Geometry-based Motion Vector Scaling for Omnidirectional Video Coding

Ramin Ghaznavi-Youvalari, Alireza Aminlou

Nokia Technologies, Tampere, Finland
Email: firstname.lastname@nokia.com

***Abstract*—Virtual reality (VR) applications make use of 360° omnidirectional video content for creating immersive experience to the user. In order to utilize current 2D video compression standards, such content must be projected onto a 2D image plane. However, the projection from spherical to 2D domain introduces deformations in the projected content due to the different sampling characteristics of the 2D plane. Such deformations are not favorable for the motion models of the current video coding standards. Consequently, omnidirectional video is not efficiently compressible with current codecs. In this work, a geometry-based motion vector scaling method is proposed in order to compress the motion information of omnidirectional content efficiently. The proposed method applies a scaling technique, based on the location in the 360° video, to the motion information of the neighboring blocks in order to provide a uniform motion behavior in a certain part of the content. The uniform motion behavior provides optimal candidates for efficiently predicting the motion vectors of the current block. The conducted experiments illustrated that the proposed method provides up to 2.2% bitrate reduction and on average around 1% bitrate reduction for the content with high motion characteristics in the VTM test model of Versatile Video Coding (H.266/VVC) standard.**

***Index Terms*—Video coding, H.266/VVC, 360° video, motion vector**

I. Introduction

Omnidirectional spherical content cover 360° field-of-view (FOV), hence are widely used in virtual reality (VR) applications. In order to use the modern video coding standards such as High Efficiency Video Coding (H.265/HEVC) [1] or Versatile Video Coding (H.266/VVC) [2] for compressing such video, these spherical content are projected onto a two-dimensional (2D) image plane.

Equirectangular projection (ERP) format is among the commonly used formats for omnidirectional content. However, the resulted 2D projection suffers from deformations which caused by different sampling properties in different parts of the ERP image plane. These deformations are more severe in the polar areas. Consequently, current video coding standards are not capable of efficiently modeling such motion behavior in the motion estimation and compensation processes. As a result, the magnitude and direction of the motion vectors in a certain area can vary a lot. Figure 1 demonstrates an example of such motion behavior in different regions of the ERP video. As can be seen, the magnitude and direction of the motion vectors of blocks in a certain area are changing. This is more severe particularly in the blocks that are closer to the polar areas. Due to this non-uniform motion behavior, the motion vector difference (MVD) between blocks would be large and this leads to increase in bitrate of such content.

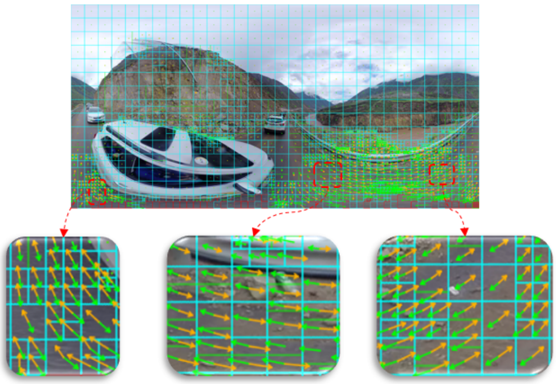


Fig. 1: Region-wise motion behavior in 360° ERP

In video coding algorithms, the motion information of the spatially neighboring blocks are used as predictors for coding the motion information of the current block. This process is applied in Advanced Motion Vector Prediction (AMVP) and Merge coding tools [1], [2]. In this work, an adaptive motion vector scaling method is proposed in order to provide a uniform motion behavior for efficient motion vector coding purposes. The proposed method calculates and applies a scaling factor for the motion vectors of the spatial neighbors based on the location of the current and neighbor block(s) in the 360° ERP video. The scaled motion vectors replace the existing predictor candidates in the AMVP and Merge coding tools.

The conducted experiments illustrated that the proposed motion vector scaling method provides up to 2.2% bitrate reduction and on average by around 1% bitrate reduction in the sequences with high motion. Moreover, there is no bitrate increase observed in the stationary sequences when the proposed method is applied.

The remainder of the paper is organized as follows. Section II reviews the related work for coding the omnidirectional video. The proposed geometry-based motion vector scaling method is described in Section III. Section IV discusses the experimental results. Finally, Section V provides the conclusion of the work.

II. Related Work

Recent studies address the issues of the 2D projection deformations by investigating projection-specific tools for omnidirectional content.

A multiresolution motion estimation method is studied in [3], where the motion estimation for the omnidirectional content is considered in spherical domain. In [4], a translational motion model is proposed for cubemap projection formats which applies the motion estimation operation in the 3D spherical domain by projecting the current and reference blocks to the sphere. A method studied in [5] that considers the motion estimation and block matching processes in the spherical coordinates by using 8×8 block sizes. Such block size limitation is used for reducing the complexity overheads of the motion estimation.

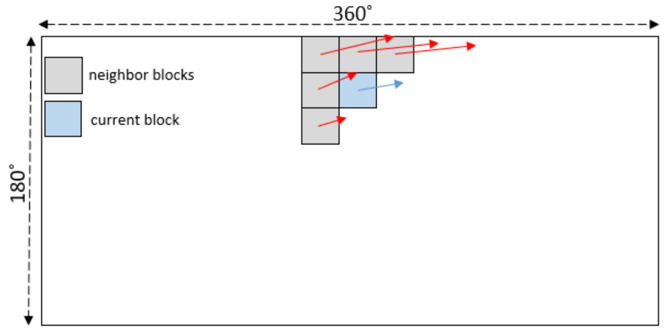
Rotational motion model is studied in [6], [7] where the motion estimation and compensation operations are done in the 3D spherical domain. Moreover, the authors proposed a radial pattern search for this operation in the spherical domain. Another motion model is introduced in [8], in which apart from the block-level motion vectors, pixel-wise motion vectors are also calculated in motion compensation process. This has been done by projection to spherical coordinates and depth calculation.

The above-mentioned methods improve the coding efficiency of the omnidirectional video by considering the near-real behavior of the motion by using the motion estimation and/or compensation operations in the spherical domain. However, since these operations are applied block-wise, the encoder and decoder complexities increase significantly. Thus, makes the deployment of such methods impractical in real-world scenarios. Furthermore, adapting the whole process of motion estimation and compensation of the current standards with the new approach requires significant changes in the standard chain of codecs.

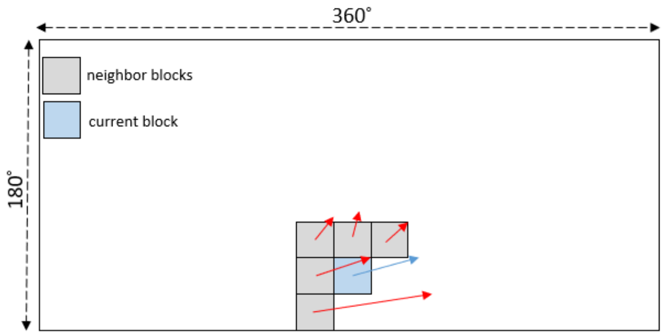
III. Proposed Geometry-based Motion Vector Scaling Algorithm

As mentioned in Section I, the equirectangular projection of 360° video results in region-wise oversamplings and deformations in the projection plane. Such characteristics in the content introduce issues in the compression process using the current video coding standards.

An example of this issue is illustrated in Figure 2. As can be seen, in blocks near the polar areas of the ERP video, the motion vector behavior changes a lot. For example in the northern areas (Figure 2a), the motion vector of current block is smaller than the above blocks which are closer to northern pole and include higher deformations. Moreover, the motion vectors of the below blocks in this area may have smaller motion vectors due to less deformations compared to the current block. Similar phenomenon can be observed for the blocks in southern polar areas (Figure 2b) in which the neighboring blocks motion vector behavior varies compared to the current block depending on the location in latitude of the 360° video. This motion vector variation is more severe if the content has

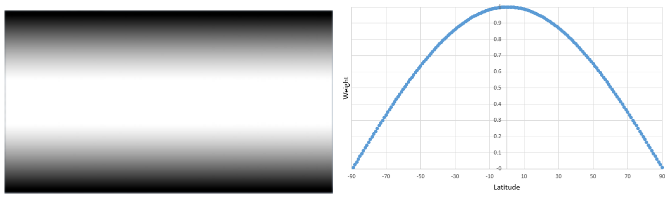


(a)



(b)

Fig. 2: Motion behavior of ERP video for blocks in the areas of a) north pole, b) south pole



(a) (b)

Fig. 3: Weight map in ERP

high motion. Such behavior of motion vectors result in larger motion vector difference (MVD) between the current block and neighbor blocks (that are used as motion vector predictors) and consequently, bitrate increase for MVDs.

A. *Geometry-based Motion Vector Scaling*

In this section, a method is proposed for scaling the neighboring motion vectors in order to make them suitable for predicting the motion vectors of the current block. This method considers the sampling density behavior in different locations of the ERP image. The sampling density distribution of the ERP content is shown in Figure 3a. Such characteristics is used for weighted quality assessment (WS-PSNR) of 360° content [9]. According to WS-PSNR scheme, for each pixel in the ERP domain, a weight depending on the latitude coordinate is assigned in a way that the samples in the polar areas have lower weights than the samples in equatorial areas. The weight derivation function is as below:

$$W[x,y] = cos(\frac{y - \frac{h}{2} + 0.5}{H} \times \pi)$$
$$Where: \begin{cases} 0 \leq x < w \\ 0 \leq y < h \end{cases} \quad (1)$$

In this equation, W is the calculated weight of the [x, y] pixel location in ERP and h and w are the height and width of the projection plane, respectively. The derived weights are illustrated in Figure 3b. Based on this representation, the equatorial areas include higher weights than the polar areas.

This formula can simulate the motion vector magnitude behaviors in different parts of the 360° video, in which the blocks that are closer to the polar areas have large motion vectors compared to their neighbors (as also shown in Figure 1). Therefore, this work takes advantage of weight map of ERP in order to calculate a scaling factor (SF) for scaling the neighboring motion vectors. In order to use such approach in block-level, the weights are derived according to the center location of each block by using (1) rather than the pixel-level weights. The reason being that the motion vectors are calculated in relation to the center points of blocks. The scaling factor is calculated as below:

$$SF = MaxWeight - (W_C - W_N) \quad (2)$$

In this formula, W_C and W_N are the calculated weights for current and neighbor blocks, respectively. *MaxWeight* represents the maximum weight in the ERP according to WS-PSNR [9] that is equal to 1. The calculated scaling factor is then applied to the horizontal and vertical components of the neighboring motion vectors with using equation (3)

$$\vec{MV}_{scaled} = round(\vec{MV} \times SF) \quad (3)$$

By using the described method, the motion vectors of the neighbor blocks are scaled up or down in a way that align with the central position of the current block. This results a more uniform motion behavior in a certain area of the 360° ERP video and smaller required MVD between the motion vector of current block and its neighbor. The scaled motion vectors are used as substitutes of the default unscaled candidates in AMVP and/or Merge lists.

The advantages of using the proposed geometry-based motion vector scaling method can be summarized as below:

- Efficient prediction of motion vectors particularly in the sequences with high motion.
- No compression loss for the content with stationary behavior (as the results in Section IV illustrate).
- Negligible complexity overhead.
- No extra signalling in the bitstream since the scaling process is applied in both encoder and decoder.
- Unlike the methods that are explained in Section II, the required changes in the video coding algorithm are quite simple and local.

IV. Experiments

A. Experimental Condition

The proposed motion vector scaling technique is implemented on top of the Versatile Video Coding (H.266/VVC) standard [2] test model VTM version 1.0 [10].

In order to evaluate the performance of the proposed method, 17 omnidirectional video clips in ERP format from JVET 360° test material [11] is used. The testset include video sequences in 4K, 6K, and 8K resolutions, each consisting 10 seconds i.e., equal to 300 and 600 frames for 30 fps and 60 fps sequences, respectively. The test materials are divided into two categories. The first category consists of sequences with high motion (e.g., camera/scene motion) and the second category contains content with lower motion.

The experiments conducted based on the proposed processing chain of Joint Video Experts Team (JVET) common test condition for 360° video [11]. According to the proposed processing chain, the high fidelity ERP sequences were downsampled to lower resolution versions (i.e., the coding resolution) prior to encoding process. The quality of the decoded content were assessed with different objective quality metrics proposed in the processing chain. Out of those, we used high fidelity ERP-PSNR, CPP-PSNR and WS-PSNR for quality evaluations. All the pre- and post-processing operations (e.g., resamplings, quality assessments, etc.) in the discussed processing chain are done by using the 360Lib software provided by JVET community [12]. Main profile Random Access (RA) configuration of common test condition [13] is used for encoding the test data. Moreover, the quantization parameters (QPs) are set to 22, 27, 32 and 37.

The performances were analyzed by using the well-known *Bjϕntegaard* Delta Bitrate (BDBR) criterion [14] for luma and chroma pictures. The negative values are the indications of how much the bitrate is decreased in the same peak signal-to-noise ratio (PSNR). Similarly, the positive values show the bitrate increment for the same quality level.

B. Analysis of the Results

Table I presents the results of the proposed motion vector scaling method when it is applied in both AMVP and Merge coding tools compared to the reference in different quality metrics. As can be seen from the table, the proposed method provided consistent bitrate reduction for the high motion category on average by 1% for luma and 0.8% for chroma components.

In the low motion category of sequences, the proposed method does not have any impact in bitrate reduction or increase. This is an expected performance behavior, since the proposed method is designed for reducing the bitrate of the sequences that the motion behavior is in a way that the conventional prediction methods are not capable of modeling it. Furthermore, as the results illustrate, similar rate-distortion performances were observed in all the objective quality metrics that are used for assessing the quality of the compressed omnidirectional video.

TABLE I: BD-Rate (%) performance of the proposed method for sequences with high and low motion

Category	Sequence	ERP PSNR			CPP-PSNR			WS-PSNR		
		Y	U	V	Y	U	V	Y	U	V
High Motion	ChairliftRide	-2.2	-2.0	-1.9	-2.1	-1.9	-1.8	-2.1	-2.0	-1.8
	Skateboard	-0.5	-0.4	0.0	-0.4	-0.4	-0.1	-0.4	-0.4	-0.1
	Balboa	-0.7	-0.3	-0.6	-0.7	-0.4	-0.8	-0.7	-0.4	-0.8
	BranCastle2	-0.6	-0.3	-0.4	-0.5	-0.3	-0.4	-0.5	-0.3	-0.4
	Landing2	-0.4	-0.1	-0.1	-0.4	-0.1	-0.2	-0.4	-0.1	-0.2
	DrivingInCountry	-1.7	-1.7	-1.6	-1.7	-1.7	-1.5	-1.7	-1.7	-1.5
	Bicyclist	-0.7	-0.4	-0.4	-0.7	-0.3	-0.4	-0.7	-0.3	-0.4
	Glacier	-2.1	-1.4	-1.5	-2.0	-1.4	-1.7	-2.0	-1.5	-1.7
	Building	-0.6	-0.4	-0.5	-0.6	-0.3	-0.5	-0.6	-0.3	-0.5
Low Motion	Gaslamp	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Trolley	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	KiteFlite	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Harbor	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1
	Broadway	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.1
	AerialCity	-0.1	0.0	0.0	-0.1	0.0	0.0	-0.1	0.0	0.0
	DrivingInCity	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.2	0.0
	Paramotor	-0.1	-0.1	-0.2	-0.1	-0.1	-0.2	-0.1	-0.1	-0.2
	High Motion Average	**-1.0**	**-0.8**	**-0.8**	**-1.0**	**-0.8**	**-0.8**	**-1.0**	**-0.8**	**-0.8**
	Low Motion Average	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
	Overall	**-0.6**	**-0.4**	**-0.4**	**-0.6**	**-0.4**	**-0.4**	**-0.6**	**-0.4**	**-0.4**

TABLE II: Average encoding and decoding complexities (%) of the proposed method compared to reference

Category	High motion	Low motion
Encoding complexity	99.1%	100.6%
Decoding complexity	101.5%	101.3%

General observation from the results is that, the geometry-based motion vector scaling method provides better bitrate reduction for the sequences that have higher global motion particularly in the polar areas of the 360° scene. For example, *ChairliftRide*, *Glacier* and *DrivingInCountry* sequences have higher global motion and as a result the proposed method performs better (around 2% bitrate reduction) in these sequences compared to other content.

Table II illustrates the average complexities in terms of encoding and decoding runtimes when the proposed method is applied compared to the reference. As it can be observed, the proposed method has negligible impact in the encoder/decoder complexity overhead for both categories of testsets.

V. Conclusion

In this work, a geometry-based motion vector scaling technique was proposed for handling the issue of large and non-uniform motion behavior of the equirectangular projected 360° content. The proposed method, based on the location of the block in the 360° image plane, derives a scaling factor between the current and neighbor coding blocks and applies a scaling method to the neighboring motion vectors that are used as predictors. The proposed motion vector scaling technique provided consistent gain in the sequences with higher motion by up to 2.2% bitrate reduction in the best cases and on average by 1% and 0.8% for luma and chroma components, respectively. Moreover, the proposed method had no impact in the rate-distortion performance of the stationary sequences. From the complexity point of view, this method had negligible impact in the encoding and decoding runtimes in all test sequences.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, *et al.*, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] B. Bross, "Versatile Video Coding (VVC) draft 1," *MPEG Joint Video Exploration Team*, vol. JVET-J1001-v21, Apr. 2018.

[3] I. Tosic, I. Bogdanova, P. Frossard, and P. Vandergheynst, "Multiresolution motion estimation for omnidirectional images," in *Signal Processing Conference, 2005 13th European*, pp. 1–4, IEEE, 2005.

[4] L. Li, Z. Li, M. Budagavi, and H. Li, "Projection based advanced motion model for cubic mapping for 360-degree video," *arXiv preprint arXiv:1702.06277*, 2017.

[5] F. De Simone, P. Frossard, N. Birkbeck, and B. Adsumilli, "Deformable block-based motion estimation in omnidirectional image sequences," in *19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, IEEE, 2017.

[6] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "Rotational motion model for temporal prediction in 360 video coding," in *19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, IEEE, 2017.

[7] B. Vishwanath, , K. Rose, Y. He, and Y. Ye, "Rotational motion compensated prediction in HEVC based omnidirectional video coding," in *Picture Coding Symposium (PCS)*, IEEE, 2018.

[8] Y. Wang, L. Li, D. Liu, F. Wu, and W. Gao, "A new motion model for panoramic video coding," in *International Conference on Image Processing (ICIP)*, pp. 1407–1411, IEEE, 2017.

[9] S. Yule, A. Lu, and Y. Lu, "WS-PSNR for 360 video objective quality evaluation," *MPEG Joint Video Exploration Team*, vol. 116, 2016.

[10] "Versatile Video Coding (VVC) reference software VTM. Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute." https://jvet.hhi.fraunhofer.de/, July 2018.

[11] E. Alshina, J. Boyce, A. Abbas, and Y. Ye, "JVET common test conditions and evaluation procedures for 360 degree video," *JVETG1030, m41362, Aug*, 2017.

[12] Y. Ye, E. Alshina, and J. Boyce, "Algorithm descriptions of projection format conversion and video quality metrics in 360Lib," *Joint Video Exploration Team of ITU-T SG*, vol. 16, 2017.

[13] J. Boyce, K. Suehring, X. Li, and V. Seregin, "Common test conditions and software reference configurations," in *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, 2018.

[14] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," *VCEG-M33*, 2001.