Front. Comput. Sci., 2016, 10(1): 157–166
DOI 10.1007/s11704-015-5066-1

RESEARCH ARTICLE

Improved rate-distortion optimized video coding using non-integer bit estimation and multiple Lambda search

Sio Kei IM (✉)[1], Mohammad Mahdi GHANDI[2]

1 MPI-QMUL Information Systems Research Centre, Macao Polytechnic Institute, Macao, China
2 Department of Electronic Systems Engineering, University of Essex, Colchestey CO4 3SQ, UK

Abstract Many modern video encoders use the Lagrangian rate-distortion optimization (RDO) algorithm for mode decisions during the compression procedure. For each encoding stage, this approach involves minimizing a cost, which is a function of rate, distortion and a multiplier called Lambda. This paper proposes to improve the RDO process by applying two modifications. The first modification is to increase the accuracy of rate estimation, which is achieved by computing a non-integer number of bits for arithmetic coding of the syntax elements. This leads to a more accurate cost computation and therefore a better mode decision. The second modification is to search and adjust the value of Lambda based on the characteristics of each coding stage. For the encoder used, this paper proposes to search multiple values of Lambda for the intra-4×4 mode decision. Moreover, a simple shift in Lambda value is proposed for motion estimation. Each of these modifications offers a certain gain in RDO performance, and, when all are combined, an average bit-rate saving of up to 7.0% can be achieved for the H.264/AVC codec while the same concept is applicable to the H.265/HEVC codec as well. The extra added complexity is contained to a certain level, and is also adjustable according to the processing resources available.

Keywords rate distortion optimization, Lambda adjustment, non-integer bit estimation, H.264/AVC, H265/HEVC video coding

Received March 13, 2015; accepted July 20, 2015

E-mail: marcusim@ipm.edu.m

1 Introduction

Video codecs are deployed in a variety of applications that offer video services, including, but are not limited to, television broadcasting, recorded media and a wide range of remote operations from space and medical to underwater explorations. In these applications, the efficiency of the video compression system to deliver better picture quality plays a key role in the success of the service in a competitive market. A video coding standard such as H.265/HEVC [1] or H.264/AVC [2] only specifies the syntax of the coded video, and there is massive flexibility and freedom in selecting the coding modes for each small unit or block of video. The role of the video encoder is to choose a combination of coding modes in order to maximize the picture quality, i.e., to minimize distortion, given a bit-rate budget. This process is called rate-distortion optimization or RDO.

One of the best solutions to find a near-optimum coding configuration is the localized Lagrangian optimization algorithm [3]. In this process, the following cost J is minimized for each block of the picture in a raster scanning order:

$$J = D + \lambda \times R, \tag{1}$$

where for an examined configuration, D is the distortion between the coded and the original samples, R is the sum of coded bits generated, and λ is proportional to the quantization parameter (QP) given by an empirical formula [4]. This process incorporates a number of simplifications, including the assumption that consecutive blocks are independent and that λ and QP have a direct relationship.

The majority of researches focus on reducing the complexity of the RDO algorithm with minimum bitrate/quality penalty. However, in applications that extra processing resources or time is available, it is possible to further improve the optimization performance with small or moderate increases in complexity. Some proposals suggest an increase in the searched cases, e.g., by multi-pass coding [5,6] or multi-QP optimization [7,8]. A few other proposals are based on the intuitive assumption that more accurate estimation of the elements in Eq. (1) yields better optimization. These include using different metrics for distortion [9] and using different values for λ [10–12]. In all the above papers, the integer number of bits estimated is supposed to be accurate enough, especially if full entropy coding is carried out for each attempted coding configuration. However, in arithmetic encoding, each syntax element is represented with a non-integer number of bits [13,14]. Moreover, the value of Lambda used for each sub-process of RDO is normally kept unchanged, even though the rate/distortion models may be different for each sub-process. These sub-processes can include intra and inter prediction modes decisions and motion estimation.

This paper proposes two modifications in order to improve the efficiency of the RDO algorithm. First, the bit rate estimation is proposed to be more accurate. We demonstrate a non-integer bit representation of the coded syntax elements and use it to improve the accuracy of the rate estimation that will yield better RDO. The impact is more noticeable for smaller blocks where the produced number of bits is small. A fractional shift can have more impact on RDO cost. Moreover, the motion vector biasing based on the average number of motion-vector bits has been modified to be non-integer based.

The second contribution is the adjustment of Lambda value according to the properties of each RDO sub-process. For the H.264/AVC codec used, in order to find more optimized intra-4×4 mode-combinations, we propose to search different values of λ for mode decision. Moreover, for motion estimation, a simple and low complexity Lambda adjustment is proposed. The combination of these improvements results in a considerable gain, while offering a simple and complexity-adjustable approach.

For the purpose of this research, the H.264/AVC codec is used as a benchmark. It appears that the H.264/AVC standard will still be utilized heavily for the next few years [15], and more optimized H.264/AVC codecs are desired during this period for the consumer and professional markets. However, the concept of this paper can also be applied to other codecs, including the High Efficiency Video Coding [16–18] that has been released in 2013 [1] and is awaiting its first major deployment.

The rest of the paper is organized as follows. Section 2 provides details of the proposed fractional bit estimation for H.264/AVC and H.265/HEVC arithmetic coding along with the application of the more accurate bit estimation for the RDO algorithm. Section 3 gives the details of the proposed Lagrangian multiplier search and adjustment for RDO. The simulation results are given in Section 4, and finally Section 5 concludes the paper.

2 Non-integer bit estimation in CABAC

For entropy coding, both H.264/AVC and H.265/HEVC employ context-based binary arithmetic coding (CABAC) [13,14]. Figure 1 illustrates the flow of CABAC, in which syntax elements are represented and coded as a sequence of binary symbols called bins. The value of a bin can be either equal to the most probable symbol (MPS) or least probable symbol (LPS). For coding each bin, the probabilities (p) of LPS and MPS occurrences are represented by *range_lps* and *range_mps* respectively where the sum of *range_lps* and *range_mps* is called range. Therefore, according to information theory [19], the estimated bits (b) needed to represent

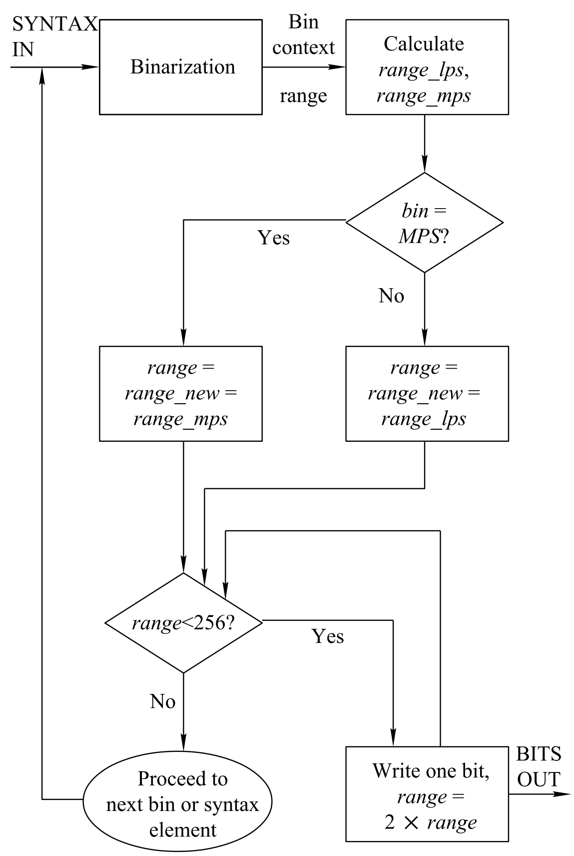

Fig. 1 CABAC encoding of syntax elements

the bin would be:

$$b = \log_2\left(\frac{1}{p}\right) = \begin{cases} \log_2\left(\dfrac{range}{range\_lps}\right), & bin = \text{LPS}; \\ \log_2\left(\dfrac{range}{range\_mps}\right), & bin = \text{MPS}. \end{cases} \tag{2}$$

For each category of bins (context), the probability estimation (*range_lps*) is adaptively updated and stored in tables, while the initial values are given in the standard. If $range\_lps = range\_mps$, the number of bits to represent the bin is equal to 1 bit. However, if $range\_lps < range\_mps$, more than 1 bit is needed for $bin = \text{LPS}$ and less than 1 bit is needed for $bin = \text{MPS}$.

In the arithmetic encoding process, the maximum and initial value of range as specified in H.264/AVC and H265/HEVC is 510 [1,2]. For encoding each bin, range is updated to *range_new* which is either *range_mps* if $bin = \text{MPS}$, or *range_lps* if $bin = \text{LPS}$ (see Fig. 1).

This procedure continues until range becomes smaller than 256. In that case, there is a renormalization phase in which range is multiplied by two as many times as required until it becomes greater than, or equal to 256. For each multiplication step, one bit is written to the output video bitstream. The above procedure implies that for each syntax element a non-integer number of bits may be generated, and only the integer portion of these bits is written while the rest are carried over to the next encoding round. The non-integer number of bits generated for each bin can be estimated by:

$$b = \log_2(range) - \log_2(range\_new), \tag{3}$$

where *range* is before coding the bin and *range_new* is before renormalization. This equation can also be worked out from Eq. (2).

For a group of syntax elements, e.g., for a block, the sum of the b values mentioned above will represent the total non-integer number of bits for the group. Note that we do not need to carry out the floating point operation of Eq. (3) for each bin. Instead, for a group of syntax elements, the *range_begin* and *range_end* values are stored, which represent the value of range at the beginning of coding the group and after coding the last syntax element including the renormalization stage. The integer number of bits is already measured separately by counting the written bits. We only add the fractional portion which is equal to:

$$f = \log_2(range\_begin) - \log_2(range\_end). \tag{4}$$

For a group of syntax elements, f can take real values between $(-1.0, +1.0)$. Positive values mean that an integer number of bits are written for the group, and the fractional portion is carried over to the next round. Negative values indicate that a fraction of the first written bit for the group had actually belonged to the previous syntax elements before the start of the group.

2.1 Fractional bit estimation for mode decision

Here, without a loss of generality, we focus on the RDO implementation in the H.264/AVC joint model (JM) reference software[1]. In JM, there are a number of different stages that use the actual bit estimation for RDO. We have identified the following stages (applicable to the main profile):

- Mode decision for 16×16 intra-prediction mode.
- Mode decision for 4×4 intra-prediction mode.
- Mode decision for sub-macroblock inter-prediction mode:
  $$8 \times 8, 8 \times 4, 4 \times 8 \text{ and } 4 \times 4.$$
- Final mode decision stage for a 16×16 macroblock.

We have modified the JM software and added the fractional bit estimation to the above stages. The cost function is modified as follows:

$$J = D + \lambda \times R + \lambda \times f, \tag{5}$$

where R is the integer number of bits, and f is its fractional portion given by Eq. (4).

The addition of the fractional portion of the bits in the cost function has different effects depending on λ, D and R values. Generally, with low λ values the effect is lower than that with higher λ, and our observations show that for higher λ (i.e., high QP) the fractional portion can shift the cost value up or down by up to 10%. Also, when R is very low, e.g., for sub-macroblock mode decision or the intra-4×4 mode decision, the effect is more than that for the final macroblock mode decision stage which considers the bits for all the syntax elements of the macroblock.

2.2 Fractional bit estimation for motion estimation

The rate-distortion optimization is at the heart of motion estimation (ME) as well. The RDO-based ME [4] finds the motion vector (MV) would minimize the following cost:

$$J_{\text{MV}} = D_{\text{MC}} + \lambda_{\text{ME}} \times R(\text{MVD}), \tag{6}$$

[1] http://iphome.hhi.de/suehring/tml/download/

where D_{MC} is the distortion of the motion-compensated block, and λ_{ME} is the Lambda used for this process as further explained in Section 3.3. $R(\mathrm{MVD})$ is the bit rate of the examined motion vector difference (MVD) which is the difference between the MV and its predicted value. However, since there are hundreds of motion vector (and MVD) values to search for each block, the bit estimation in the reference software is not performed by actual entropy coding. Instead, it is extracted based on a statistical table that relates each motion vector difference (MVD) value to an integer number of bits.

In this paper, the MVD bit estimation is table-based as well. However, we developed a new table that relates each MVD to a non-integer number of bits. This table is produced after a number of experiments with different contents and measuring the actual non-integer bits for MVDs. The mean values are computed and stored in the table. It should be added that in our implementation, the floating point values are approximated in integer mode to minimize the additional complexity. For each MVD, $R'(\mathrm{MVD})$ is defined as an integer value equal to the statistical, non-integer bit rate estimation multiplied by 256 rounded to an integer value and stored in the table. Therefore, for motion estimation, we replace Eq. (6) with:

$$J_{\mathrm{MV}} = D_{\mathrm{MC}} + (\lambda_{\mathrm{ME}} \times R'(\mathrm{MVD})) >> 8. \tag{7}$$

This offers the additional accuracy of the non-integer bit estimation to motion estimation while only adding a small extra complexity. The gain offered by accurate bit estimation is reported in the following section.

2.3 Evaluation of non-integer bit estimation

The H.264/AVC reference software version 17.2 is modified to handle all proposed methods in this paper, and a series of simulations are carried out to measure the improvement offered by accurate bit estimation. For most of the comparisons, the Bjontegaard PSNR and bit-rate differences (BD-PSNR and BD-Rate) between the proposed and the reference method are shown. The procedure detailed in Ref. [20] is used to calculate BD-PSNR and BD-Rate with QP points of 28, 32, 36, 40 and 44. For all tests the full-RDO mode is enabled, entropy coding mode is CABAC and all the main-profile features of JM in progressive format are enabled. For all-intra tests, 100 intra frames are coded; and for intra-inter (IPPP) tests, 150 frames are coded (frame rates given in the table captions) with one intra update every second, and the number of reference frames is set to 4.

A number of simulations are conducted with non-integer bit estimation enabled and disabled (i.e., reference). Tables 1 and 2 demonstrate the BD-PSNR and BD-Rate results of all-intra pictures for QCIF and 4CIF resolutions respectively, and Tables 3 and 4 show the results for IPPP tests. From the tables it can be seen that enabling the non-integer bit estimation reduces the average bit rate from 0.2% to 7.0% for IPPP video. However, the gain is less for intra-only videos from 0% to 0.5%. This is because for intra-pictures, the number of bits for each sub-block is more than that for inter-pictures. Therefore, the additional fractional bit estimation does not have a significant impact on all-intra RDO processes. Our observations show that this impact can be large only when QP is high [21], because a fraction of a bit is proportionally more significant when fewer bits are produced at higher QPs.

Table 1 Non-integer bit-estimation vs. reference QCIF@10Hz, all intra pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Foreman	–0.37	+0.03	+0.00	+0.01
Carphone	–0.31	+0.02	+0.01	+0.02
News	–0.08	+0.01	+0.01	+0.01
Akio	–0.27	+0.02	+0.01	+0.00
Silent	–0.52	+0.03	+0.02	+0.02
Paris	–0.20	+0.02	+0.00	+0.00

Table 2 Non-integer bit-estimation vs. reference 4CIF@30Hz, all intra pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Crowd-run	–0.29	+0.02	+0.01	+0.01
Into-tree	–0.31	+0.01	+0.01	+0.00
Life	–0.28	+0.02	+0.01	+0.01
Park-joy	–0.32	+0.02	+0.01	+0.01
Rush-hour	–0.18	+0.01	+0.00	+0.00
Sunflower	–0.33	+0.02	+0.01	+0.01

In terms of complexity, experiments show that an additional 3% average encoding time is needed when non-integer bit estimation is enabled. Therefore, due to its simple nature, we believe that non-integer bit estimation even on its own, is a worthwhile addition to the RDO algorithm. The following section explains the second improvement in the RDO process.

3 Lambda search and adjustment for RDO

In the usual RDO algorithm, a fixed quantization parameter (QP) is first chosen for the picture and then for the purpose of mode selection, the value of λ is computed as:

Table 3 Non-integer bit-estimation vs. reference QCIF@15Hz, IPPP pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Foreman	–3.06	+0.16	+0.09	+0.08
Carphone	–2.40	+0.12	+0.04	+0.03
News	–3.81	+0.25	+0.13	+0.13
Akio	–1.93	+0.12	+0.09	+0.08
Silent	–2.62	+0.12	+0.07	+0.04
Paris	–1.93	+0.22	+0.14	+0.15

Table 4 Non-integer bit-estimation vs. reference 4CIF@30Hz, IPPP pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Crowd-run	–7.09	+0.31	+0.20	+0.22
Into-tree	–0.17	+0.04	+0.03	–0.02
Life	–4.41	+0.13	+0.15	+0.17
Park-joy	–3.23	+0.13	+0.10	+0.07
Rush-hour	–0.25	+0.00	+0.00	+0.00
Sunflower	–0.98	+0.02	+0.04	+0.04

$$\lambda(\text{QP}) = K \times 2^{(\text{QP}-12)/3}, \tag{8}$$

with K being a constant which may be tuned based on picture type (typically 0.85). This Lambda value is used for all stages of RDO-based mode decision, which is also referred to as λ_{MODE}. The relationship between Lambda and QP (see Eq. (8)) is drawn based on experiments described in Ref. [4]. In these experiments, a series of fixed Lambda values for the RDO process of a whole picture are used. At each Lambda, different QPs for each macroblock are examined, and the QP value that minimizes the Lagrangian cost is chosen for the macrolock. Then, the highest occurrence percentage of each QP value is noted. A relationship is established between Lambda and the highest occurred QP and Eq. (8) is concluded. To illustrate this relationship, we repeated this experiment for four Lambda values of a selected video sequence, and the QP occurrences are depicted in Fig. 2. As can be seen from the figure, this relationship is only statistically correct, and for a significant portion of the picture the best Lambda value may not be equal to the average value.

For motion estimation, a λ_{ME} which has the same dependence to QP is used, and it will be further explained in Section 3.3. Moreover, while Eq. (8) would offer a good compromise for simple macroblock mode decision and motion estimation, the relationship between rate and distortion for each sub-process of mode decision is not the same as the macroblock mode decision. Furthermore, the best value of Lambda may be different depending on the content and other configurations. This is our motivation for exploring the possibility of adjusting Lambda for two important sub-processes in the RDO algorithm, namely intra-4×4 mode-decision and motion estimation.

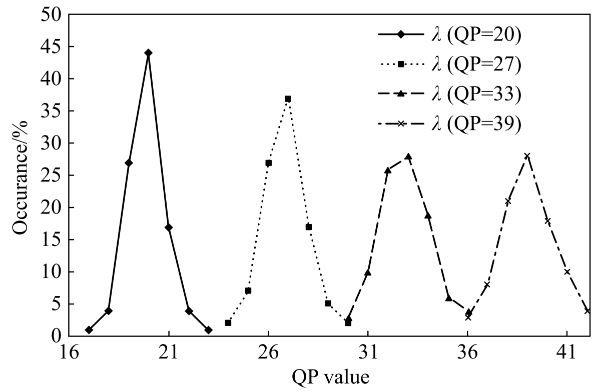

Fig. 2 An example of relative QP occurrences in multi-QP search with different λ (QP) values of Eq. (8)

3.1 Intra-4×4 mode-decision in H.264/AVC reference model

A 16×16 pixel macroblock in intra-4×4 mode is divided into sixteen 4×4 blocks, each having an intra-4×4 prediction-mode (P) for luminance samples. Let the intra-4×4 mode-combination be $P_0, P_1, \ldots, P_{15}$, where P_i is the intra-4×4 prediction-mode of the ith 4×4 block. Each P_i can have nine different values. The context adaptive entropy coding and intra-prediction between adjacent blocks create a dependency between them in which the bit rate and distortion properties of each of the nine modes of each 4×4 block depend on the selected modes of the previous blocks. Therefore, the total possible mode-combinations for the intra-4×4 macroblock are 16^9. Trying all these combinations is not practical, since this number is extremely large. In the H.264/AVC reference model software, a compromise is made by assuming independence between adjacent 4×4 blocks. In that case, in a sequential scanning of the 4×4 blocks, each P_i is decided by examining the nine possible modes. The chosen mode is the one that has the minimum $J_{4\times4}$ cost:

$$J_{4\times4} = D_{4\times4} + \lambda_{4\times4} \times R_{4\times4}, \tag{9}$$

where for an examined mode, $D_{4\times4}$ and $R_{4\times4}$ are the distortion and rate of the 4×4 block respectively, and $\lambda_{4\times4} = \lambda(\text{QP})$ given in Eq. (8). As soon as the intra-4×4 mode combination $(P_0, P_1, \ldots, P_{15})$ is decided, the total D and R for the macroblock in intra-4×4 mode is computed, and this mode is chosen if it offers the lowest cost among other macroblock modes such as intra-16×16. This process is shown in Fig. 3.

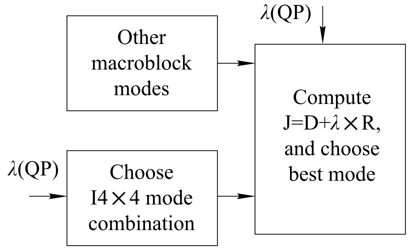

Fig. 3 Flow of the intra mode selection in reference RDO (I4×4: Intra-4×4)

3.2 Proposed Lambda search for intra-4×4 mode decision

With the above reference method, only 1 out of 9^{16} of the intra-4×4 mode-combinations is actually examined. Also, the $\lambda_{4\times4}$ used in Eq. (9) is the same as the one used for the overall macroblock mode selection given in Eq. (8), even though $R_{4\times4}$ and $D_{4\times4}$ are significantly smaller than the R and D of the macroblock and their statistics are different. This suggests that the used $\lambda_{4\times4}$ may not be appropriate. To address these issues, we propose to increase the number of examined intra-4×4 mode-combinations from 1 to N, each decided using a different $\lambda_{4\times4}$. The complexity of the encoder will increase as N increases, so we limit our experiments to N values between 2 to 7. The core of the proposal is to find the N intra-4×4 mode-combinations, each of them resulting in lowest RDO cost if using a different value of $\lambda_{4\times4} = \lambda_{i4\times4}$ with $i = 0$ to $N-1$. The values of $\lambda_{i4\times4}$ are determined based on two parameters: Qd and Qoff_0 which adjust the QP given to Eq. (8) for computing each $\lambda_{i4\times4}$. For each $i = 0$ to $N-1$, we try:

$$\lambda_{i4\times4} = \lambda(\text{QP} + \text{Qoff}_i). \tag{10}$$

$$\text{Qoff}_i = \text{Qoff}_0 + i \times \text{Qd}. \tag{11}$$

After running several experiments, we have worked out some practical values for Qd and Qoff_0 for each N that produce good results. These adjustments are listed in Table 5 which also shows the values of Qoff_i.

Table 5 Selected Qd and Qoff_0 for each N

N	Qoff_0	Qd	$\text{Qoff}_0, \ldots, \text{Qoff}_{N-1}$
2	–3.0	3.5	–3.0, +0.5
3	–3.5	2.5	–3.5, –1.0, +1.5
4	–3.0	1.5	–3.0, –1.5, +0.0, +1.5
5	–3.0	1.5	–3.0, –1.5, +0.0, +1.5, +3.0
6	–3.0	1.0	–3.0, –2.0, –1.0, +0.0, +1.0, +2.0
7	–3.0	1.0	–3.0, –2.0, –1.0, +0.0, +1.0, +2.0, +3.0

Note that the QP used for quantization remains unchanged for all these processes. After the N intra-4×4 mode combinations are determined, the best combination is chosen in the usual RDO with the original $\lambda(\text{QP})$, as depicted in Fig. 4.

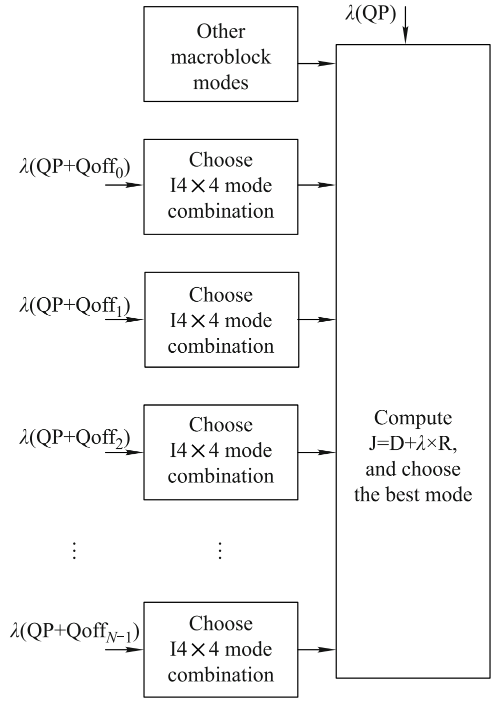

Fig. 4 Flow of the intra mode selection in proposed RDO (I4×4: Intra-4×4)

3.3 Lambda adjustment for motion estimation

The rate used for RDO-based motion estimation (ME) is the rate of motion vectors as described in Section 2.2. The distortion metric used for ME can be the sum of squared difference (SSD) or the sum of absolute difference (SAD) between the motion-compensated and the original blocks. As for the Lambda,

$$\lambda_{\text{ME}}(\text{QP}) = \begin{cases} \lambda_{\text{Mode}}(\text{QP}), & \text{Mode} = \text{SSD}; \\ \sqrt{\lambda_{\text{Mode}}(\text{QP})}, & \text{Mode} = \text{SAD}. \end{cases} \tag{12}$$

Due to the complexity of motion estimation, the rate used is only the bit rate of motion vectors drawn from a table, and the rate of the other syntax elements including the resulting residual data is not taken into consideration for the RDO-based ME. Some proposals [22] have suggested that the extra bits should be computed and added at the cost of extra complexity. In this paper, however, we aim to contain the motion estimation complexity and therefore, an adjustment to the λ_{ME} is suggested to account for the lack of extra bits, illustrated as follows:

$$\lambda^P_{\text{ME}} = \lambda_{\text{ME}}(\text{QP} + \text{Qoff}_{\text{ME}}), \tag{13}$$

where Qoff_{ME} determines a shift to the QP when computing the modified Lambda) for the process of motion estimation. Again, note that this Qoff_{ME} is only used for the purpose of Lambda calculation and the QP of the picture remains intact.

Based on experiments, we have adjusted $\mathrm{Qoff}_{\mathrm{ME}}$ to a fixed value of +0.5.

3.4 Performance evaluation of Lambda search for intra-4×4 mode decision

To assess the Lambda-search algorithm, simulations are carried out for intra-only videos. The number of $\lambda_{i4\times4}$ values searched (N) are from $N = 2$ to 7, with the configurations given in Table 5 for each N, and the results are compared with the reference algorithm. Note that the fractional bit estimation is kept disabled for this assessment while in Section 4 we will examine the effect of both together. Tables 6 and 7 demonstrate the performance gain offered by the Lambda search algorithm. From the tables, it can be seen that the proposed method offers consistent improvement over the reference algorithm, and the higher the number of Lambdas searched (N), the better the results. A bit rate saving of up to 2.1% is achieved with N=2, while this saving is up to 3.1% for N=7. It should be noted that increasing N to larger values than 7 will still offer extra gain, but it will become less significant as N grows while the complexity still increase proportionally.

Compared to the reference, the proposed Lambda search would approximately take 70% extra encoding time for N=2, and 2.5 times for N=5, and 5 times for N=7. However, even for N=7, the total encoding time for the intra picture is less than that of an inter picture due to the absence of motion estimation. Also, if multiple cores or hardware resources are available, the N processes can be easily handled in parallel, resulting in no extra encoding time.

Therefore, due to the relatively lower complexity of intra-pictures and their significant impact on video quality, we believe that the extra complexity added by the Lambda-search is worth the gain. Note that for inter-frames we will disable Lambda-search, and it will only be enabled for intra-frames. Therefore, the impact on overall encoding time of IPPP videos will not be significant.

4 Simulation results of the combined proposed methods

The proposed fractional bit-estimation explained in Section 2 and the Lambda search and adjustments proposed in Section 3 can be combined to improve the RDO algorithm. We expect that this combination offers improvement to both intra and inter pictures. For the purpose of this assessment we chose N=7, but lower values of N can be used to lower the complexity as discussed in Section 3. Furthermore, while the fractional bit estimation is used for both intra and inter pictures, the Lambda search for intra-4×4 mode decision is only applied to intra pictures and not for the inter pictures. For P pictures, only the basic Lambda adjustment for motion estimation is applied.

Table 6 Lambda-search for intra-4×4 enabled vs. reference QCIF@10Hz, all-intra pictures

	Foreman		Carphone		News		Akio		Silent		Paris	
N	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB
2	–1.51	+0.10	–1.22	+0.08	–0.84	+0.07	–1.06	+0.08	–1.02	+0.05	–0.48	+0.04
3	–1.96	+0.13	–1.55	+0.11	–1.02	+0.08	–1.27	+0.09	–1.34	+0.07	–0.58	+0.05
4	–2.24	+0.15	–1.80	+0.12	–1.17	+0.10	–1.58	+0.12	–1.76	+0.09	–0.82	+0.06
5	–2.33	+0.15	–1.87	+0.13	–1.20	+0.10	–1.56	+0.12	–1.83	+0.10	–0.94	+0.07
6	–2.38	+0.16	–1.91	+0.12	–1.33	+0.11	–1.62	+0.12	–1.98	+0.10	–1.13	+0.09
7	–2.56	+0.17	–2.04	+0.14	–1.43	+0.12	–1.67	+0.12	–2.01	+0.11	–1.15	+0.09

Table 7 Lambda-search for intra-4×4 enabled vs. reference 4CIF@30Hz, all intra pictures

	Foreman		Carphone		News		Akio		Silent		Paris	
N	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB	BD-Rate /%	BD-PSNR -Y /dB
2	–0.70	+0.04	–0.42	+0.02	–1.17	+0.06	–0.55	+0.04	–2.10	+0.09	–2.15	+0.12
3	–0.99	+0.06	–0.71	+0.03	–1.47	+0.08	–0.85	+0.06	–2.45	+0.10	–2.50	+0.14
4	–1.06	+0.06	–0.78	+0.03	–1.60	+0.08	–0.92	+0.06	–2.68	+0.11	–2.76	+0.15
5	–1.06	+0.06	–0.84	+0.03	–1.64	+0.09	–0.89	+0.06	–2.79	+0.12	–2.82	+0.16
6	–1.18	+0.07	–0.91	+0.03	–1.76	+0.09	–1.02	+0.07	–2.99	+0.13	–2.97	+0.16
7	–1.18	+0.07	–0.95	+0.03	–1.79	+0.09	–1.01	+0.07	–3.12	+0.13	–3.03	+0.17

Tables 8 and 9 show the BD-Rate and BD-PSNR results of the combined method for all-intra, QCIF and 4CIF videos respectively. Comparing these results with the ones reported in Sections 2 and 3, we can find how much additional gain is achieved by combining the two methods together. As expected, for intra pictures, most of the gain is offered by Lambda search while the fractional bit estimation adds a small additional gain.

Table 8 Combined proposed methods vs. reference QCIF@10Hz, all-intra pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Foreman	–2.99	+0.20	+0.00	+0.01
Carphone	–2.38	+0.16	+0.00	+0.00
News	–1.51	+0.12	+0.01	+0.00
Akio	–1.69	+0.13	+0.01	+0.01
Silent	–2.51	+0.13	+0.00	+0.01
Paris	–1.40	+0.11	+0.00	+0.00

Table 9 Combined proposed methods vs. reference 4CIF@30Hz, all-intra pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Crowd-run	–1.38	+0.08	+0.01	+0.01
Into-tree	–1.09	+0.04	+0.01	+0.02
Life	–2.04	+0.11	+0.00	+0.01
Park-joy	–1.21	+0.08	+0.00	+0.01
Rush-hour	–3.36	+0.14	+0.00	+0.01
Sunflower	–3.40	+0.19	+0.00	+0.00

Tables 10 and 11 show the BD-RATE and BD-PSNR results for IPPP videos for QCIF and 4CIF resolutions. In order to further assess the gain, the simulation results for some CIF videos are added for this mode. As expected, the combination of the two methods offers an additional gain across different contents.

Table 10 Combined proposed methods vs. reference QCIF@15Hz, IPPP pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Foreman	–4.48	+0.23	+0.08	+0.11
Carphone	–3.48	+0.17	+0.04	+0.07
News	–4.80	+0.31	+0.18	+0.12
Akio	–3.34	+0.20	+0.09	+0.08
Silent	–4.11	+0.20	+0.06	+0.04
Paris	–4.77	+0.27	+0.13	+0.12

Finally, to demonstrate the range of PSNR values assessed, Figs. 5–7 illustrate the PSNR vs. bit-rate curves for three selected videos. The gain achieved by the proposed method is clearly demonstrated in these figures.

Table 11 Combined proposed methods vs. reference 4CIF@30Hz, IPPP pictures

	BD-Rate /%	BD-PSNR -Y /dB	BD-PSNR -U /dB	BD-PSNR -V /dB
Crowd-run	–7.11	+0.32	+0.21	+0.22
Into-tree	–2.13	+0.04	+0.01	+0.00
Life	–4.17	+0.14	+0.15	+0.16
Park-joy	–5.06	+0.22	+0.17	+0.09
Rush-hour	–0.95	+0.03	+0.02	+0.04
Sunflower	–0.99	+0.02	+0.04	+0.04

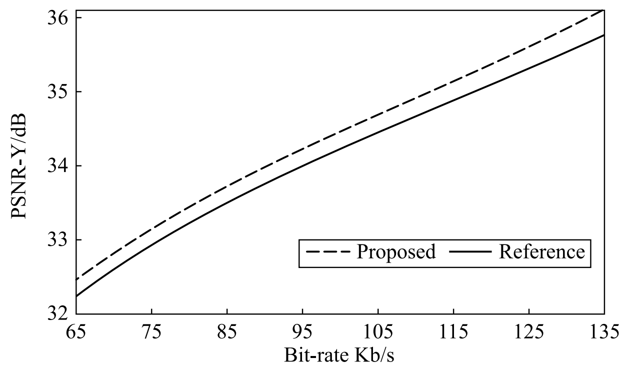

Fig. 5 The combined proposed rate-distortion curves vs. reference Foreman QCIF@15Hz, IPPP pictures

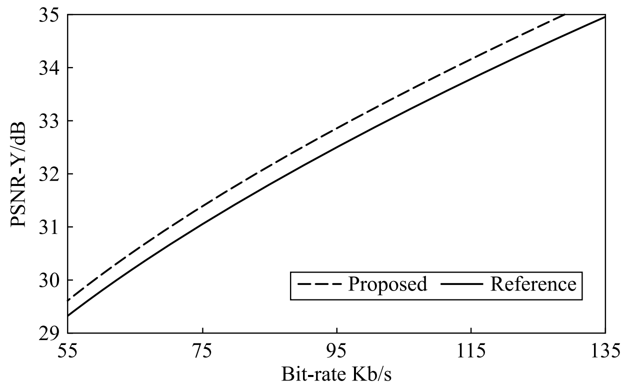

Fig. 6 The combined proposed rate-distortion curves vs. reference Pairs QCIF@15Hz, IPPP pictures

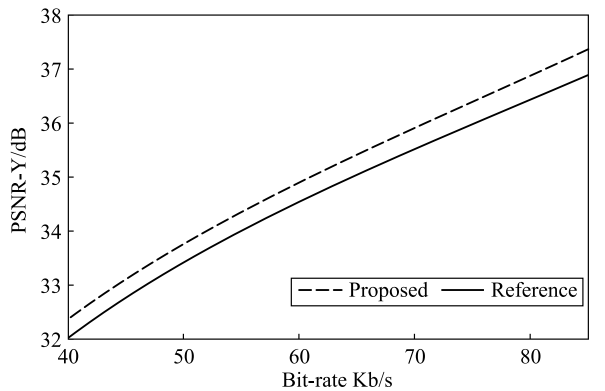

Fig. 7 The combined proposed rate-distortion curves vs. reference News QCIF @15Hz, IPPP pictures

As for the complexity of the combined methods, for intra-only tests, the total encoding time is approximately 5 times more than reference, for the chosen N=7 (i.e., maximum N). For intra-inter tests, the additional encoding time is 20% on average. As discussed earlier, this additional encoding time can be reduced by either lowering N or parallel processing.

5 Conclusions

This paper has explored some of the potentials of the rate-distortion optimization (RDO) algorithm for video coding. Improvements are achieved by increasing the accuracy of rate estimation, and also by adjusting the Lagrangian multiplier (Lambda) for some RDO sub-processes. For accurate bit estimation, a practical formula to compute the non-integer number of bits for coded syntax elements in CABAC mode is proposed and employed. The Lambda adjustment is made to intra-4×4 mode decision and also to motion estimation. Simulations show that each of these improvements offers some gain in rate-distortion efficiency while incurring a controlled and adjustable complexity overhead. The combination of these improvements is shown to offer a more significant gain in the H.264/AVC encoder. While the proposed algorithms offer no divergence from the H.264/AVC standards and can be used in the current systems, they can also be adopted by any encoder that applies rate distortion optimization including the new H.265/HEVC.

References

1. ITU-T Recommendation H.265. Series H: audiovisual and multimedia systems: infrastructure of audiovisual services — coding of moving video. High Efficiency Video Coding, 2013
2. ITU-T Recommendation H.264. Advanced video coding for generic audiovisual services. ISO/IEC, 2013
3. Sullivan G J, Wiegand T. Rate-distortion optimization for video compression. IEEE Signal Processing Magazine, 1998, 15(6): 74–90
4. Wiegand T, Schwarz H, Joch A, Kossentini F, Sullivan G J. Rate-constrained coder control and comparison of video coding standards. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 688–703
5. An C, Nguyen T Q. Iterative rate-distortion optimization of H.264 with constant bit rate constraint. IEEE Transactions on Image Processing, 2008, 17(9): 1605–1615
6. You J, Choi C, Jeong J. Modified rate distortion optimization using inter-block dependence for H.264/AVC intra coding. IEEE Transactions on Consumer Electronics, 2008, 54(3): 1383–1388
7. Li B, Xu J, Zhang D, Li H. QP refinement according to Lagrange multiplier for high efficiency video coding. In: Proceedings of 2013 IEEE International Symposium on Circuits and Systems (ISCAS). 2013, 477–480
8. Wu S, Liu P, Huang Y, Liu Q, Ikenaga T. On bit allocation and Lagrange Multiplier adjustment for rate-distortion optimized H. 264 rate control. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing. 2009, 1–6
9. Lee Y M, Sun Y T, Lin Y. SATD-based intra mode decision for H.264/AVC video coding. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(3): 463–469
10. Ghandi M M, Ghanbari M. A Lagrangian optimized rate control algorithm for the H. 264/AVC encoder. In: Processings of 2004 International Conference on Image Processing. 2004, 123–126
11. Zhang J, Yi X, Ling N, Shang W. Context adaptive Lagrange multiplier (CALM) for rate-distortion optimal motion estimation in video coding. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(6): 820–828
12. Li X, Oertel N, Hutter A, Kaup A. Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(2): 193–205
13. Marpe D, Schwarz H, Wiegand T. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 620–636
14. Sze V, Budagavi M. High throughput CABAC entropy coding in HEVC. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1778–1791
15. Richardson I E. The H.264 Advanced Video Compression Standard. New York: John Wiley & Sons, 2011
16. Ugur K, Andersson K, Fuldseth A, Bjontegaard G, Endresen L P, Lainema J, Wu Z. High performance, low complexity video coding and the emerging HEVC standard. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(12): 1688–1697
17. Han W J, Min J, Kim I K, Alshina E, Alshin A, Lee T, Park J H. Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(12): 1709–1720
18. Wiegand T. WD3: working draft 3 of high-efficiency video coding. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO. IEC JTC1/SC29/WG11, 2011
19. Shannon C E, Weaver W. The Mathematical Theory of Communication. Urbana: The University of Illinois Press, 1949
20. Senzaki K, Chono K, Aoki H, Tajime J, Senda Y. BD-PSNR/Rate computation tool for five data points, In: Proceedings of the Meeting of Joint Collaborative Team on Video Coding. 2010
21. Im S K, Ghandi M M, Lam C T. Non-integer bit estimation for rate-distortion optimized video coding. In: Processings of 2012 IEEE International Conference on Consumer Electronics. 2012, 88–89
22. Li Z, Tourapis A M. Motion estimation with entropy coding considerations in H. 264/AVC. In: Processings of the 15th IEEE International Conference on Image Processing. 2008, 2140–2143

Sio Kei Im received his degree in computer science in 1998 and the master degree in enterprise information system in 1999, both from King's College, University of London, UK. He received the PhD degree in electronic engineering from Queen Mary University of London (QMUL), UK in 2007. He gained the position of Lecturer within the Computing Programme at Macao Polytechnic Institute (MPI) in 2001. In 2005, he became the operations manager of the MPI-QMUL information systems research centre jointly operated by MPI and QMUL where he carried out signal processing work. In 2009, he was promoted to be the associate professor at the Macao Polytechnic Institute. Currently, he is a visiting scholar at School of Engineering, UCLA and a member of IET.

Mohammad Mahdi Ghandi received his BS and MS degrees in electronics engineering from the University of Tehran, Iran in 1998 and 2001 respectively. After two years of industrial experience in image and video coding, he, as a senior research officer, joined the Video Networking Group at the University of Essex, UK in 2003, where he published several papers in the field of video communications. He was also granted a PhD degree from this university. Since 2006, he returned back to TV broadcasting industry as a designer of compression systems while maintaining some of his academic activities. His research interests include reliable image and video transmission, advanced video codecs, and video transcoding.