A LOW-COMPLEXITY VIDEO ENCODER FOR EQUIRECTANGULAR PROJECTED 360 VIDEO CONTENT

Bappaditya Ray[1,2], *Joel Jung*[1] *and Mohamed-Chaker Larabi*[2]

[1]Orange Labs
[2]CNRS, Univ. Poitiers, XLIM, UMR 7252, Poitiers, France
{bappaditya.ray, joelb.jung}@orange.com
{bappaditya.ray, chaker.larabi}@univ-poitiers.fr

ABSTRACT

360-video is gaining a lot of interest because of the immersive feeling brought by such a technology. Several projection formats are used to represent this type of content. Equirectangular projection (ERP) is one of the most widely used projection scheme for 360 panoramic content. The main drawback of ERP is its latitude dependent sampling density unlike conventional 2D content. Consequently, conventional 2D codecs such as HEVC are not optimal for the coding of ERP projected 360 content. To cope with this dependency, this work proposes an adaptation of motion vector resolution and minimum width of the coding block depending on its latitude. Experimental results show up to 0.5% BD-rate savings for motion contained sequences with 15% encoding time reduction in random access configuration.

Index Terms— 360 video, equirectangular projection, adaptive MV resolution, HEVC

1. INTRODUCTION

In recent days, 360-videos are gathering a lot of attention, as the same is often associated with virtual reality (VR). 360-video provides 360×180 degrees of field of view. So, users have the flexibility to watch in all directions (with head rotation), which provides a feel of immersion. For this, a very high spatial resolution is required (4K and above) to maintain a sufficient visual quality. Therefore, it also raises a significant challenge for video compression technology in terms of bandwidth requirement and processing power. As a result, the development of new compression tools specific to 360-video has become of paramount importance.

Spherical representation is the natural way of representing a 360 scene. However, this representation can not be efficiently compressed using a standard 2D video coder. To cope with this issue, 360 scenes are mapped to a 2D plane in multiple ways with the help of different projection schemes such as the equirectangular (ERP), cubemap (CMP), octahedron, icosahedral, etc. Among the aforementioned projections, ERP and CMP are the most widely used in 360 content creation processes. Every projection generates specific geometric distortions making the conventional 2D video coders, such as HEVC [1], sub-optimal for compressing such content. Moreover, codec computational complexity is an important issue when dealing with high resolution 360 panoramic video content.

Several works address efficient encoding of ERP and CMP. For instance, in [2], a homography based face padding mechanism for improved motion compensation of CMP is presented. Similarly, Li et al. use co-projection plane based 3D padding to derive the pixels outside face boundaries [3]. Relying on the sphere to CMP mapping, Li et al. propose a model to derive the motion of each CMP block [4]. All those methods yield improvements for sequences with global motion. Zhou proposes an unrestricted motion compensation scheme for ERP based on the idea that left and right boundaries of the reference picture are not discontinuous due to spherical continuity [5]. While addressing ERP and CMP coding, He et al. propose geometry padding based reference pixel derivation for intra prediction [6].

In [7], a spherical rate-distortion optimization (RDO) scheme for ERP is proposed by taking into account the latitude based variation of sampling density associated with the used projection scheme. By considering that the polar regions have less interest in 360 scenes and with the aim to save bit budget, Budagavi et al. propose a region adaptive smoothing used as a preprocessing step prior to ERP coding [8].

ERP suffers from an increased sampling density at high latitude regions. Consequently, uniform motion vector (MV) resolution and coding unit (CU) partitioning are not optimal for coding this kind of content. In this paper, we propose to adapt the MV resolution and block partitioning structure depending on the latitude of the coding block. The proposed scheme of adaptive MV resolution is described in section 2 and the adaptation of block partitioning is presented in section 3. In section 4, experimental results are presented, followed by the conclusion in section 5.

2. LATITUDE DEPENDENT MV RESOLUTION ADAPTATION

In HEVC, quarter pixel MV resolution is applied uniformly to all coding blocks [1]. However in ERP, the sampling density of the polar region is higher than that of the equator. For a given latitude ϕ, the sampling density corresponds to $1/\cos(\phi)$ times that of equator ($\phi = 0$). In other words, the content density in high latitude regions is generally lower than at the equator. Consequently, uniform quarter pixel MV resolution might not be optimal as coarse MV resolution might be sufficient for the polar region.

Adaptive MV resolution has been an active research topic in the last decade [9–13]. In this section, we propose an adaptive approach dedicated to 360-video where MV resolution of a block is adapted using the latitude of the block. The main idea is to gradually decrease the MV resolution with respect to the increase of the latitude. However, ERP sampling density is increased only in the horizontal direction (the vertical sampling density is constant). Accordingly, it ap-

Table 1: Different latitude regions and corresponding resolutions for horizontal component of MV.

Region	Latitude	resolution
$Region_{MV_1}$	$0 \le \|\phi\| < TH_{MV_1}$	$1/4pel$
$Region_{MV_2}$	$TH_{MV_1} \le \|\phi\| < TH_{MV_2}$	$1/2pel$
$Region_{MV_3}$	$TH_{MV_2} \le \|\phi\| \le 90$	$1pel$

pears reasonable to adapt only the horizontal component of the MV resolution, keeping the resolution of the vertical component fixed to quarter pixel. Therefore, three different resolutions are used, *i.e*, quarter, half and integer pixel. Note that, the different resolutions do not necessarily need to be dyadic for this adaptation. However, dyadically distributed MV resolutions can be easily handled by using the set of interpolation filters already available at the codec.

The frame is divided into three regions and each region uses different resolutions for the horizontal component of MV. A set of thresholds $TH_{MV} = (TH_{MV_1}, TH_{MV_2})$ is defined to distinguish different regions, as illustrated in Table 1. It is later chosen empirically through an optimization process (see section 4.1).

3. LATITUDE DEPENDENT MINIMUM CU WIDTH ADAPTATION

In recent video codecs, variable CU size is employed to cope with the variation of spatial characteristics inside a frame. Generally, smaller CU size is used in regions having higher spatial variation and *vice versa*. Minimum and maximum CU sizes are signalled in the bitstream as a syntax element and they are constant for the whole frame.

Due to higher horizontal sampling density, ERP generally has less content density at polar regions. So, when moving away from the equator, the spatial variation becomes lower and the encoder tends to use bigger blocks. The minimum CU size may be increased without impacting the coding performance. Accordingly, the minimum CU size can be adapted to the latitude of the block.

3.1. Statistics of CU size distribution

The distribution of CU sizes for each coding tree block (CTB) row (indicator of latitude) in a coded bitstream is collected with the aim to explore latitude dependency (see Fig. 1). For this study, the HEVC reference software HM-16.6 is used with enabled quadtree plus binary tree (QTBT) partitioning structure for the flexibility of non-square CU sizes [14]. Its default structure uses separate partitioning for luma and chroma components for intra pictures and common partitioning for inter pictures. The statistics are collected over 12 different sequences from the dataset listed in Table 2. CU size distribution of luma component of intra pictures obtained from 4K sequences are shown on Fig. 2.

From Fig 2, it can be observed that the average CU width is low in the middle CTB rows (indicating close to the equator regions), and it increases gradually with the latitude. The average CU height does not significantly depend on the latitude (see Fig. 2). However, for some sequences (e.g, Trolley, Gaslamp, Harbor), the average CU width and height of north polar region (CTB rows near zero) are considerably higher. This can be explained for those sequences by the homogeneous nature of the north polar region (e.g. sky). So, the encoder mostly uses bigger CU size which results in higher average CU width and height. Similar phenomena are also observed for inter pictures.

Table 2: Test sequences.

Name	Resolution	Frame Rate
Skateboardinginlot	4096×2048	60
ChairliftRide	4096×2048	60
KiteFlite	4096×2048	30
Harbor	4096×2048	30
Trolley	4096×2048	30
Gaslamp	4096×2048	30
PoleVault	3328×1664	30
AerialCity	3328×1664	30
DrivingInCity	3328×1664	30
DrivingInCountry	3328×1664	30
Glacier	3328×1664	24
Bicycle	3328×1664	25

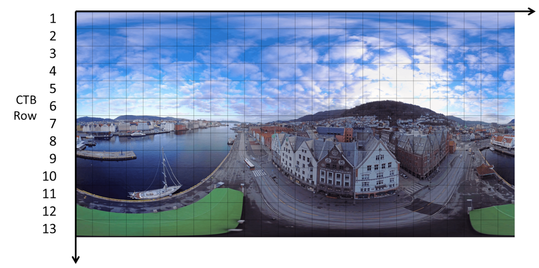

Fig. 1: CTB partitioning (128×128) of sequence *AerialCity* (3328×1664).

Overall the obtained statistics show both latitude dependent (geometric) and spatio-temporal effects. Moreover, the average CU width shows a latitude dependency (unlike CU height). This is reasonable as ERP has a high sampling density only in the horizontal direction. Accordingly, we propose to adapt the minimum CU width only. Note that this adaptation is targeted to cope only with the latitude dependent effect, which is solely specific to the geometric nature of ERP and not dependent on the spatio-temporal characteristics of the sequence.

3.2. Minimum CU width adaptation

In this section, minimum CU width adaptation is proposed. First of all, the latitude is derived from the center of each CTB. Based on the derived latitude value, the minimum CU width is defined empirically using a set of thresholds $TH_{CU} = (TH_{CU_1}, TH_{CU_2}, TH_{CU_3})$ (see section 4.2). Consequently, the signalling associated with the CU partitioning is modified as explained below.

According to QTBT scheme [14], quadtree leaf nodes are further partitioned by a binary tree structure, using symmetric horizontal or vertical splitting, as illustrated in Fig. 3. In our work, two parameters of QTBT are modified based on the minimum CU width (*minCUwidth*). First, *minQTsize* defining the minimum allowed quadtree leaf node size, *i.e.* when CU size is equal to *minQTsize*, further quadtree partitioning is not permissible. The second parameter is *minBTsize* representing the minimum binary tree width (*minBTwidth*) and minimum binary tree height (*minBTheight*). So, when the binary tree node has a width (*resp.* height) equal to *minBTsize*, no further vertical (*resp.* horizontal) splitting is allowed. The

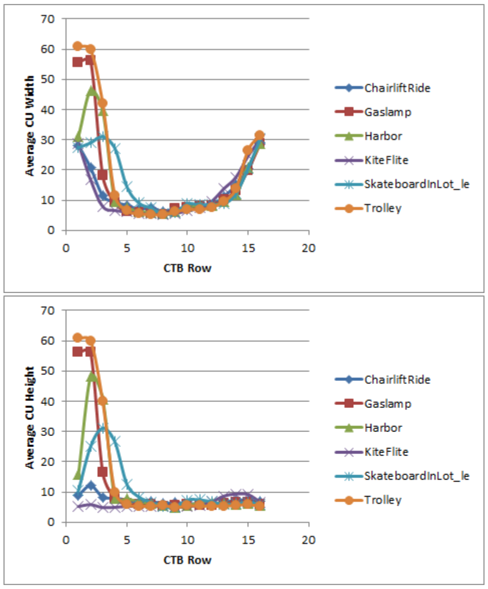

Fig. 2: Average CU size statistics of luma component of intra pictures: (top) average CU width, and (bottom) average CU height.

signalling modification is carried out by modifying the aforementioned two parameters (see Eq.1 and 2).

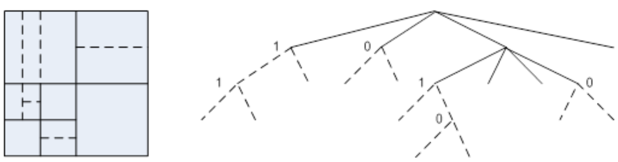

Fig. 3: QTBT partitioning scheme [14]. Solid lines indicate quadtree splitting and dashed lines indicate binary tree splitting. "0" and "1" respectively indicate horizontal and vertical splitting of the binary tree.

$$minQTsize = max(minQTsize, minCUwidth) \quad (1)$$

$$minBTwidth = max(minBTsize, minCUwidth) \quad (2)$$

Based on the modifications, the CU partitioning mechanism and the corresponding signalling are altered. Fig.4 provides an example of the CU partitioning and the corresponding signalling for *minCUwidth = 16*. It can be observed that for the modified QTBT, the flag *QTSplit* is not present anymore, since a 16×16 block cannot be quadtree partitioned to 8×8 blocks. Moreover, the flag *BTSplitHor* is also not present either, as it can be inferred that the splitting is horizontal (vertical splitting is not allowed due to *minCUwidth* restriction). Thus, the modification removes the redundant signalling of original QTBT.

4. EXPERIMENTAL RESULTS

Simulations are performed on the top of HM-16.6 with QTBT partitioning enabled. Tests are performed with main10 profile (10 bits for internal processing) for RA, LP (low-delay P), and LD (low-delay B)

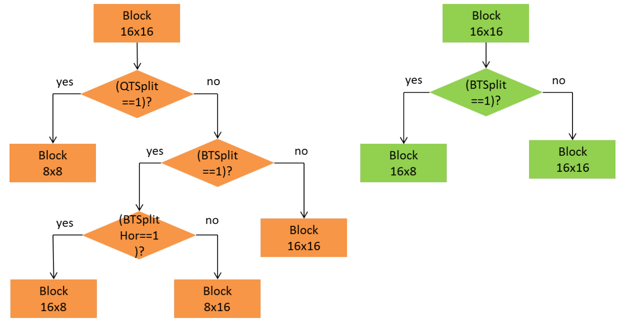

Fig. 4: Flowchart for CU partitioning and corresponding signalling: (Left) original QTBT, (Right) modified QTBT. The value of *minCUwidth* is 16.

coding configurations. A set of 12 sequences, described in Table 2, is selected for simulations where the first 50 frames are used. For objective quality evaluation, we use the WS-PSNR metric, one of the quality metric used by the Joint Video Exploration Team (JVET) activities [15, 16].

4.1. Latitude dependent MV resolution adaptation

This section provides implementation details and results about the MV resolution adaptation based on latitude. The optimization of MV resolution is done in two steps. First, only half and quarter pixel MV resolution layers are enabled. In this setup, several values of TH_{MV_1} are tested in order to find the optimal values (in terms of BD rate reduction). Next, the integer pixel layer is enabled and a set of values of TH_{MV_2} are tested in turn while keeping TH_{MV_1} fixed to the optimal value found previously. As optimal MV resolution may change depending on the coding configuration, the optimization steps are applied separately for different coding configurations, *i.e.* RA, LD and LP. This is due to the fact that optimal MV resolution depends on the nature of the inter prediction (uni-or bi-prediction) and the GOP structure. The optimal TH_{MV} is found to be equal to $(40, 60)$ for RA and $(60, 70)$ for LP and LD configurations. Note that, the optimal TH_{MV} is different for RA and low-delay (LP and LD) configurations. Low-delay configuration utilizes finer MV resolution compared to RA at high latitude regions. This is because low-delay configurations have a GOP size of 4, which is much smaller than the GOP size of 16 for RA. Due to this smaller GOP size, MVD magnitude is comparatively lower for low-delay configuration and thus fine MV resolution is beneficial for compression.

From the results shown in Table 3, it can be observed that the performance is sequence dependent. The sequences can be classified into two categories. The first category corresponds to sequences containing significant motion in high latitude regions such as *Glacier*, *Chairlift*, *DrivingInCountry* and *Bicycle*. On average, these sequences provide respectively 0.5%, 0.2% and 0.3% gain for RA, LP and LD configuration. The second category corresponds to sequences that have no significant motion in high latitude areas. Most of those sequences are captured with a static camera. They provide almost no gain. Overall, the proposed method reduces significantly the encoding runtime. This is because in high latitude regions, the proposed scheme avoids the computation of some fractional pixel MV positions during the process of motion estimation. Runtime reduction for RA configuration (11%) is higher than for LP (6%) and LD (7%).

Table 3: BD-rate and complexity comparison between the proposed scheme of MV resolution adaptation and the anchor for RA, LP and LD configurations using the test set of Table 2.

Configuration	RA	LP	LD
AerialCity	0.0%	-0.1%	-0.1%
DrivingInCity	0.0%	0.0%	0.0%
DrivingInCountry	-0.5%	-0.2%	-0.2%
PoleVault_le	-0.1%	0.0%	0.0%
Glacier	-0.9%	-0.5%	-0.6%
Bicycle	-0.2%	0.0%	0.0%
ChairliftRide	-0.6%	-0.2%	-0.3%
Harbor	0.0%	0.0%	-0.1%
KiteFlite	0.0%	0.1%	0.1%
Skateboardinginlot	0.0%	-0.1%	0.0%
Trolley	0.0%	0.0%	0.0%
Gaslamp	0.0%	0.0%	0.0%
Average (All)	-0.2%	-0.1%	-0.1%
Average (no Motion)	0.0%	-0.0%	0.0%
Average (Motion)	-0.5%	-0.2%	-0.3%
EncT	89%	94%	93%
DecT	99%	100%	99%

Table 4: Different latitude regions for the adaptation of MV resolution.

Region	Latitude	Minimum CU Width		
		I slice (Luma)	I slice (Chroma)	P/B slice
$Region_{CU_1}$	$0 \leq \|\phi\| < 60$	4	4	4
$Region_{CU_2}$	$60 \leq \|\phi\| < 70$	4	4	8
$Region_{CU_3}$	$70 \leq \|\phi\| < 80$	8	8	16
$Region_{CU_4}$	$80 \leq \|\phi\| \leq 90$	16	16	32

4.2. Latitude dependent minimum CU width adaptation

The performance of the minimum CU width adaptation based on latitude is addressed in this section. The optimization of TH_{CU} is carried out in a similar manner as illustrated in Section 4.1. The optimized threshold configuration is found to be $TH_{CU} = (60, 70, 80)$ and the minimum CU width associated with each region is provided in Table 4. Hence, the proposed scheme reduces the encoding runtime by 5%-7% while preserving the compression performance. The runtime reduction is due to the fact that the encoder does not evaluate the blocks having width smaller than *minCUwidth* for RDO.

4.3. Combined impact of minimum CU width and MV resolution adaptation

In this section, MV resolution and minimum CU width adaptation are enabled jointly. The results, given in Table 6 show that the effect of the two proposed tools is additive (both in terms of coding performance and encoding runtime reduction). The combination of the tools provides respectively 0.2%, 0.1%, and 0.1% gain for RA, LP, and LD configurations. In addition, the encoding runtime reduction is significant and is in average equal to 15%, 11%, and 12% respectively for RA, LP and LD configurations. Similar to the results of MV resolution adaptation, the gain is higher for sequences with motion (0.5%, 0.2%, and 0.2% respectively for RA, LP, and LD configurations).

Table 5: BD-rate and complexity comparison between the proposed scheme of minimum CU width adaptation and the anchor for RA, LP and LD configurations using the test set of Table 2.

Configuration	RA	LP	LD
AerialCity	-0.1%	-0.1%	0.0%
DrivingInCity	0.0%	-0.1%	0.0%
DrivingInCountry	0.0%	0.0%	0.0%
PoleVault_le	0.0%	0.1%	0.0%
Glacier	0.0%	0.0%	0.0%
Bicycle	0.0%	-0.1%	0.0%
ChairliftRide	0.1%	0.0%	-0.1%
Harbor	0.0%	0.0%	0.0%
KiteFlite	0.1%	0.1%	0.1%
Skateboardinginlot	0.0%	-0.1%	0.0%
Trolley	0.0%	0.0%	0.0%
Gaslamp	0.0%	0.2%	0.0%
Average (All)	0.0%	0.0%	0.0%
EncT	95%	94%	93%
DecT	100%	99%	99%

Table 6: BD-rate and complexity comparison between both schemes combined and the anchor for RA, LP and LD configurations using the test set of Table 2.

Configuration	RA	LP	LD
AerialCity	0.0%	-0.1%	-0.1%
DrivingInCity	-0.1%	0.0%	0.0%
DrivingInCountry	-0.4%	-0.2%	-0.2%
PoleVault_le	-0.1%	0.0%	0.0%
Glacier	-0.9%	-0.4%	-0.5%
Bicycle	-0.2%	-0.1%	0.0%
ChairliftRide	-0.6%	-0.2%	-0.3%
Harbor	0.0%	0.0%	-0.2%
KiteFlite	0.1%	0.3%	0.1%
Skateboardinginlot	0.0%	-0.2%	0.0%
Trolley	0.0%	0.0%	0.0%
Gaslamp	0.0%	0.1%	0.1%
Average (All)	-0.2%	-0.1%	-0.1%
Average (no Motion)	0.0%	0.0%	0.0%
Average (Motion)	-0.5%	-0.2%	-0.2%
EncT	85%	89%	88%
DecT	99%	99%	101%

5. CONCLUSION

Due to non-uniform (latitude dependent) sampling density of ERP, traditional 2D video coding tools are not optimal for compression of 360 content using such a projection. In this work, two techniques are proposed to adapt the coding tools depending on the latitude of the coding block in ERP frames. Depending on the latitude, the first technique involves the adaptation of horizontal component of MV resolution while the second adapts the minimum width of a CU. The former approach provides compression benefit and encoding runtime reduction and the latter provides encoding runtime reduction with almost no loss of coding performance. Moreover, the tools have an additive impact when they are used jointly. Overall, the proposed tools act significantly on encoding runtime reduction. This aspect is very beneficial for encoding high resolution 360 video content, as encoding time is one of the bottleneck in terms of latency and processing power. As a future work, both tools can be extended for other projection schemes. Furthermore, intra prediction tools can also be adapted based on the latitude for ERP projected 360 content.

6. REFERENCES

[1] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "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] J. Sauer, J. Schneider, and M. Wien, "Improved motion compensation for 360 video projected to polytopes," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 61–66.

[3] L. Li, Z. Li, X. Ma, H. Yang, and H. Li, "Co-projection-plane based 3-d padding for polyhedron projection for 360-degree video," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 55–60.

[4] L. Li, Z. Li, X. Ma, M. Budagavi, and H. Li, "Projection based advanced motion model for cubic mapping for 360-degree video," in *2017 IEEE International Conference on Image Processing (ICIP)*, September 2017, pp. 1–5.

[5] M. Zhou, "Unrestricted motion compensation for 360 video in erp format," in *Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. JVET-E0065, 2016, pp. 1–3.

[6] Y. He, Y. Ye, P. Hanhart, and X. Xiu, "Reference samples derivation using geometry padding for intra coding," in *Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. JVET-D0092, 2016, pp. 1–6.

[7] Y. Li, J. Xu, and Z. Chen, "Spherical domain rate-distortion optimization for 360-degree video coding," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 709–714.

[8] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, and A. Dickerson, "360 degrees video coding using region adaptive smoothing," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 750–754.

[9] S. Corrado, M. A. Agostini, M. Cagnazzo, M. Antonini, G. Laroche, and J. Jung, "Improving H. 264 performances by quantization of motion vectors," in *Proc. IEEE Picture Coding Symposium*, Chicago, USA, 2009, pp. 1–4.

[10] J. Ma, J. An, K. Zhang, S. Ma, and S. Lei, "Progressive motion vector resolution for HEVC," in *Proc. IEEE Visual Communications and Image Processing (VCIP)*, Kuching, Malaysia, 2013, pp. 1–6.

[11] Z. Wang, S. Wang, J. Zhang, and S. Ma, "Adaptive progressive motion vector resolution selection based on rate–distortion optimization," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 400–413, 2016.

[12] Z. Wang, J. Ma, F. Luo, and S. Ma, "Adaptive motion vector resolution prediction in block-based video coding," in *Proc. IEEE Visual Communications and Image Processing (VCIP)*, Singapore, 2015, pp. 1–4.

[13] B. Ray, J. Jung, and M. C. Larabi, "A block level adaptive mv resolution for video coding," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 49–54.

[14] J. Chen, E. Alshina, G. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 3," in *Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. JVET-C1001, 2016, pp. 1–30.

[15] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "JVET common test conditions and evaluation procedures for 360 video," in *Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. JVET-E1030, 2017, pp. 1–6.

[16] Y. Sun, A. Lu, and L. Yu, "WS-PSNR for 360 video objective quality evaluation," in *Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*. JVET-D0040, 2016, pp. 1–5.