Transform Domain Temporal Prediction and Geodesic Motion Compensation in Spherical Video Coding

Kruthika Koratti Sivakumar, Bharath Vishwanath, Kenneth Rose
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA, 93106
{ kruthika, bharathvishwanath, kenrose } @ucsb.edu

2020 54th Asilomar Conference on Signals, Systems, and Computers | 978-0-7381-3126-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF51394.2020.9443469

***Abstract*—This paper considers spherical videos dominated by camera motion, which are prevalent in many virtual and augmented reality applications whose deployment critically depends on efficient compression that is tailored to the signal characteristics. Existing approaches project the spherical video onto a plane (or planes) via one of several known projection geometries, followed by compression by a standard 2D codec. However, performance is compromised by the fact that motion in the projected domain is warped by the projection, and is not effectively compensated with the simple motion model employed. This paper leverages our recently proposed geodesic translation motion model to capture the exact pixel-wise motion on the sphere, and embeds it within a transform-domain temporal prediction framework, which is designed to disentangle spatial and temporal correlations. The approach circumvents a major shortcoming of standard temporal prediction, which effectively consists of simple pixel-copying (after motion compensation) from the reference frame, and thus largely ignores underlying spatial correlations. Transform-domain temporal prediction exploits the (spatial) decorrelating properties of the transform, accounts for both spatial and temporal correlations, and explicitly captures variations in temporal correlations across frequencies. We design the correlation filters by using an iterative open loop procedure that asymptotically converges to closed loop operation. Experimental results provide evidence that the overall approach, comprising geodesic motion compensation in conjunction with transform domain temporal prediction, offers considerable gains over the state-of-the-art.**

***Index Terms*—spherical video, transform-domain temporal prediction, HEVC**

I. Introduction

An omnidirectional video is a video that captures the entire surroundings and enables users to see in any direction. It is pivotal in all virtual reality related applications which are prevalent in education, healthcare, entertainment etc. In this paper, we focus on spherical videos dominated by translational camera motion, which are frequently encountered in many important applications like robotics and navigation systems. Given the growing usage of spherical videos and the enormous amounts of data generated, we need efficient compression techniques tailored to this class of videos.

Standard approaches project spherical videos onto plane(s) via several projection geometries (e.g., equirectangular projection (ERP), cubemap projection [1]). These projected videos are then compressed by regular 2D codecs such as HEVC [2]. However, the simple translation motion model used by standard codecs, and its affine extensions in [3] and [4], are highly suboptimal for projected spherical videos, since: (i) the projection introduces unintended warping, resulting in complex non-linear motion that is not captured by simple translation models, (ii) motion vectors in the projected domain lack a sound physical meaning due to the fact that object motion on the sphere does not map to horizontal or vertical straight lines in the projected domain, (iii) many typical polyhedral sphere-to-plane projections arrange multiple faces on the 2D frame by means of frame packing which introduces discontinuities that compromise subsequent motion compensation. Some recently proposed approaches in [5], [6] and [7] perform motion compensation on the sphere, but do not account for the perceived motion of objects due to camera motion.

To address these shortcomings, we proposed a geodesic motion model in [8] that perfectly captures the motion of objects due to camera motion. The model is based on the observation that with translation motion of the camera, surrounding objects are perceived to move along geodesics, all of which intersect at the points of intersection of the camera motion axis and the sphere. In this approach, a block of pixels to be temporally predicted is mapped to the sphere, the pixels are moved along their respective geodesics, and mapped back to the projected domain, where the prediction signal is obtained by interpolation, as necessary. While this approach captures the exact motion of pixels on the sphere, the prediction itself was done by the standard pixel copying technique, which ignores underlying spatial correlations, thus rendering it suboptimal. This shortcoming is circumvented in our current approach by an effective strategy within the framework of transform domain temporal prediction (TDTP) [9]. In TDTP, spatial and temporal correlations are properly decoupled, by first spatially decorrelating the block via a spatial transform (e.g., the discrete cosine transform) followed by temporal prediction *per transform coefficient*. Moreover, TDTP exploits the observation that transform coefficients of blocks along a motion trajectory exhibit higher correlation for lower frequencies and

the correlation decays for higher frequencies, the frequency dependent nature of which is masked in the pixel domain. Therefore, the proposed method first employs the geodesic motion model to derive the block of pixels from the reference frame. Then, instead of using this block as is for prediction, it proceeds to apply a spatial transform to spatially decorrelate samples, and performs temporal prediction in the transform domain. To realize the full potential of transform domain prediction, we need effective prediction filters that capture the variations in correlations across frequencies. Design of prediction filters is a challenging problem and is well known to suffer from instabilities due to closed-loop nature of the coder. To address this, we use an asymptotic closed-loop design paradigm [10] that is inherently stable due to its open-loop structure which nevertheless ensures asymptotic filter optimization for closed-loop operation. Thus, the proposed method benefits from precise motion compensation due to the geodesic motion model, and disentanglement of spatial and temporal correlation via carefully designed transform domain prediction filters, thereby overcoming the shortcomings enumerated before. It is important to emphasise that even though we report experimental results with the widely used equirectangular projection, the proposed method is applicable in conjunction with any projection format.

The rest of the paper is organized as follows. Section II provides an overview of the equirectangular projection (ERP) and the geodesic motion model from our earlier work and addresses the shortcomings of pixel domain temporal prediction. The proposed approach and algorithm are described in section III. Section IV covers offline design challenges and the ACL operation, followed by experimental results in section V and conclusions in section VI.

II. Background

In this section, we review equirectangular projection as an example projection format for spherical videos, then illustrate the perceived motion of objects on the sphere resulting from translational camera motion followed by an enumeration of drawbacks of performing temporal prediction in pixel domain.

A. Overview of Equirectangular Projection

ERP sampling is illustrated in Fig. 1. It maps longitudes to vertical straight lines and latitudes to horizontal straight lines. Thus, any point p on the sphere, with an elevation (pitch) θ and an azimuth (yaw) ϕ, is mapped to the position obtained on the 2D grid as the intersection of the vertical and horizontal lines corresponding to the same pitch and yaw (θ and ϕ, respectively) on the sphere. ERP maintains constant vertical sampling density. However, the horizontal sampling density increases as we move towards the poles. Conversion tools for various projections are available in [11].

B. Geodesic Translation Motion Model

In order to illustrate the perceived motion of surrounding objects on the sphere due to camera motion, let us consider

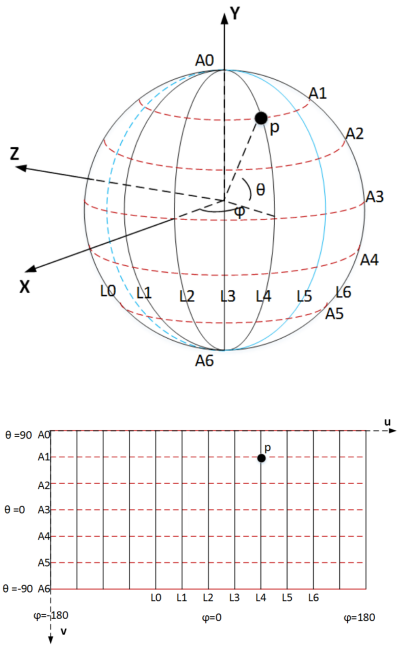

Fig. 1: ERP Sampling: on the sphere (top) and in two-dimensions (bottom)

a simple scenario illustrated in Fig. 2. A point P in the environment is viewed as its projection S on the sphere. Suppose the camera moves along the velocity vector v. Due to camera motion, the point P appears to move in space along PP'. The point P' is viewed as its projection S' on the sphere. Hence, it can be deduced that the arc SS' is the perceived motion of the point P on the sphere, due to translational motion of the camera. In our earlier work, we observed that the arc SS' is part of a geodesic, and all such geodesics intersect at the points where an axis along vector v intersects the sphere. In light of this observation, we employed a geodesic motion model that yields accurate temporal prediction by translating pixels along their respective geodesics on the sphere. In our previous

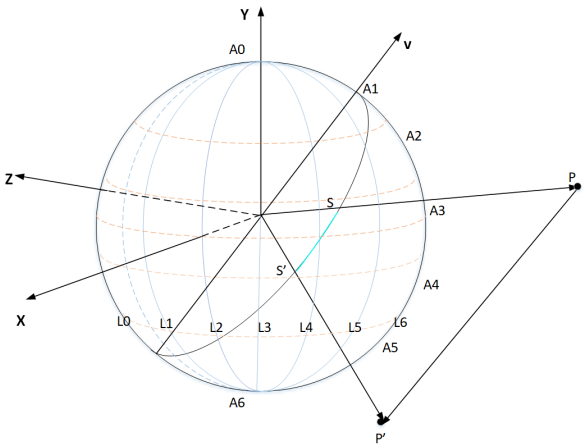

Fig. 2: Illustration of geodesic translation

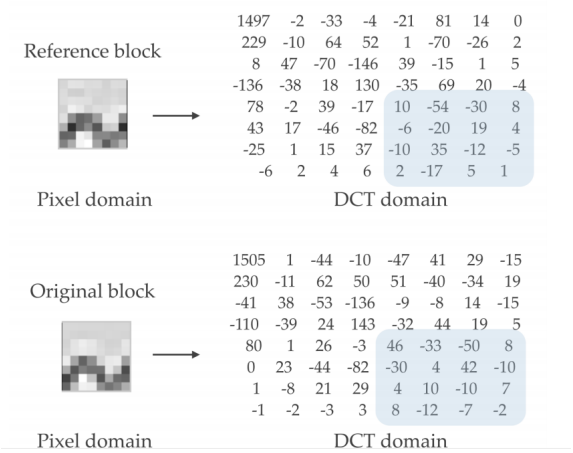

Fig. 3: Reference and original blocks in pixel and DCT domains

approach, after motion compensation using this motion model, prediction was performed by simple pixel copying, as done in standards, the drawbacks of which are discussed next.

C. Shortcomings of Pixel Domain Prediction

Conventional motion compensation is performed by simple block-based pixel copying. This completely ignores the underlying spatial correlation in the block. Moreover, transform coefficients of blocks that form a motion trajectory, have varying temporal correlations across frequencies. Fig. 3 shows an example pair of reference and original blocks in pixel and transform domains. Even though the blocks look similar in pixel domain, it can be seen in the DCT domain that temporal correlation varies across frequencies. Lower frequency transform coefficients show high temporal correlation whereas, higher frequency transform coefficients (shaded blue) have much lower temporal correlation. Simple pixel copying doesn't account for these variations in temporal correlation across frequencies and also ignores spatial correlations, rendering it sub-optimal. In order to overcome these drawbacks, we propose to employ the geodesic translation motion model in conjunction with TDTP which we describe next.

III. Proposed Geodesic Motion Compensation and Transform Domain Temporal Prediction

In this section, we explain how we perform transform domain temporal prediction in part A, followed by a detailed description of the algorithm employed for geodesic motion compensation with TDTP, in part B.

A. Transform Domain Temporal Prediction

In TDTP, we first achieve spatial decorrelation by transforming the block by DCT and then accounting for the temporal correlations of DCT coefficients. We model the evolution of each transform coefficient along a motion trajectory as a first order AR process. Let x_n denote a DCT coefficient in the current block in frame n and $\hat{x}_{n-1}$ the corresponding reconstructed coefficient in the reference block in frame $n-1$. The AR process is given by,

$$x_n = \rho \hat{x}_{n-1} + e_n \tag{1}$$

where, ρ is the prediction coefficient and e_n is the innovation. The prediction for each DCT coefficient is

$$\tilde{x}_n = \rho \hat{x}_{n-1} \tag{2}$$

We minimize the mean squared prediction error given by,

$$J = E((x_n - \rho \hat{x}_{n-1})^2) \tag{3}$$

to obtain the optimal prediction coefficient ρ_{opt} for each DCT coefficient as,

$$\rho_{opt} = \frac{E(x_n \hat{x}_{n-1})}{E(\hat{x}_{n-1})^2} \tag{4}$$

We note that the conventional pixel copying technique for temporal prediction is equivalent to employing TDTP with $\rho = 1$ for all frequencies.

B. Overall Approach

Fig. 4 shows a standard coding pipeline for spherical videos. Consider a current block of pixels in the projected domain which is to be predicted using temporal prediction. The proposed prediction consists of the following steps:

- *Sphere mapping*: We map all pixels in the block to their corresponding locations on the sphere using the inverse projection mapping procedure. Let (ϕ_{ij}, θ_{ij}) denote the yaw and pitch coordinates of the $(i,j)^{th}$ pixel in the block, in the orientation in which the polar axis of the sphere coincides with the camera velocity vector.
- *Geodesic translation*: For a given motion vector (m, n), we move the pixels along its geodesic as

$$\phi'_{ij} = \phi_{ij} + m\Delta\phi_s, \quad \theta'_{ij} = \theta_{ij} + n\Delta\theta_s$$

  where, $\Delta\phi_s$ and $\Delta\theta_s$ are predefined step sizes.
- *Mapping back to projected domain*: We map the pixels at $(\phi'_{ij}, \theta'_{ij})$ back to the 2D projected plane and perform interpolation as required to obtain an *intermediate block* of pixels. It should be noted that in our previous approach, we used this intermediate block of pixels as our final prediction signal to perform pixel domain temporal prediction.
- *Prediction in DCT Domain*: Finally, we transform this intermediate block of pixels and apply the TDTP filters to obtain the transform domain prediction signal.

Effective TDTP filters are crucial to realize the full potential of the proposed approach. The design of these filters poses challenges which we consider next.

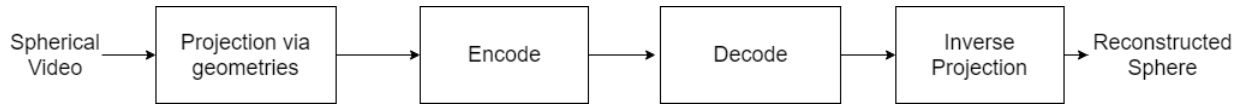

Fig. 4: Standard spherical video codec

IV. Asymptotic Closed Loop Design

The design of predictors is a challenging task due to complex interplay between predictions and reconstructions. Predictors are used with reconstructions and predictors in turn depend on the reconstructions. Due to this inter-dependency, we resort to iterative design paradigms in order to design our TDTP predictors.

In the standard closed-loop design, the predictors designed for a given reconstruction set is used with the reconstructions in the next iteration causing statistical mismatch. To illustrate this, let us consider a sequence of N DCT coefficients (corresponding to a given frequency), from blocks along a motion trajectory, denoted as, x_1, x_2, ... , x_N, where the subscript denotes the frame number. Let us consider the first design iteration with closed-loop initialization. The first frame is intra-coded to produce reconstructed coefficient $\hat{x}_1$. The DCT coefficient x_2 is predicted from $\hat{x}_1$ as $\tilde{x}_2 = \hat{x}_1$, assuming we initialize the prediction coefficient to $\rho = 1$ for all frequencies. The transform domain residue for frame 2 is given by, $r_2 = x_2 - \tilde{x}_2$, and the reconstruction of x_2 denoted as $\hat{x}_2$ is obtained as $\hat{x}_2 = \hat{r}_2 + \tilde{x}_2$, where $\hat{r}_2$ is the reconstructed residue of the DCT coefficient of the pixel in frame 2. This process continues from frame 3 to frame N. Finally, based on the statistics observed in the above iteration, the optimal prediction coefficient at the given frequency is determined as,

$$\rho_1 = \frac{E(x_n \hat{x}_{n-1})}{E(\hat{x}_{n-1})^2}$$

In the next design iteration, predictor ρ_1 is used to generate predictions,

$$\tilde{x}'_2 = \rho_1 \hat{x}_1 \tag{5}$$

$$\tilde{x}'_3 = \rho_1 \hat{x}'_2 \tag{6}$$

$$\vdots$$

$$\tilde{x}'_N = \rho_1 \hat{x}'_{N-1} \tag{7}$$

From these equations, it can be clearly seen that $\tilde{x}'_2$ from this iteration is different from $\tilde{x}_2$ from the previous iteration. Since the reconstruction of a coefficient depends on its prediction, this implies that $\hat{x}'_2 \neq \hat{x}_2$, which by extension is true for all reconstructions (except the initial $\hat{x}_1$). There is a clear statistical mismatch since ρ_1 is applied to reconstructions in the current iteration, even though it was designed to be optimal for reconstructions in the previous iteration. This statistical mismatch leads to errors that propagate through the prediction loop, and build up over the sequence, sometimes catastrophically. In order to overcome this, we use Asymptotic Closed Loop (ACL) design. In ACL, reconstructions are updated in open-loop fashion by applying designed predictors to the reconstructions they were designed for to obtain the reconstructions for the next iteration. Thus, the TDTP filters are used with the same reconstructions they were optimized for. The ACL update of reconstructions is illustrated in Fig. 5. A particular DCT coefficient from frame n in design iteration i is denoted as $\hat{x}_{n,i}$ and ρ_i is the prediction coefficient designed using reconstructions from iteration i. As can be seen in Fig. 5, ρ_i is applied on the same set of reconstructions $\hat{x}_{n,i}$ for which it is optimal, to obtain new reconstructions for iteration $i+1$, which are then used to design ρ_{i+1}. The prediction coefficient estimation and reconstruction updates are given by.

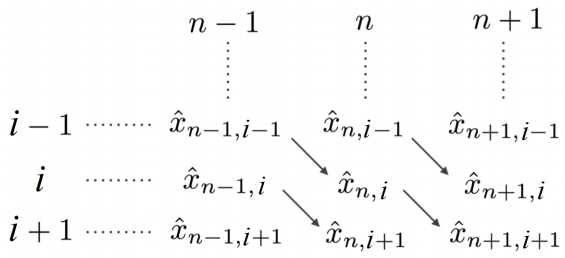

Fig. 5: ACL Design Paradigm

$$\rho_i = \frac{E(x_n \hat{x}_{n-1,i})}{E(\hat{x}_{n-1,i})^2} \tag{8}$$

$$\hat{x}_{n,i+1} = \rho_i \hat{x}_{n-1,i} \tag{9}$$

Across iterations, the predictions improve, which leads to improvement in subsequent reconstructions, thereby ensuring convergence. Upon convergence, i.e., $\hat{x}_{n,i} = \hat{x}_{n,i-1}$, ACL design is identical to closed-loop operation. Thus ACL provides a stable design platform and asymptotically optimizes the predictors for closed-loop operation.

V. Experimental results

The geodesic model was implemented with HM-16.15 [12] as the video codec. Geometry and sample rate conversion between source and coding formats were performed using the projection tool 360Lib-3.0 [13]. We chose the low delay P profile in HEVC with the only restriction that the prediction is obtained from the previous frame. The spherical videos with translational camera motion are projected to low-resolution ERP. The step sizes $\Delta\phi_s$ and $\Delta\theta_s$ are chosen to be $\frac{\pi}{H}$, where H is the height of the ERP video. For geodesic model, we use sinc interpolation at $\frac{1}{64}^{th}$ pixel accuracy to derive prediction signal from the reference frame. We test two approaches; the first one being the geodesic motion model with simple pixel copying, the second one being the proposed approach

Sequence	Geodesic model with pixel copying	Geodesic model with TDTP
Bicyclist	10.5	18.0
Chairlift	12.6	17.6
Balboa	21.5	24.8
Broadway	14.3	19.1
Harbor	50.7	54.3
Average	**21.9**	**26.8**

TABLE I: Bit-rate savings (%) for Y component over HEVC

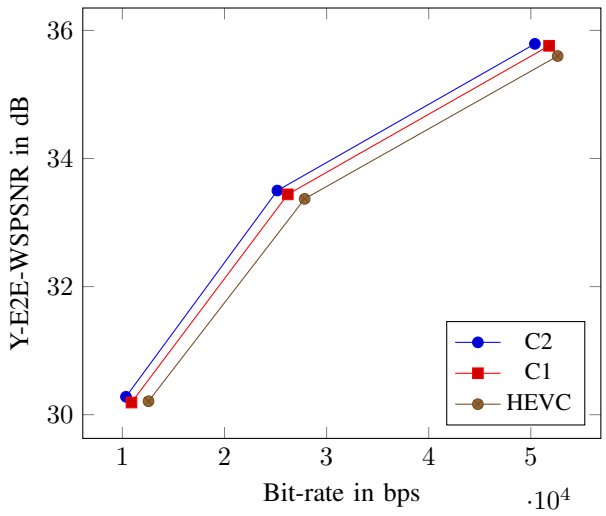

Fig. 6: RD curves for *Bicyclist* sequence with ERP as the projection format: C1 corresponds to our earlier approach of geodesic translation with pixel copying and C2 corresponds to our current geodesic translation with TDTP

of performing geodesic motion compensated prediction with TPTP. For both the approaches, we use HEVC as the anchor. The two methods were assessed by their respective distortion and bit-rate measures. R-D points were obtained by encoding at QP values of 22, 27, 32, and 37. We measured the distortion in terms of end-to-end weighted spherical PSNR [14], as recommended in [15]. Average bit-rate reduction is calculated using the Bjontegaard function, as per [16]. Table I shows the bit-rate savings (in %) of our earlier approach and current approach over standard temporal prediction in HEVC. Fig. 3 shows the rate-distortion curves for the *Bicyclist* sequence for QP values 22, 27 and 32 in ERP. This sequence had constant-rate PSNR gains of up to 0.7dB over HEVC.

As can be seen in Table I, our current approach has substantial gains over HEVC, the highest being bit rate savings of 54.3% for the Harbor sequence. Moreover, compared to our earlier approach, our current approach has gains as high as 7.5% for the Bicyclist sequence and an average of 4.9%. These results verify the validity and benefits of the proposed method.

VI. Conclusions

We proposed a method that reaps the benefits of precise motion compensation due to the geodesic motion model and disentanglement of spatial and temporal correlation via transform domain prediction. Critical problems of design instability were mitigated by an asymptotic closed-loop design approach. Experimental results demonstrate the effectiveness of the approach with significant performance improvement.

References

[1] J. P. Snyder, *Flattening the earth: two thousand years of map projections*, University of Chicago Press, 1997.

[2] G. J. Sullivan, J. Ohm, W. 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.

[3] M. Narroschke and R. Swoboda, "Extending HEVC by an affine motion model," in *Picture Coding Symposium (PCS)*, 2013, pp. 321–324.

[4] H. Huang, J. W. Woods, Y. Zhao, and H. Bai, "Control-point representation and differential coding affine-motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp. 1651–1660, 2013.

[5] 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.

[6] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "Rotational motion model for temporal prediction in 360 video coding," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 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)*, 2018, pp. 323–327.

[8] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "Motion compensated prediction for translational camera motion in spherical video coding," in *International Workshop on Multimedia Signal Processing (MMSP)*, 2018, pp. 1–4.

[9] J. Han, V. Melkote, and K. Rose, "Transform-domain temporal prediction in video coding: exploiting correlation variation across coefficients," *IEEE International Conference on Image Processing (ICIP), pp. 953–956*, 2010.

[10] H. Khalil, K. Rose, and L. Regunathan, "The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding," in *IEEE Transactions on Image Processing, vol. 10, no. 1*, 2001, p. 15–23.

[11] Y. He, B. Vishwanath, X. Xiu, and Y. Ye, "AHG8: Algorithm description of InterDigital's projection format conversion tool (PCT360)," *Document JVET-D0021*, 2016.

[12] "High efficiency video coding test model, HM-16.15," *https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/ tags/*, 2016.

[13] Y. He, B. Vishwanath, X. Xiu, and Y. Ye, "AHG8: InterDigital's projection format conversion tool," *Document JVET-D0021*, 2016.

[14] B Vishwanath, Y He, and Y Ye, "AHG8: Area weighted spherical PSNR for 360 video quality evaluation," *JVET-D0072, Chengdu, CN*, 2016.

[15] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "JVET common test conditions and evaluation procedures for 360-degree video," *JVET-F1030*, 2017.

[16] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, April*, 2001.