

Accurate, Non-integer Bit Estimation for H.265/HEVC and H.264/AVC Rate-Distortion Optimization

Sio Kei Im

Macao Polytechnic Institute,
Macao, China
marcusim@ipm.edu.mo

Mohammad Mahdi Ghandi

University of Essex,
United Kingdom
mahdighandi@gmail.com

Ka Hou Chan

Macao Polytechnic Institute,
Macao, China
chankahou@ipm.edu.mo

Abstract—Lagrangian Rate Distortion Optimization (RDO) is widely employed in modern video encoders such as H.264/AVC and H.265/HEVC for mode decision. During the RDO process the distortion and bit rate of each mode has to be estimated and more an accurate estimation leads to better optimized decisions. This paper explores accurate, non-integer bit estimation and its impact on better RDO. Simulations show 0.2% to 13.0% bit rate saving for H.265/HEVC and 0.2% to 8.0% for H.264/AVC.

Keywords—Non-integer/Fractional Bit Estimation; CABAC; Accurate Rate-Distortion Optimization;

I. INTRODUCTION

In many applications that offer video services, compression plays a major role for delivering the video. In these applications, the efficiency of the video encoder to offer better picture quality is essential for 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 the larger task of selecting the optimum coding modes for each block or unit of the picture is left to the encoder design. One of the best solutions to this problem is the localized Lagrangian optimization algorithm [3] in which the following cost J is minimized for each block or unit of the picture, usually in a raster scanning order:

$$J = D + \lambda \times R \quad (1)$$

where D is distortion, R is the bit rate for each selected configuration and λ is proportional to the Quantization Parameter (QP) given by an empirical formula [4].

Most of the research in the literature is to simplify the RDO process and reduce complexity. However, when/where extra processing power or time is available. It is possible to improve the RDO performance. Along this direction, some proposals suggest an increase in the cases searched, e.g. by multi-pass coding [5], [6] or multi-QP optimization [7], [8]. A few other proposals focus on more accurate estimation of the elements of Equation-1. These include using different metrics for distortion [9] and using different values for λ [10], [11]. In all the above papers, the integer number of bits estimated is assumed to be accurate enough, especially if full entropy coding is carried out

for each attempted coding configuration. However, we note that the number of bits produced by arithmetic coding is not integer [12], [13]. In [14] we suggested that by using non-integer bit estimation in the RDO process of H.264/AVC, gains can be achieved. In this paper, we expand this concept to H.265/HEVC and our observations show that the improvement is more significant in H.265/HEVC than H.264/AVC. The reason appears to be a significantly larger number of modes to choose from in H.265/HEVC.

II. NON-INTEGER BIT ESTIMATION IN CABAC

For entropy coding, both H.264/AVC and H.265/HEVC employ Context-based Binary Arithmetic Coding (CABAC) [12], [13]. In this process each syntax element is represented as a sequence of binary symbols called bins. The value of each bin can be either equal to the Most Probable Symbol (MPS) or Least Probable Symbol (LPS). The proportional probability of LPS and MPS occurrences are predicted by $range_{lps}$ and $range_{mps}$ respectively, where the sum of these two is called $range$. Therefore, according to information theory [15], the estimated bits b needed to represent the bin would be:

$$b = \log_2 \left(\frac{1}{2} \right) = \begin{cases} \log_2 \left(\frac{range}{range_{lps}} \right), & bin = LPS \\ \log_2 \left(\frac{range}{range_{mps}} \right), & bin = MPS \end{cases} \quad (2)$$

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 a bin is equal to 1 bit. However, if $range_{lps} < range_{mps}$, more than 1 bit is needed for $bin = LPS$ and less than 1 bit is needed for $bin = MPS$.

In the arithmetic encoding process, $range$ is first initialized to 510 according to H.264/AVC and H.265/HEVC, and after encoding each bin it is updated to $range_{new}$ which is either $range_{lps}$ or $range_{mps}$. This procedure is illustrated in Figure-1. It continues until $range$ becomes smaller than 256. In that case, there is a renormalization phase in which $range$ is multiplied by 2 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.

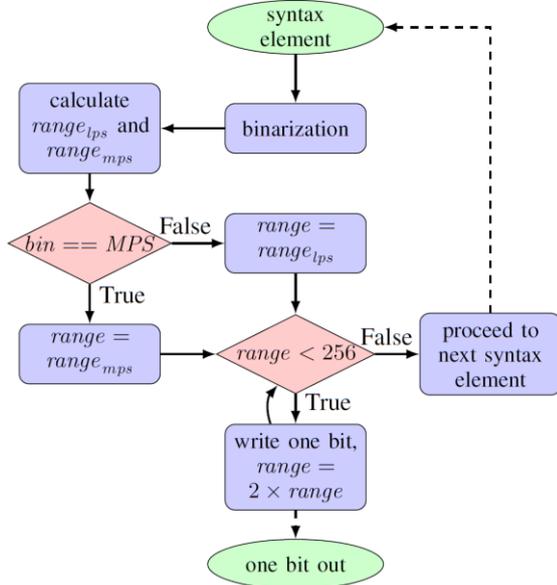


Fig. 1. Context-based Binary Arithmetic Coding (CABAC) encoding of syntax elements.

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:

$$b = \log_2(range) - \log_2(range_{new}) \quad (3)$$

where $range$ is before coding the bin and $range_{new}$ is before renormalization. This equation can also be worked out from Equation-1.

A. Fractional bit estimation for mode decision

For RDO mode decision, the total number of bits generated by a group of syntax elements for each mode is computed. In the original RDO implementation the integer number of bits is already measured separately by counting the written bits. We can simply add the fractional portion to this which is equal to:

$$f = \log_2(range_{begin}) - \log_2(range_{end}) \quad (4)$$

where the $range_{begin}$ and $range_{end}$ represent the value of range at the beginning of coding the group and after coding the

last syntax element including the renormalization stage. The f is within the range $(-1.0, +1.0)$:

- **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.

In this paper, we focus on the RDO implementation in the H.265/HEVC Test Model (HM) and H.264/AVC Joint Model (JM) reference software. In HM 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 intra-prediction mode: (64×64) , (32×32) , (16×16) , (8×8) and (4×4) .
- Mode decision for inter-partitioning mode: $(2N \times 2N)$, $(N \times 2N)$, $(2N \times N)$, $(N \times N)$ and the asymmetrical modes.
- Transform unit (TU) size decision: (32×32) , (16×16) , (8×8) and (4×4) .
- Coded Tree Unit (CTU) partitioning mode.
- Intra/Inter/Merge Decision for Coding Units (CUs).

In JM, on the other hand, the mode decision is carried out for the following steps:

- Mode decision for (16×16) intra-prediction mode.
- Mode decision for (4×4) intra-prediction mode.
- Mode decision for sub-macroblock inter-prediction mode: (8×8) , (8×4) , (4×8) and (4×4) .
- Final mode decision stage for a (16×16) macroblock.

It is clear that a significantly larger number of modes have to be chosen from in H.265/HEVC than H.264/AVC. That means there could be many more modes with close Lagrangian cost value where the fraction of the produced bits can influence the decision. To explore this, we have modified the JM and HM software and added the fractional bit estimation to the above stages. The cost function is evolved as follows:

$$J = D + \lambda \times (R + f) \quad (5)$$

where R is the integer number of bits and f is its fractional portion given by Equation-4.

III. SIMULATION RESULTS

The H.265/HEVC HM software version 16.3 and H.264/AVC JM software version 17.2 are modified to handle the proposed method, 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-Rate and BD-PSNR) between the proposed and the reference method are shown.

TABLE I. H.265/HEVC, NON-INTEGER BIT-ESTIMATION VS. REFERENCE, 4CIF@30Hz, IIIII PICTURES.

H.265/HEVC	BD-Rate (%)	BD-PSNR-Y (dB)	BD-PSNR-U (dB)	BD-PSNR-V (dB)
aspen	-3.15	+0.21	+0.06	+0.09
blue_sky	-3.03	+0.33	+0.15	+0.04
controlled_brun	-2.77	+0.23	+0.07	+0.09
crowd_run	-1.01	+0.23	-0.04	+0.02
life	-2.27	+0.22	+0.10	-0.02
riverbed	-3.95	+0.24	-0.02	+0.15
sunflower	-3.04	+0.24	+0.13	+0.10

TABLE II. H.265/HEVC, NON-INTEGER BIT-ESTIMATION VS. REFERENCE, 4CIF@30Hz, IPPPP PICTURES.

H.265/HEVC	BD-Rate (%)	BD-PSNR-Y (dB)	BD-PSNR-U (dB)	BD-PSNR-V (dB)
aspen	-0.44	+0.02	+0.01	+0.00
blue_sky	-2.38	+0.07	+0.10	+0.06
controlled_brun	-0.17	+0.01	+0.05	-0.04
crowd_run	-1.70	+0.01	+0.03	+0.07
life	-0.89	+0.03	+0.06	-0.03
riverbed	-13.25	+0.43	+0.28	+0.42
sunflower	-1.76	+0.11	+0.05	+0.03

TABLE III. H.265/HEVC, NON-INTEGER BIT-ESTIMATION VS. REFERENCE, 4CIF@30Hz, IBBBP PICTURES.

H.265/HEVC	BD-Rate (%)	BD-PSNR-Y (dB)	BD-PSNR-U (dB)	BD-PSNR-V (dB)
aspen	-3.33	+0.06	+0.05	+0.08
blue_sky	-4.36	+0.16	+0.21	+0.11
controlled_brun	-4.80	+0.13	+0.13	+0.10
crowd_run	-6.79	+0.08	+0.16	+0.21
life	-4.52	+0.10	+0.19	+0.09
riverbed	-0.26	+0.19	-0.19	+0.09
sunflower	-6.92	+0.19	+0.23	+0.24

The procedure detailed in [16] is used to calculate BD-PSNR and BD-Rate with $QP \in \{20,24,28,32,36,40\}$. For all tests the full-RDO mode was enabled and all the main-profile features of HM and JM in progressive format were enabled. For all-intra or intra-inter (GOP: IIIII, IPPPP and IBBBP) tests, 100 frames are coded (frame rates given in the table captions) with one intra update every second.

TABLE IV. H.264/AVC, NON-INTEGER BIT-ESTIMATION VS. REFERENCE, 4CIF@30Hz, IIIII PICTURES.

H.264/AVC	BD-Rate (%)	BD-PSNR-Y (dB)	BD-PSNR-U (dB)	BD-PSNR-V (dB)
aspen	-0.43	+0.02	+0.01	+0.01
blue_sky	-0.28	+0.02	+0.01	+0.01
controlled_brun	-0.49	+0.03	+0.01	+0.03
crowd_run	-0.30	+0.02	+0.01	+0.01
life	-0.28	+0.02	+0.01	+0.02
riverbed	-0.27	+0.02	+0.00	+0.01
sunflower	-0.33	+0.02	+0.01	+0.01

TABLE V. H.264/AVC, NON-INTEGER BIT-ESTIMATION VS. REFERENCE, 4CIF@30Hz, IPPPP PICTURES.

H.264/AVC	BD-Rate (%)	BD-PSNR-Y (dB)	BD-PSNR-U (dB)	BD-PSNR-V (dB)
aspen	-0.87	+0.02	+0.00	+0.04
blue_sky	-0.81	+0.01	-0.01	+0.04
controlled_brun	-0.10	+0.01	-0.02	+0.04
crowd_run	-7.53	+0.29	+0.20	+0.20
life	-3.92	+0.12	+0.14	+0.13
riverbed	-1.06	+0.02	+0.03	+0.03
sunflower	-1.35	+0.05	+0.05	+0.05

Tables-I, II and III (H.265/HEVC experiments), show the simulation results for H.265/HEVC for IIIII, IPPPP and IBBBP structures respectively. From these tables it can be seen that enabling the non-integer bit estimation reduces the average bit rate from 0.17% to 13.25% Note that the gain is less for intra-only videos. This is because for intra-pictures, the number of bits for each block or unit is more than that for inter-pictures.

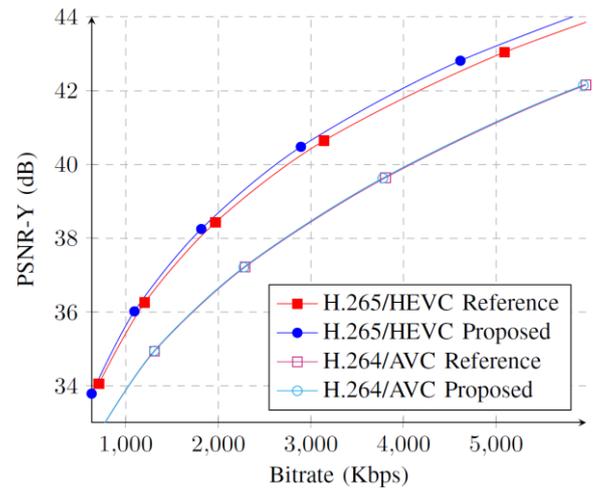


Fig. 2. Rate-Distortion curves proposed vs. reference aspen 4CIF@30Hz, IIIII pictures.

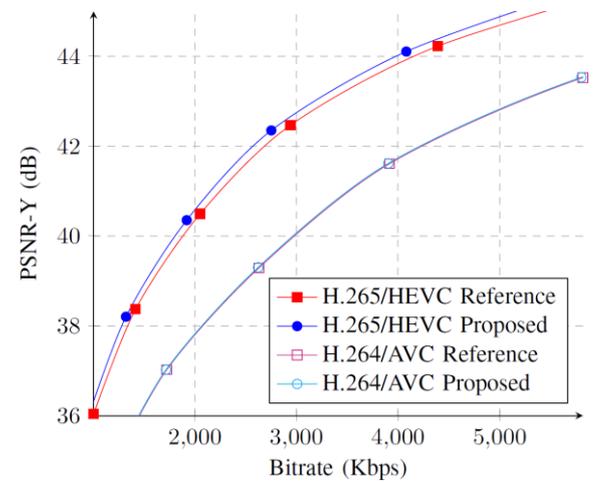


Fig. 3. Rate-Distortion curves proposed vs. reference sunflower 4CIF@30Hz, IIIII pictures.

Table-IV and V include the results of the same tests but with H.264/AVC. Also, to demonstrate the range of PSNR

values assessed, Figure-2 to 5 illustrate the PSNR-Y vs. Bitrate curves for videos in H.265/HEVC and H.264/AVC. Comparing H.265/HEVC with H.264/AVC for the same contents in Table-I to V, it can be seen that the gain offered by non-integer bit estimation is somewhat smaller in H.264/AVC on average even though highly content dependent. This is because the number of modes is significantly larger in H.265/HEVC than in H.264/AVC and accurate, non-integer bit estimation becomes more effective.

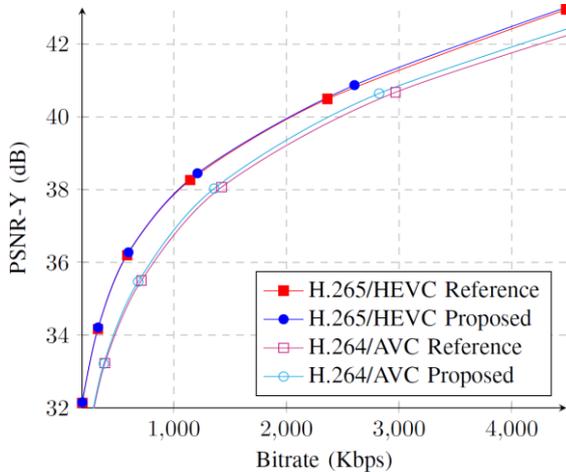


Fig. 4. Rate-Distortion curves proposed vs. reference life 4CIF@30Hz, IPPPP pictures.

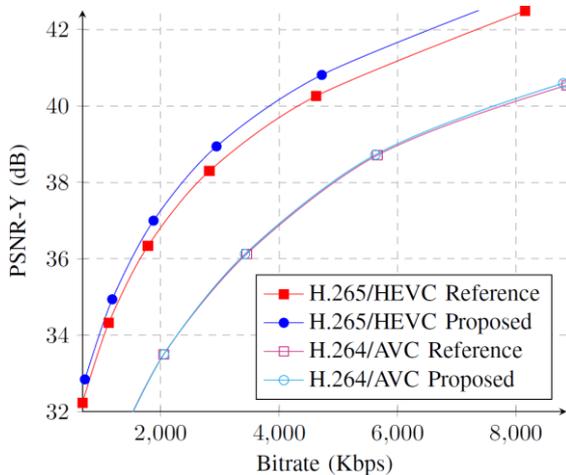


Fig. 5. Rate-Distortion curves proposed vs. reference riverbed 4CIF@30Hz, IPPPP pictures.

IV. CONCLUSION

This paper has explored some of the potentials of the Rate-Distortion Optimization (RDO) algorithm for video coding. For accurate bit estimation, a practical formula to compute the non-integer number of bits for coded syntax elements in H.264/AVC and H.265/HEVC CABAC is proposed and employed. Simulations show that using this more accurate bit representation in the RDO results in a better optimized mode decision without incurring a significant complexity overhead. This approach can be adopted by any encoder that applies RDO in the arithmetic coding mode. The result of these

improvements is shown to offer a more significant gain in the H.265/HEVC than H.264/AVC. This is due to a larger number of modes to choose from in HEVC.

REFERENCES

- [1] G. J. Sullivan, J. Ohm, W.-J. Han and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. IEEE, pp. 1649--1668, 2012.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H. 264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. IEEE, pp. 560--576, 2003.
- [3] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, no. IEEE, pp. 74--90, 1998.
- [4] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. IEEE, pp. 688--703, 2003.
- [5] C. An and T. Q. Nguyen, "Iterative rate-distortion optimization of H. 264 with constant bit rate constraint," *Image Processing, IEEE Transactions on*, vol. 17, no. IEEE, pp. 1605--1615, 2008.
- [6] J. You, C. Choi and J. Jeong, "Modified rate distortion optimization using inter-block dependence for H. 264/AVC intra coding," *Consumer Electronics, IEEE Transactions on*, vol. 54, no. IEEE, pp. 1383--1388, 2008.
- [7] B. Li, J. Xu, D. Zhang and H. Li, "QP refinement according to Lagrange multiplier for High Efficiency Video Coding," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, IEEE, 2013, pp. 477--480.
- [8] S. Wu, P. Liu, Y. Huang, Q. Liu and T. Ikenaga, "On bit allocation and Lagrange Multiplier adjustment for rate-distortion optimized H. 264 rate control," in *Multimedia Signal Processing, 2009. MMSP'09. IEEE International Workshop on*, IEEE, 2009, pp. 1--6.
- [9] Y.-M. Lee, Y.-T. Sun and Y. Lin, "SATD-based intra mode decision for H. 264/AVC video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. IEEE, pp. 463--469, 2010.
- [10] M. M. Ghandi and M. Ghanbari, "A Lagrangian optimized rate control algorithm for the H. 264/AVC encoder," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, IEEE, 2004, pp. 123--126.
- [11] J. Zhang, X. Yi, N. Ling and W. Shang, "Context adaptive Lagrange multiplier (CALM) for rate-distortion optimal motion estimation in video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. IEEE, pp. 820--828, 2010.
- [12] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. IEEE, pp. 620--636, 2003.
- [13] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. IEEE, pp. 1778--1791, 2012.
- [14] S. Im, M. Ghandi and C. Lam, "Non-integer bit estimation for rate-distortion optimized video coding," in *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, IEEE, 2012, pp. 88--89.
- [15] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. ACM, pp. 3--5, 2001.
- [16] K. Senzaki, K. Chono, H. Aoki, J. Tajime and Y. Senda, "BD-PSNR/Rate computation tool for five data points," in *Proc. of the Meeting of Joint Collaborative Team on Video Coding. Geneva, Switzerland:[sn]*, 2010.