ROTATIONAL MOTION MODEL FOR TEMPORAL PREDICTION IN 360 VIDEO CODING

Bharath Vishwanath, Tejaswi Nanjundaswamy and Kenneth Rose

Department of Electrical and Computer Engineering, University of California Santa Barbara, CA 93106
{bharathv, tejaswi, rose}@ece.ucsb.edu

ABSTRACT

The recent boom in the field of virtual and augmented reality has dramatically increased the prevalence of spherical video. Given the enormous amount of data consumed by spherical video, it is critical to achieve efficient compression for storage and transmission. Prevalent approaches simply project (via different geometries) the spherical video onto planes for processing with traditional 2D video coding standards. However, such approaches are significantly sub-optimal as standard video coders only allow for block translations in the critical tool of motion compensated prediction, which is incompatible with the expected motion in projected spherical video. Specifically, the effective sampling density varies over the sphere and the resulting locally varying warping yields complex non-linear motion in the projected domain. Hence, translation in the projected domain does not preserve an object's shape and size on the sphere, and its corresponding motion vector does not have a useful physical interpretation. Instead, we propose to characterize the motion directly on the sphere with a *rotational* motion model, specifically, in terms of sphere rotations along geodesics. This model preserves object shape and size on the sphere. A motion vector in this model implicitly specifies an axis of rotation and the degree of rotation about that axis, to convey the actual motion of objects on the sphere. Complementary to the novel motion model, we further propose an effective motion search technique that is tailored to the sphere's geometry. Experimental results demonstrate that the proposed framework achieves significant gains over prevalent motion models, across various projection geometries.

Index Terms— inter prediction, 360 video, motion compensation, virtual reality, HEVC, video coding

1. INTRODUCTION

An immersive experience for users is enabled by capturing video with 360° view of the world on a sphere, allowing end users to dynamically control the viewing direction. To simplify storage, transmission and efficient access to desired portions of the 360° video, the data are projected onto planes via one of several possible geometries, e.g., equirectangular, cubemap, octahedron or icosaherdon [1]. In each case a uniform sampling of the plane induces a variable sampling density on the sphere which, in turn, introduces significant warping that varies in magnitude depending on location.

With its increased field of view, 360° video represents a considerably larger volume of data than that of standard 2D video, and hence the practicality of applications using such video critically depends on powerful compression algorithms that are tailored to this signal characteristics. A central component in modern video codecs such as H.264 [2] and HEVC [3] is motion compensated prediction, often referred to as "inter-prediction", which is tasked with exploiting temporal redundancies. Standard video codecs use a (piecewise) translational motion model for inter prediction, while some non-standard approaches considered extensions to affine motion models that may be able to handle more complex motion, at a potentially significant cost in side information (see recent approaches in [4, 5]). Still, in 360° video, the amount of warping induced by the projection varies for different regions of the sphere, and yields complex non-linear motion in the projected plane, for which both the translation motion model and its affine motion extension are ineffective. Note that even a simple translation of an object on the unit sphere leads to complex non-linear motion in the projected domain. Thus, a new motion compensated prediction technique that is tailored to the setting of 360° video signals is needed.

A closely related problem is that of motion compensated prediction in video captured with fish-eye cameras, where projection to a plane also leads to significant warping. A few interesting approaches have been proposed to address this problem in [6, 7], but these do not apply to motion under different projection geometries for 360° videos. Li et al., recently proposed an interesting 3D translational motion model for the cubemap projection [8]. In this approach, the centers of the current coding block and the reference block are mapped to the sphere and the 3D displacement between these vectors is calculated. The remaining pixels in the current coding block are also mapped to the sphere and then translated by the same displacement vector obtained for the block center. These translated vectors are not guaranteed to be on the sphere and thus need to be reprojected to it. Due to

This work was supported by a gift from InterDigital Communications, Inc.

978-1-5090-3649-3/17/$31.00

this final projection, object shape and size are not preserved, and some distortion is introduced. Moreover, motion search in this approach depends on the projection geometry, and thus the search range, pattern and precision vary across the sphere, depending on the sampling density. An early recognition of potential benefits of considering 360 video motion on the sphere is due to Tosic et al. [9], where they propose a multiresolution motion estimation method with a motion search model that is, nevertheless, equivalent to operation in the ERP domain.

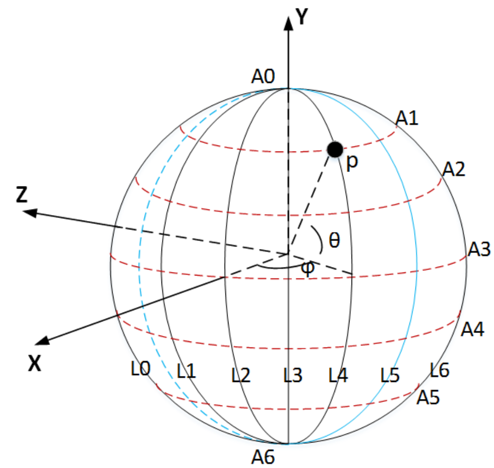
Since 360° video represents the scene captured on the unit sphere, it is most natural to characterize motion on that sphere. We thus propose a rotational model to characterize angular motion on the sphere. In the proposed framework, we define motion as rotation of a block of pixels on the surface of the sphere along geodesics and transmit information specifying this rotation as "motion vector" in lieu of the block displacement in the 2D projected geometry. Since rotations are unitary transformations, the proposed motion model preserves the shape and area of the objects on the sphere. This model also ensures that given a motion vector, regardless of a block's location on the sphere, it is rotated to the same extent. This feature addresses the motion search suboptimalities of current approaches, by allowing the search pattern, range and precision to be independent of the position of the block on the sphere. Complementary to the motion model, we propose employing a new pattern of "radial" search around the center of the coding block on the sphere for further performance improvement. Performing motion compensation on the sphere and having a fixed motion search pattern makes the proposed approach agnostic of the projection geometry and hence universally applicable to all projection geometries. Substantial gains in experiments validate the efficacy of the proposed approach.

2. OVERVIEW OF PROJECTIONS

While we propose to perform motion compensation on the sphere, the remainder of the video coding process is performed after projection onto the plane as usual. Below we briefly review two popular projection formats:

- Equirectangular Projection (ERP): This format is obtained by considering the latitude and longitude of a point on the sphere to be 2D Cartesian coordinates on a plane. The sampling pattern for ERP and the corresponding 2D projection are shown in Fig. 1. Clearly, objects near the pole get stretched dramatically in this format.

- Cubemap Projection (CMP): This format is obtained by radially projecting points on the sphere to the six faces of a cube enclosing the sphere (as illustrated in Fig. 2), and then unfolding the six faces. Warping is reduced in



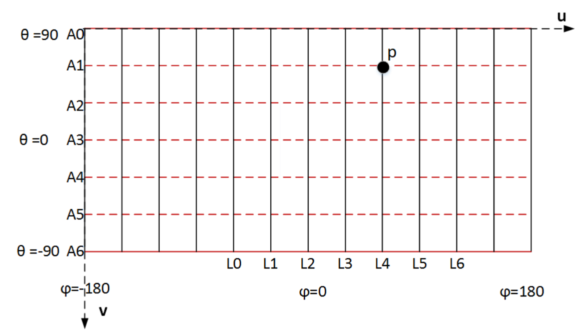


Fig. 1. Sphere sampling pattern for equirectangular projection (top) and corresponding 2D projection (bottom)

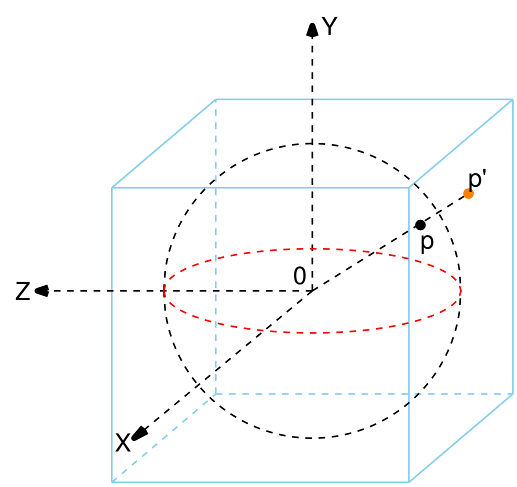


Fig. 2. Cubemap Projection

this format when compared to ERP, but it is still significant near the corners of the faces.

Please refer to the JVET document [10] for a more detailed discussion of these formats including procedures to map back and forth from a sphere to these formats.

3. PREDICTION FRAMEWORK WITH A ROTATIONAL MOTION MODEL

Since motion compensation in the projected domain lacks a precise physical meaning, we propose to perform motion compensation directly on the sphere. Let us consider a block of pixels in the current frame in the projected domain, which we seek to predict from the reference frame. An example of such a block in the ERP domain is illustrated in Fig. 3(a). We first map the block of pixels in the current frame to the sphere using the inverse projection mapping. The example block in Fig. 3(a) mapped back to the sphere is illustrated in Fig. 3(b). Let the center of this coding block in the projected domain correspond to vector $\mathbf{v}$ on the sphere. Our proposed motion search grid around the vector $\mathbf{v}$ is described next.

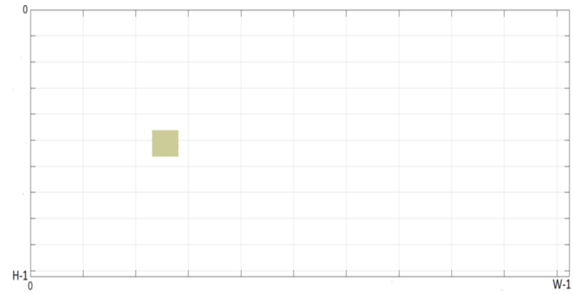
3.1. Proposed Motion Search

As previously mentioned, one of the main shortcomings of performing motion search in the projected domain is that the corresponding (on the sphere) search range, pattern and precision vary across the sphere. Since we propose to perform motion compensation directly on the sphere we overcome such arbitrary variations and employ the same search pattern for blocks everywhere on the sphere, agnostic of the projection geometry.

Let $\{(m, n)\}$ be the set of integer motion vectors and let R be the predefined search range, i.e., $-R \leq \{m, n\} \leq R$. To illustrate the search grid, let us pretend for a moment that $\mathbf{v}$ is the north pole. Then the motion vector (m, n) defines the rotation of $\mathbf{v}$ to a new point $\mathbf{v}'$ whose spherical coordinates (ϕ', θ') are given by:

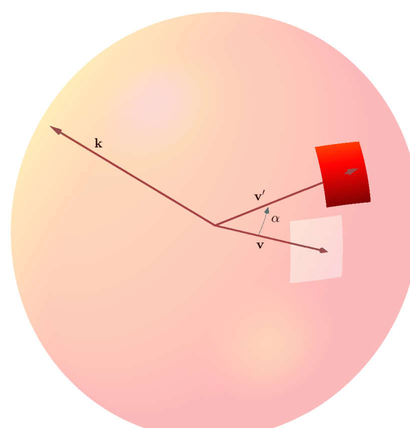
$$\phi' = m\Delta\phi, \theta' = \frac{\pi}{2} - n\Delta\theta \tag{1}$$

where, $\Delta\phi$ and $\Delta\theta$ are predefined step sizes. This search pattern is the intersections of latitudes and longitudes around the (pretend) north pole, effectively forming a radial grid. The pattern is tailored to the sphere's geometry with denser search grid near the center of the block and sparser search grid as we move away from the center. Fig. 4 illustrates the difference between the proposed search pattern and the search pattern for ERP in HEVC as seen on the sphere, wherein the search grid is arbitrarily denser closer to the poles.

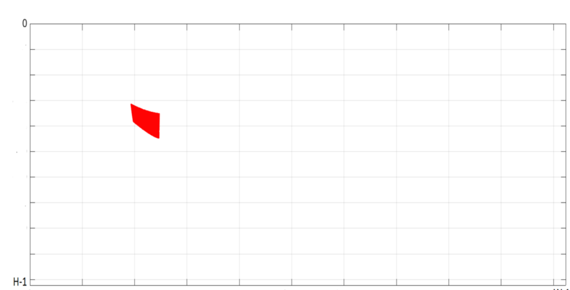


(a) A block in current ERP frame.

(b) Block mapped to sphere



(c) Geodesic rotation of block on sphere



(d) Rotated block mapped to reference ERP frame

Fig. 3. Illustration of various steps in the proposed motion compensation technique

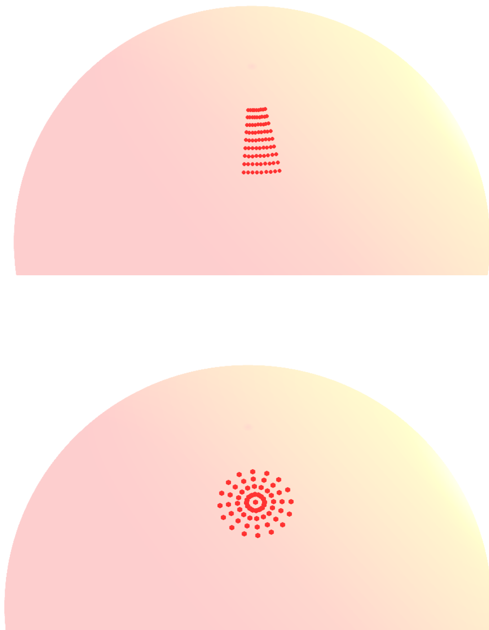


Fig. 4. HEVC search pattern (top) and proposed radial search pattern (bottom)

3.2. Proposed Rotation of the Block

Once we have the new vector $\mathbf{v}'$ corresponding to a motion vector (m, n), we rotate $\mathbf{v}$ to $\mathbf{v}'$ along the geodesic from $\mathbf{v}$ to $\mathbf{v}'$, via the Rodrigues' rotation formula [11]. This formula gives an efficient method for rotating a vector $\mathbf{v}$ in 3D space about an axis defined by unit vector $\mathbf{k}$, by an angle α. Let (x, y, z) and (u, v, w) be the coordinates of the vectors $\mathbf{v}$ and $\mathbf{k}$ respectively. The coordinates of the rotated vector $\mathbf{v}'$ will be:

$$\begin{aligned} x' &= u(\mathbf{k}\cdot\mathbf{v})(1-\cos\alpha) + x\cos\alpha + (-wy+vz)\sin\alpha, \\ y' &= v(\mathbf{k}\cdot\mathbf{v})(1-\cos\alpha) + y\cos\alpha + (wx-uz)\sin\alpha, \\ z' &= w(\mathbf{k}\cdot\mathbf{v})(1-\cos\alpha) + z\cos\alpha + (-vx+uy)\sin\alpha \end{aligned} \quad (2)$$

where $\mathbf{k}\cdot\mathbf{v}$ is the dot product of vectors $\mathbf{k}$ and $\mathbf{v}$. Since we want to rotate $\mathbf{v}$ to $\mathbf{v}'$ along the geodesic from $\mathbf{v}$ to $\mathbf{v}'$, we calculate the corresponding axis of rotation $\mathbf{k}$ and angle of rotation α, to employ Rodrigues' rotation formula. The axis of rotation $\mathbf{k}$ is the vector perpendicular to the plane defined by the origin, $\mathbf{v}$ and $\mathbf{v}'$ and is obtained by taking the cross product of vectors $\mathbf{v}$ and $\mathbf{v}'$, i.e,

$$\mathbf{k} = \frac{\mathbf{v}\times\mathbf{v}'}{|\mathbf{v}\times\mathbf{v}'|}. \quad (3)$$

The angle of rotation is given by,

$$\alpha = \cos^{-1}(\mathbf{v}\cdot\mathbf{v}'). \quad (4)$$

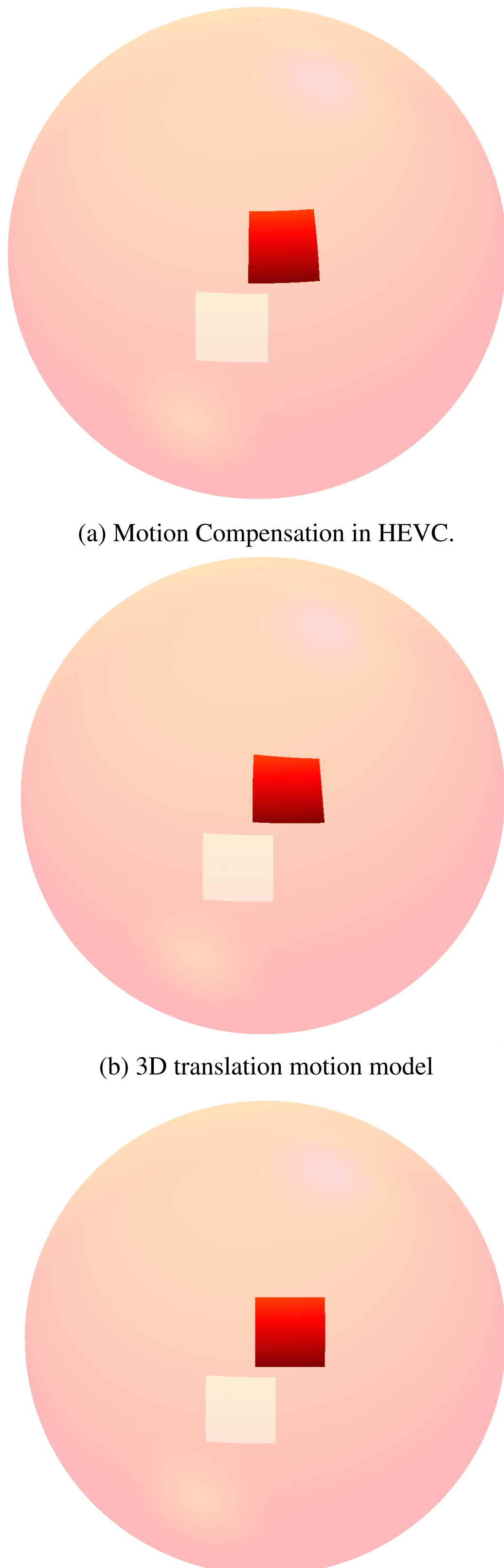
Given this axis and angle, we rotate all the points in the current block with same rotation operation. Rotation of block in Fig. 3(b) along the geodesic from $\mathbf{v}$ to $\mathbf{v}'$ is illustrated in Fig. 3(c). After rotation, we map the rotated block to the reference frame using the forward projection. An illustration of rotated block mapped back to ERP domain is shown in Fig. 3(d). Since the projected location might not be on the sampling grid of the reference frame, we perform interpolation in the reference frame to get the pixel value at the projected coordinate. The proposed motion compensation technique is summarized in Algorithm 1.

Algorithm 1 Proposed motion compensation technique

1: Map the block of pixels in the current coding unit on to the sphere.
2: Define a radial search pattern around the center of the block $\mathbf{v}$, to get the possible set of reference locations $\{\mathbf{v}'\}$.
3: Define a rotation operation which rotates $\mathbf{v}$ to $\mathbf{v}'$ along the geodesic from $\mathbf{v}$ to $\mathbf{v}'$.
4: Rotate all the pixels in the block with the rotation operation defined in Step 3.
5: Map the rotated coordinates on the sphere to the reference frame in projected geometry.
6: Perform interpolation in the reference frame to get the required prediction.

3.3. Comparison of Motion Models

Different motion compensation techniques lead to different shape changes of the object on the sphere. Fig. 5 illustrates the differences in the proposed approach, the motion model proposed in [8], and the motion compensation in HEVC. Marked in yellow is the block of pixels in ERP projected on to the sphere. The pixel locations in the reference frame derived based on different motion models are marked in red. Translation in ERP leads to a shrinkage of the block as we move away from the equator and is clearly seen in Fig. 5(a). As discussed earlier, 3D translation followed by projection on to sphere leads to change in shape and size of the block which is clearly seen in Fig. 5(b). The proposed approach preserves the shape and size of the block which is illustrated in Fig.5(c). While both our approach and the approach in [8] perform motion estimation on the sphere, our approach significantly differentiates in that the motion model is in terms of rotations on the sphere instead of translation in 3D space. Moreover, the search pattern in [8] depends on the projection geometry and varies across the sphere in contrast to the fixed search pattern employed in the proposed approach.



(a) Motion Compensation in HEVC.

(b) 3D translation motion model

(c) Proposed rotational motion model

Fig. 5. Motion model effect on block shape (same translation of block senter)

Sequence	3D translation (P frames)	Proposed approach (P frames)	3D translation (overall)	Proposed approach (overall)
bicyclist	-9.51	-12.9	-7.57	-10.35
chair	-5.89	-13.79	-4.1	-8.94
skate	-2.09	-9.9	-1.59	-7.68
glacier	-20.41	-34.83	-14.3	-25.97
train	-7.9	-7.95	-4.03	-4.03
average	-9.16	-15.87	-6.32	-11.39

Table 1. Bitrate savings for Y component over HEVC in ERP

4. EXPERIMENTAL RESULTS

To obtain experimental results, the proposed motion model is implemented in HM-16.14 [12]. The geometry mappings are done using the projection conversion tool given by [13]. We provide results for the low delay P profile in HEVC. To simplify the experiments, we only use the previous frame as the reference frame. Without loss of generality, subpixel motion compensation is disabled. We used our own implementation of the 3D translation motion model proposed in [8] with some improvements included for fairness. In [8], reference frame is interpolated to $\frac{1}{64}^{\text{th}}$ pixel precision and nearest neighbor is used in the interpolated reference frame to get the pixel values in the projection geometry. However, we use Lanczos 2 filter at the projected coordinate for interpolation in the reference frame for both our approach and our implementation of the approach in [8]. Also, we employ sphere padding [14] in the reference frame for improved prediction along the frame edges for all the competing methods. The step size $\Delta\phi$ is chosen to be $\frac{\pi}{2R}$ (where the search range R is same as what HEVC employs). $\Delta\theta$ in ERP is chosen to be $\frac{\pi}{H}$ as it corresponds to the change in the pitch (elevation) when we move by a single integer pixel in vertical direction. For CMP, since each face has field of view of $\frac{\pi}{2}$, we choose $\Delta\theta$ to be $\frac{\pi}{2W}$. We encoded 30 frames of five video sequences over four QP values of 22, 27, 32 and 37 in both ERP and CMP. All the sequences in ERP are at 2K resolution and the sequences in CMP have a face-width of 512. We measured the distortion in terms of Weighted-Spherical PSNR as proposed in [15]. Bitrate reduction is calculated as per [16]. Bitrate savings for Y component over HEVC in ERP are tabulated in Table 1 and the bitrate savings for Y component over HEVC in CMP are tabulated in Table 2. It is evident that the proposed approach provides significant overall bitrate reduction of about 11% and 6%, over HEVC and the approach of [8], respectively, in both ERP and CMP domain[1].

[1]Note that the differences in implementation and the distortion measure employed explain the slightly better results we obtain for [8] in comparison to what was reported in that paper.

Sequence	3D translation (P frames)	Proposed approach (P frames)	3D translation (overall)	Proposed approach (overall)
bicyclist	-0.96	-2.8	-0.84	-2.41
chair	-9.42	-17.36	-6.03	-11.66
skate	-9.25	-15.86	-6.98	-12.26
glacier	-10.72	-33.47	-7.53	-24.22
train	-8.4	-10.95	-4.50	-6.1
average	-7.75	-16.08	-5.18	-11.33

Table 2. Bitrate savings for Y component over HEVC in CMP

5. CONCLUSIONS

This paper proposes a novel rotational motion model for 360° video coding, which effectively captures the motion of the objects directly on the sphere. The paper also proposes a radial motion search pattern that is independent of the position of the block on the sphere. Unlike current approaches, the proposed framework retains the shape and size of the object after motion, while being agnostic of the projection geometry. The substantial gains compared to standard HEVC and other motion models, demonstrate the effectiveness of the proposed technique.

6. REFERENCES

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

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[3] 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.

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

[5] 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.

[6] A. Ahmmed, M. M. Hannuksela, and M. Gabbouj, "Fisheye video coding using elastic motion compensated reference frames," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2027–2031.

[7] G. Jin, A. Saxena, and M. Budagavi, "Motion estimation and compensation for fisheye warped video," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 2751–2755.

[8] 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.

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

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

[11] O. Rodrigues, "Des lois géométriques qui regissent les déplacements d'un systéme solide dans l'espace et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire," *Journal de Mathématiques Pures et Appliquées*, vol. 5, pp. 380–440, 1840.

[12] "High efficiency video coding test model, HM-16.14," *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] Y. He, Y. Ye, P. Hanhart, and X. Xiu, "AHG8: Geometry padding for 360 video coding," *Document JVET-D0075*, 2016.

[15] Y. Sun, A. Lu, and L. Yu, "AHG8: WS-PSNR for 360 video objective quality evaluation," *Document JVET-D0040*, 2016.

[16] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April*, 2001.