Spherical Position Dependent Rate-Distortion Optimization for 360-degree Video Coding

Yuyang Liu*, Hongwei Guo*†, Ce Zhu*, Yipeng Liu*
*University of Electronic Science and Technology of China, Chengdu, China
†School of Engineering, Honghe University, Mengzi, China
yuyangliu@std.uestc.edu.cn, ghw202@foxmail.com, {eczhu, yipengliu}@uestc.edu.cn

***Abstract*—360-degree video in spherical format cannot be well handled by the conventional video coding tools. Currently, most of the 360-degree video coding methods first project the spherical video content onto a 2-dimensional plane and then compress the projected video using a conventional video codec. However, the projection conversion process will cause an irreversible conversion error, which indicates that the reconstruction quality of the projected video cannot fully represent that of the spherical video. In view of this, this paper proposes a spherical position dependent rate-distortion optimization (RDO) approach for 360-degree video coding. During the RDO process, spherical reconstruction quality is taken into consideration and calculated according to the spherical position of the pixels in each coding unit (CU). Furthermore, the Lagrangian multiplier and quantization parameter are adjusted accordingly. The proposed method is implemented on HEVC reference software HM-16.7. Experimental results show that the proposed method can achieve better coding performance, compared with HM-16.7.**

Keywords— 360-degree video, video coding, rate-distortion optimization, HEVC.

I. INTRODUCTION

360-degree video can provide immersive viewing-experience with a head-mounted display device. Specifically, a free view navigation can be easily attained through switching viewpoints. With such advances compared with traditional video, 360-degree video has been rapidly commercialized in a few applications, such as immersive gaming, social media and streaming, etc. However, 360-degree video with much bigger data volume has put a great pressure on cost-effective storage and transmission. Hence, high efficiency compression has become a vital issue for 360-degree video applications.

Capturing 360-degree video needs multiple cameras mounted in different directions and the acquired sequences are further stitched together in spherical format, i.e., a spherical video. For the spherical video compression, Tosic [1] proposed a dictionary learning and sparse representation approach, which needs to transmit both the dictionary atoms and sparse coefficients to the decoder side. However, the coding efficiency is not good enough in practice. Currently, the most commonly used compression flowchart for 360-degree video is shown in Fig. 1. The 360-degree video content in spherical format is first converted onto a 2-dimensional plane and the projected video is then compressed using a conventional video codec, such as HEVC (High Efficiency Video Coding) [2]. On the decoder side, the reconstructed video is converted into spherical format.

In order to further improve the coding efficiency, international standard organizations ITU-I VCEG (Video Coding Expert Group) and ISO/IEC MPEG (Motion Picture Expert Group) formed Joint Video Exploration Team (JVET) to explore promising video coding technologies for the next generation video coding standard Versatile Video Coding (VVC) [3]. As one part of the explorations, JVET established a 360Lib software package [4] for 360-degree video coding and processing, including projection format conversion and quality assessment. The 360Lib software package can be combined with HEVC and the developing video coding standard VVC. In this paper, 360Lib is used for sphere-to-plane mapping. As shown in Fig. 1, the spherical format is converted into equirectangular projection (ERP) format. Except for ERP format, 360Lib also supports other projection formats, such as cube-map projection format, octahedron projection format and so on. However, the sphere-to-plane mapping is non-linear yielding to an irreversible conversion error. Under a lossy compression scheme, reconstruction quality of the projected video cannot well represent that of the spherical video. Considering the coding efficiency in terms of rate-distortion (RD) performance, distortion from different kinds of source (projected video or spherical video) would possess different characteristics, which would lead to different coding options for a same video codec.

As a matter of the fact, the projection format conversion step shown in Fig. 1 is performed using non-uniform sampling across the whole picture. Taking ERP format as an example, areas near the north/south pole have denser sampling density compared with the equator areas. Using a fixed quantization parameter (QP) to encode ERP video will result in a relatively higher reconstruction quality near the poles compared with the other areas. In view of this, a few adaptive QP selection methods were proposed by considering the sampling density difference in the projected video. In [5-6], the QP value of each coding tree unit (CTU) is adjusted according to the sampling density in the corresponding spherical area. Such kinds of QP adaptation methods can achieve better RD performance based on quality assessment metrics for spherical video, such as WS-PSNR [7].

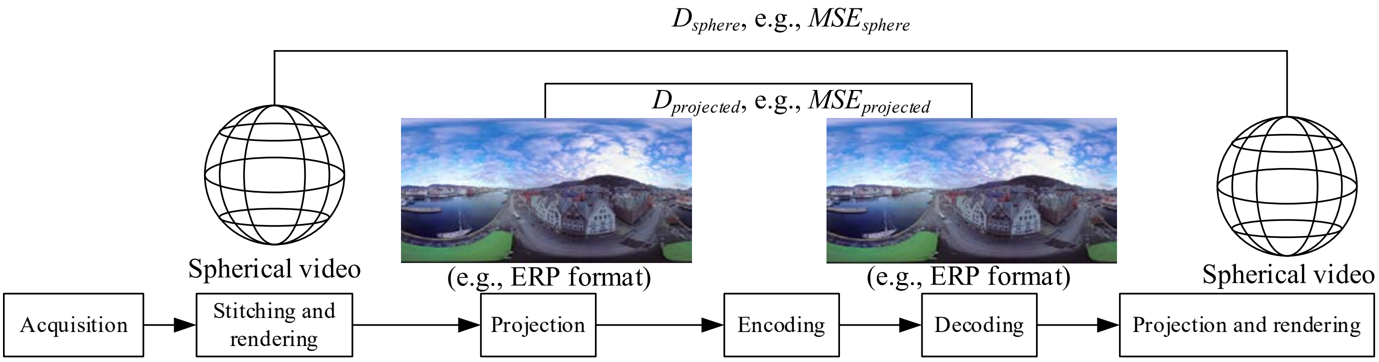


Fig. 1 Typical pipeline for 360-degree video coding

The core idea of this paper is the same as that in [5-6]. Compared with [5-6], the contributions of this paper are as follows. First, a spherical position dependent RDO approach is proposed by considering the sampling density in different spherical positions. Second, the Lagrangian multiplier and QP are adjusted at CU level. Experimental results show that the proposed method can achieve about 5.12% bit-rate savings based on WS-PSNR compared with HM-16.7 [8].

The remainder of this paper is organized as follows. In Section II, details of the proposed method are presented. In Section III, experimental results are presented. Section IV concludes this paper.

II. SPHERICAL POSITION DEPENDENT RATE-DISTORTION OPTIMIZATION

Rate-distortion optimization technique plays an important role in modern video encoders to trade off the coding bits and distortion. As for 360-degree video coding shown in Fig. 1, optimal coding options could be determined through RDO for the projected video, rather than for the spherical video. Hence, this paper aims to improve the reconstruction quality of spherical video while maintaining the coding bits of the projected video.

A. Traditional Rate-distortion Optimization

As one of the key technologies in video coding, rate-distortion optimization technology has been widely used in the hybrid coding framework, including CTU partition, mode decision, transform unit partition and quantization, etc. The goal of rate-distortion optimization is to minimize the compression distortion under a constrained bit budget, which can be written as below

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

where D and R denote the compression distortion and the bit cost and λ is the Lagrangian multiplier. As shown in Fig. 1, the input of the video codec is the projected video. In Eq. (1), D denotes the compression distortion of the projected video, which is denoted as $D_{projected}$. Meanwhile, in the following of this paper, the fidelity between the original and the reconstructed spherical video is represented by spherical distortion, which is termed as D_{sphere}.

B. Approximation for Spherical Distortion

As a matter of the fact, compression distortion $D_{projected}$ is caused by the quantization process in hybrid coding framework. Assuming the distribution of the quantized residual is uniform, $D_{projected}$ can be represented as

$$D_{projected} = \frac{QP_{step}^2}{12} = \frac{1}{12}\cdot 2^{\frac{QP-4}{3}}\,, \tag{2}$$

where QP denotes the QP value and QP_{step} is the quantization step. Eq. (2) is used in video encoders to demonstrate the relationship between compression distortion $D_{projected}$ and QP. Meanwhile, both $D_{projected}$ and D_{sphere} can be evaluated by means of mean square error (MSE), which can be written as

$$MSE = \sum_i \sum_j (x(i,j) - \hat{x}(i,j))^2\,, \tag{3}$$

where $x(i,j)$ and $\hat{x}(i,j)$ denote the pixel at position (i,j) in an uncompressed picture and the reconstructed picture, respectively. As shown in Fig. 1, $D_{projected}$ and D_{sphere} calculated by MSE is denoted as $MSE_{projected}$ and MSE_{sphere}, respectively. Except for MSE, a few quality assessment metrics for D_{sphere} have been proposed by JVET, such as WS-PSNR [7]. In this paper, WS-PSNR is used to evaluate D_{sphere}. Eq. (4) illustrates the definition of WS-PSNR

$$\begin{aligned} &WS\text{-}PSNR = 10\log(\frac{MAX_p^2}{WMSE}) \\ &WMSE = \sum_{i=1}^{W}\sum_{j=1}^{H}(x(i,j) - \hat{x}(i,j))^2 \cdot w(i,j) \end{aligned}\,, \tag{4}$$

where MAX_p is the maximum possible intensity of a frame and W and H denote the frame width and frame height. $w(i,j)$ is the weighting parameter at position (i,j) in a projected frame. For ERP format, $w(i,j)$ can be calculated as

$$w(i,j)=\frac{W(i,j)}{\sum_{i}\sum_{j}W(i,j)}, \quad (5)$$
$$W(i,j)=\cos((j-H/2+0.5)\cdot\pi/H)$$

where $W(i,j)$ is a scaling factor at position (i,j) and H is the height of ERP picture.

When encoding the projected video, only $D_{projected}$ is concerned. Aiming to improve the reconstruction quality of the spherical video, D_{sphere} should be taken into consideration during the RDO process. However, it is difficult and time-consuming to feedback D_{sphere} (e.g., MSE_{sphere}) to the video codec when encoding the projected video. One feasible solution is to approximate D_{sphere} through $D_{projected}$ during the encoding process. Since D_{sphere} is evaluated by WS-PSNR (or *WMSE*), Eq. (6) can be used to approximate D_{sphere} at CU level in the projected video, which is written as

$$WMSE\approx w\cdot MSE_{projected}^{CU}, \quad (6)$$

where w is a weighting parameter and $MSE_{projected}^{CU}$ denotes the mean square error of a coding unit. When w is one, *WMSE* is reduced to $MSE_{projected}$. Through Eq. (6), we can get the relationship between D_{sphere} and $D_{projected}$ at CU level which can be expressed as

$$D_{sphere}^{CU}\approx w\cdot D_{projected}^{CU}, \quad (7)$$

where $D_{projected}^{CU}$ denotes the quantization error of a CU and D_{sphere}^{CU} is the distortion of the spherical area. In this paper, weighting parameter w in Eq. (6) is set as the maximum scaling factor in each CU. Hence, weighting parameter w would vary in each CU and reflects the sampling density of the spherical area. Specifically, weighting parameter w could also be set as the scaling factor of top left position and the average of a CU according to [5-6].

C. *Quantization Parameter Adaptation*

Since the weighting parameter w of each CU would be different along with the spherical position, the QP value of each CU should be adjusted accordingly for better RD performance. Combining Eq. (2) and Eq. (7), we can obtain Eq. (8)

$$D_{sphere}^{CU}\approx w\cdot D_{projected}^{CU}=\frac{w}{12}\cdot 2^{\frac{QP_{base}-4}{3}}, \quad (8)$$

where QP_{base} denotes the QP value of current CU in a projected frame. Thus, the adjusted QP (denoted as QP_{CU}) can be obtained by

$$QP_{CU}=QP_{base}-QP_{offset}, \quad (9)$$
$$QP_{offset}=round(3\log_2 w)$$

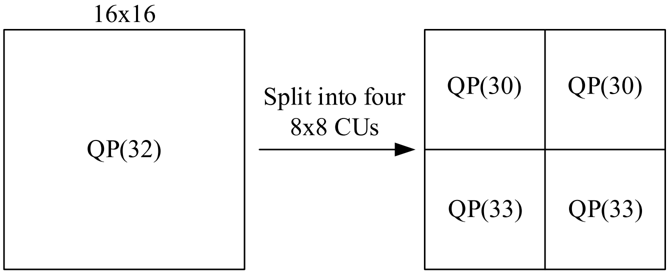


Fig. 2 A toy example to show QP adaptation results for CUs with different size

where $round(\cdot)$ represents rounding operation to keep the adjusted QP integral.

Since the proposed QP adaptation method is applied at CU level, CUs with different sizes would be encoded by different QPs. Fig. 2 illustrates a toy example of QP adaptation results. As shown in Fig. 2, after a further quad-tree partition, the QPs of four 8x8 CUs would be different from that of 16x16 CU. The RD cost comparison would be unfair for the projected video since CUs with different sizes would be encoded using different QPs. However, CUs in the projected video would cause different reconstruction-quality in the spherical video due to the various sampling density. Hence, a more uniform reconstruction quality could be achieved by using different QP values for different CUs in the projected video.

D. *Spherical Position Dependent RDO*

In order to achieve better RD performance, the Lagrangian multiplier should be updated according to the QP value of each CU. Meanwhile, the distortion part in Eq. (1) should be replaced by D_{sphere}, which can be further expressed as

$$J=w\cdot D_{projected}+\lambda\cdot R, \quad (10)$$

where w is the weighting parameter illustrated in Eq. (7). Then, we can get Eq. (11)

$$J=D_{projected}+\frac{\lambda}{w}\cdot R, \quad (11)$$

where the value of Lagrangian multiplier λ in Eq. (11) is the same as that in Eq. (1). From Eq. (11), Lagrangian multiplier λ is updated by

$$\lambda_{update}={}^{\lambda}\!/_{w}. \quad (12)$$

III. EXPERIMENTAL RESULTS

A. *Experimental settings*

Comparative experiments are conducted on HEVC test model (HM-16.7). Four 360-degree video sequences in ERP format with a resolution of 3840x1920 are selected from Common Test Condition published by JVET [9]. Video sequences are encoded under Low-delay P configuration using four QP values {22, 27, 32, 37}. 20 frames are encoded without any skipping frame. The other coding parameters are set as the default case. The Bjontegaard delta (BD)-rate is used to evaluate the coding performance [10]. In order to show the RD performance from different angles, two quality metrics (i.e., PSNR and WS-PSNR) are used. In Table I and Table II, a negative number means bit-rate savings, which represents the average bit-rate saving at the same WS-PSNR and PSNR, respectively.

Three weighting parameter selection methods are used in comparative experiments. In [5-6], weighting parameters are set as the scaling factor of the top left position and the average in a CU, respectively. In this paper, the weighting parameter is set as the maximum scaling factor in each CU. In the following, three weighting parameter selection methods are denoted as Top Left [5], MEAN [6] and MAX, respectively.

B. Results and Discussion

Experimental results of Luma component are shown in Table I and Table II. In Table I, it can be seen that the proposed method can outperform the state-of-the-art method. About 5.12% BD-rate savings can be achieved by the proposed method compared with HM-16.7 anchor. Meanwhile, experimental results shown in Table I indicate that the maximum scaling factor in a CU can reflect the sampling density difference better than the other two weighting parameters.

Table I

RD performance of Luma component in terms of BD-rate based on WS-PSNR (%)

Sequence	Top Left [5]	Mean [6]	Max
AerialCity	-4.98	-5.08	-5.00
DrivingInCity	-0.94	-1.04	-1.00
DrivingInCountry	-4.96	-5.13	-5.53
PoleVault	-8.50	-8.85	-8.97
Average	-4.85	-5.03	-5.12

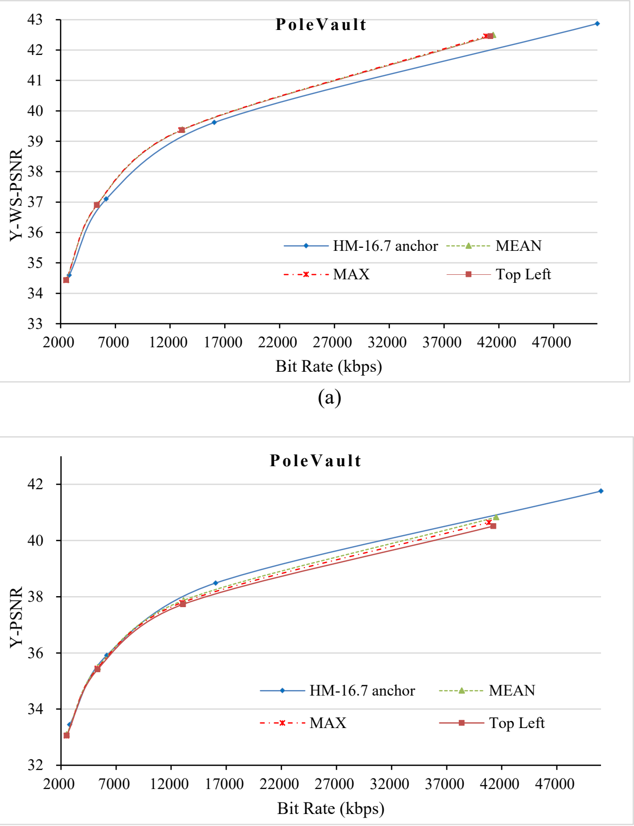
It would be quite different for the RD performance based on different quality assessment metric. Table II shows the RD performance based on PSNR. It can be seen that there is at least 3% BD-rate loss for three methods compared with HM-16.7 anchor.

Table II

RD performance of Luma component in terms of BD-rate based on PSNR (%)

Sequence	Top Left [5]	Mean [6]	Max
AerialCity	2.38	2.02	3.19
DrivingInCity	3.22	1.74	2.26
DrivingInCountry	7.25	6.95	9.98
PoleVault	7.91	2.90	5.04
Average	5.19	3.40	5.11

Fig. 3 illustrates the RD curves of Sequence "PoleVault". It can be seen that HM-16.7 consumes more bits at each QP point than the other three methods. The reason is that a smaller QP is used to encode each CU in HM-16.7 compared with the other three methods. From Fig. 3(a), it can be observed that the coding performance is significantly improved by three methods, i.e., about {8.5%, 8.85%, 8.97%} BD-rate savings in average, respectively. The result indicates that there exists much room to further improve the coding performance based on WS-PSNR and other objective quality assessment metrics except PSNR. Fig. 3(b) illustrates the RD curves based PSNR. It can be seen that HM-16.7 can achieve better RD performance based on PSNR compared with the other methods.



(a)

(b)

Fig.3 RD-curves of sequence PoleVault in ERP format. (a) RD curve based on WS-PSNR; (b) RD curve based on PSNR

IV. CONCLUSIONS

This paper proposed a spherical position dependent rate-distortion optimization approach for 360-degree video coding. In the proposed method, the reconstruction quality of spherical video was first represented by the compression distortion of projected video through linear approximation. A weighting parameter was then used to denote the sampling density of the spherical video and changed with the spherical position. Meanwhile, quantization parameter and Lagrangian multiplier were adjusted at CU level according to the weighting parameter. Experimental results show that the proposed method can achieve 5.12% bit-rate savings in average based on WS-PSNR. It should be pointed out that weighting parameter plays an important role in RDO and the proposed weighting parameter selection method can reflect the sampling density of a CU better than the other kinds of methods. In the future work, we will further explore weighting parameter selection methods considering both the spherical position information and video content.

ACKNOWLEDGMENT

This work was supported in part by the Key Project of Sichuan Provincial Department of Science and Technology under Grant 2018JY0035, in part by the National Natural Science Foundation of China under Grant 61571102, in part by the Yunnan Local Colleges Applied Basic Research Project under Grant 2018FH001-056.

REFERENCES

[1] I. Tosic and P. Frossard, “Low bit-rate compression of omnidirectional images,” *in Picture Coding Symposium*, 2009, pp. 1–4.
[2] G.J. Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuit. Sys. Video Techn.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
[3] J. Chen, Y. Chen, M. Karczewicz, X. Li, H. Liu, L. Zhang, and X. Zhao, “Coding tolls investigation for next generation video coding,” ITU-TSG16, Doc. COM16-C806, Feb. 2015.
[4] Y. Ye, E. Alshina, J. Boyce, “Algorithm descriptions of projection format conversion and video quality metrics in 360Lib,” JVET-H1004, Oct. 2017.
[5] X. Xiu, Y. He, Y. Ye, “An adaptive quantization method for 360-degree video coding,” *proc. SPIE 10752, Applications of Digital Image Processing XLI*, 107520X, Sep. 2018.
[6] Y. Sun, L. Yu, “AHG8: stretching ratio based adaptive quantization for 360 video,” JVET-F0072, Hobart, AU, Mar. 2017.
[7] Y. Sun, A. Lu, L. Yu, “WS-PSNR for 360 video quality evaluation,” MPEG document m38551, Geneva, Switzerland, Jun. 2016.
[8] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.
[9] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, “JVET common test conditions and evaluation procedures for 360 video,” Document JVET-D1030, Oct. 2016.
[10] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” VCEG-M33, Mar. 2001.