time and uploading time) is measured, using the relative error between the specified total target time and the actual total, as in (15). The proposed method is also compared with using a fixed QP to encode the whole length of each test sequence. For this purpose, all selected test sequences were encoded using all QPs ranging from 11 to 45, in a fixed QP configuration. For a given target time and uploading link bandwidth, the QP that best matched the total target time was considered. It is important to note that such ideal fixed QP value cannot be determined before the encoding process is completed, and as such, the results obtained with fixed QP are only presented for comparison purposes. They represent a theoretical limit in terms of the best quality that can be achieved when encoding a given sequence.

The table shows that the method is capable of achieving the target time with low accuracy error. Average error of 0.3 % was obtained, with maximum error of 0.5 % in the case of 1920 × 1280 sequences at 24 fps when considering a bandwidth of 128 kbps. When using instead a fixed QP encoding, at best the target time can be achieved with an error of 3.6 %, with errors as high as 6.9 % in some cases. It is worth noticing that, in the fixed-QP scenario, the reported errors correspond to the best case scenario where the encoding is performed using the optimal QP. In practice, users would need to blindly select a QP, which could lead to a total time very different than the target time. On the other hand, encoding with fixed QP ensures that, by definition, content is compressed with the best possible quality. Nonetheless, Table 2 reports the difference between PSNR obtained with fixed-QP configuration and that obtained by the proposed method, referred to as Δ in the table. The results show that the method achieves a quality very close to the theoretical limit. An average Δ of 0.55 dB is obtained, with maximum Δ value of 0.85 dB in specific cases.

ISBN 978-90-827970-1-5

V. Conclusions

This paper proposed an approach for adapting video compression in order to meet a total target time, comprising encoding and uploading under fixed bandwidth conditions. The method is based on estimation steps to predict the encoding time and number of bits obtained on a given group of frames while encoding with a different QP from the one actually being used. This allows the approach to select the QP accordingly in order to meet the target time and at the same time maximise the output quality. The method was extensively tested on a variety of content under different conditions. The experimental evaluation shows that the total target time can be successfully met, with an average estimation error of 0.3 %, while at the same time obtaining output qualities very close to the theoretical limits. As future research work, more complex prediction steps may be investigated to further improve the estimation accuracy.

Table I. Estimation errors for encoding time and number of bits.

Resolution	**Estimation error**			
	Overall bits	**SOP-level bits**	**Overall enc. time**	**SOP-level enc. time**
3840 × 2160@60fps	1.3%	6.6%	2.1%	8.0%
3840 × 2160@50fps	1.0%	8.5%	1.8%	8.0%
1920 × 1080@60fps	1.3%	10.4%	2.7%	9.1%
1920 × 1080@50fps	1.3%	11.2%	2.1%	9.9%
1920 × 1080@24fps	2.9%	15.0%	1.6%	15.0%
Total average	**1.2%**	**8.4%**	**2.2%**	**8.4%**

Table II. Overall performance.

Bandwidth	**Resolution**	**Accuracy**		**Quality Δ [dB]**
		Proposed method	Fixed QP	
128 kbps	3840x2160@60fps	0.3%	3.2%	0.31
128 kbps	3840x2160@50fps	0.2%	4.0%	0.60
128 kbps	1920x1080@60fps	0.3%	5.2%	0.03
128 kbps	1920x1080@50fps	0.3%	2.3%	0.63
128 kbps	1920x1080@24fps	0.5%	2.6%	0.63
256 kbps	3840x2160@60fps	0.3%	3.6%	0.44
256 kbps	3840x2160@50fps	0.2%	3.2%	0.67
256 kbps	1920x1080@60fps	0.2%	6.9%	0.27
256 kbps	1920x1080@50fps	0.2%	2.9%	0.65
256 kbps	1920x1080@24fps	0.3%	2.1%	0.78
512 kbps	3840x2160@60fps	0.3%	2.3%	0.56
512 kbps	3840x2160@50fps	0.2%	3.7%	0.75
512 kbps	1920x1080@60fps	0.2%	6.3%	0.20
512 kbps	1920x1080@50fps	0.2%	2.8%	0.83
512 kbps	1920x1080@24fps	0.2%	2.5%	0.85
Total average		**0.3%**	**3.6%**	**0.55**

References

[1] G. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J. Ohm and G. Sullivan, "Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 76–90, Jan. 2016.

[4] G. Correa, P. Assuncao, L. Agostini, L. Cruz, "Pareto based method for high efficiency video coding with limited encoding time," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 9, pp. 1734–1745, Aug. 2015.

[5] K. P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," Joint Video Team, Tech. Rep. JVT N046, Hong Kong, China, Jan. 2005.

[6] T. Chiang and Y. Q. Zhang, "A new rate control scheme using quadratic rate distortion model," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 1, pp. 246–250, Feb. 1997.

[7] H. Choi, J. Nam, J. Yoo, D. Sim, and I. V. Bajic, "Rate control based on unified RQ model for HEVC," ITU T, Tech. Rep. JCTVC H0213, San Jose, CA, USA, Feb. 2012.

[8] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. Sullivan, "Rate constrained coder control and comparison of video coding standards," IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 688–703, Jul. 2003.

[9] X. Li, N. Oertel, A. Hutter, A. Kaup, "Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 19, pp. 193–205, Feb. 2009.

[10] B. Li, H. Li, L. Li, and J. Zhang, "Lambda domain rate control algorithm for high efficiency video coding," IEEE Trans. Image Process., vol. 23, no. 9, pp. 3841–3854, Sep. 2014.

[11] S. Wang, S. Ma, S. Wang, D. Zhao, and W. Gao, "Quadratic ρ domain based rate control algorithm for HEVC," in IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, Vancouver, BC, Canada.

[12] F. Bossen, B. Bross, K. Sühring, D. Flynn, "HEVC complexity and implementation analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[13] X. Li, M. Wien, and J. Ohm, "Rate complexity distortion optimization for hybrid video coding," IEEE Trans. Circuits Syst. Video Technol., Vol. 21, no. 7, pp. 957–970, Jul. 2011.

[14] L. Merritt, "x264: A high performance H.264/AVC encoder," Feb. 2016. [Online]. Available: http://akuvian.org/src/x264/overview_x264_v8_5.pdf.

[15] J. Vanne, M. Viitanen, T. Hamalainen, and A. Hallapuro, "Comparative Rate Distortion Complexity Analysis of HEVC and AVC Video Codecs," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1885–1898, Dec. 2012.

[16] F. Bossen, "Common HM Test Conditions and Software Reference Configurations," in JCTVC L1100, Geneva, Switzerland, January 2013.

[17] D. Marpe, H. Schwarz, T. Wiegand, "Context based adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 620–636, Jul. 2003.

[18] Turing codec software repository, [Online]. Available: https://github.com/bbc/turingcodec

[19] S. G. Blasi, M. Naccari, R. Weerakkody, J. Funnell and M. Mrak, "The open source turing codec: towards fast, flexible and parallel HEVC encoding," in International Broadcasting Convention, Amsterdam, Netherlands, Sept. 2016.

ISBN 978-90-827970-1-5