# Motion Compensated Prediction for Translational Camera Motion in Spherical Video Coding

Bharath Vishwanath, Tejaswi Nanjundaswamy and Kenneth Rose
*Department of Electrical and Computer Engineering*
*University of California, Santa Barbara*
Santa Barbara, CA, 93106
{bharathv,tejaswi,rose}@ece.ucsb.edu

*Abstract*—**Spherical video is the key driving factor for the growth of virtual reality and augmented reality applications, as it offers truly immersive experience by capturing the entire 3D surroundings. However, it represents an enormous amount of data for storage/transmission and success of all related applications is critically dependent on efficient compression. A frequently encountered type of content in this video format is due to translational motion of the camera (e.g., a camera mounted on a moving vehicle). Existing approaches simply project this video onto a plane and use block based translational motion model for capturing the motion of the objects between the frames. This ad-hoc simplified approach completely ignores the complex deformities of objects caused due to the combined effect of the moving camera and projection onto a plane, rendering it significantly suboptimal. In this paper, we provide an efficient solution tailored to this problem. Specifically, we propose to perform motion compensated prediction by translating pixels along their geodesics, which intersect at the poles corresponding to the camera velocity vector. This setup not only captures the surrounding objects' motion exactly along the geodesics of the sphere, but also accurately accounts for the deformations caused due to projection on the sphere. Experimental results demonstrate that the proposed framework achieves very significant gains over existing motion models.**

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

## I. Introduction

Spherical video offers an immersive experience for users by capturing the entire surroundings and allowing the users to view in any desired direction. This format is gaining significant popularity among outdoor enthusiasts, where the video of the surroundings is captured by cameras mounted on a moving vehicle. Given this popularity, content with translational motion of camera obviously requires compression tools that are tailored to this scenario to efficiently manage the enormous amount of spherical video data.

The prevalent approaches to compression of spherical video simply employ a standard (2D) video coder, and to do so they first project the spherical video onto one or more planes via one of several well known projection geometries, such as equirectangular projection (ERP), cubemap, etc. [1], each of which induces a different sampling density that varies with location on the sphere. In the standard video coders such as H.264 [2] and HEVC [3] the motion compensated prediction or inter-prediction which exploits temporal redundancies, is the major contributor to overall compression efficiency. This predictor matches a current block of pixels with a block in a previously reconstructed frame, assuming a simple block-based translational motion model. The difference between the reference block and the original block is then encoded and sent to the decoder. However, this motion model and its affine extensions [4], [5] fail to characterize motion in spherical video, due to the warping introduced by the projection to planes, and result in highly suboptimal performance.

A few more recent approaches have been proposed to address this difficulty, including attempts to derive motion vectors in 3D space before projection to the plane [6], modeling motion in terms of translation on a plane tangential to the sphere [7], and aligning the sphere to a stationary point, if identified, to perform ERP encoding [8]. The best results to date, by a significant margin, were obtained in our prior research, wherein the motion is modeled directly on the sphere via rotations that preserve the shape and size of objects [9]. Although these approaches try to characterize the motion on the sphere, they do not directly account for the nature of the perceived motion of objects and background, when the dominant element is in fact camera motion.

In this paper we propose a motion compensation procedure which perfectly accounts for the translational motion of the camera, thus capturing much of the effective motion field, with only some correction needed for independent motion of moving objects in the scene. At the heart of the approach is the realization that all background pixels on the sphere will move along their respective geodesics that intersect at the poles corresponding to the camera velocity vector. To illustrate this concept, let us pretend that the polar axis of the sphere coincides with the direction of the motion of the camera. In this setup all the surrounding objects' perceived motion will be parallel to the camera velocity vector, which upon projection to the sphere becomes motion along the "longitudes" corresponding to this imaginary pole. Thus, the displacement along the above "longitudes", i.e., geodesics intersecting at the camera motion poles, is the only motion vector required for motion compensated prediction to account for camera motion.

This proposed motion model exactly accounts for the perspective caused deformation resulting from the camera motion, and yields significantly more accurate prediction and thus considerable savings in the rate required to encode the prediction residual. Moreover, since a 1-D motion vector is (largely) sufficient to capture the motion that is mostly along the geodesics, unlike the 2-D motion vector required by all existing approaches, the proposed approach will also net considerable savings in side information bit rate. Overall, pixels in a prediction unit are mapped to the sphere, then moved along the geodesics defined by the camera motion and finally mapped back to the reference frame in the projected geometry to derive the prediction signal.

Note that the proposed approach is general and applicable to any projection geometries. Nevertheless, a particularly simple implementation is obtained when the projection format used is ERP. If ERP is performed after rotating the sphere so that its new polar axis aligns with the camera velocity vector, then the desired geodesics coincide with vertical lines on the plane, which significantly simplifies the motion-compensated prediction.

The rest of the paper is organized as follows. In section II we give an overview of equirectangular (ERP) and equi-angular cubemap (EAC) projections. The proposed approach is described in section III. Section IV has the experimental results followed by conclusions in section V.

## II. Overview of Projections

### A. *Equirectangular Projection*

The sampling pattern induced on the sphere by ERP, and the corresponding 2D projection are shown in Fig. 1. ERP induces on the sphere a vertical (along longitude) sampling density that is constant. However, the horizontal (along latitude) sampling density increases as we move towards the poles. Please refer to the JVET document [10] for a more detailed discussion on procedures to map back and forth from a sphere to ERP.

### B. *Equi-Angular Cubemap*

Equi-angular cubemap is shown in Fig. 2. In a traditional cubemap, the sphere is enclosed in a cube and each face of the cube is uniformly sampled. However, in EAC, the sampling is done such that it is uniform on the sphere rather than the projected cubemap faces. Please refer to the JVET document [11] for a more detailed discussion on procedures to map back and forth from a sphere to EAC.

## III. Proposed Approach to Account for Camera Motion

In order to illustrate the motion on the sphere resulting from the translational motion of the camera, let us consider a simple case in which a viewer is at the origin, enclosed by a sphere as shown in Fig. 3. The viewer sees a point P in the environment through its projection point S on the sphere. As the camera moves forward along its velocity vector $v$, the stationary point P is perceived as displaced to point P' relative to the viewer. Clearly, its corresponding projection on the sphere traverses along S-S', which is a part of a geodesic. Extending this scenario, we can see that, with a given constant

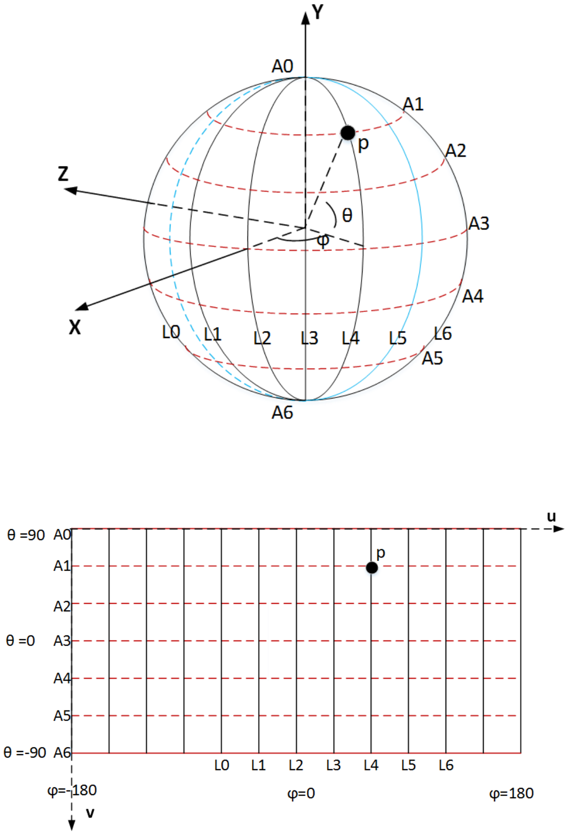

Fig. 1. Sphere sampling pattern for equirectangular projection (top) and corresponding 2D projection (bottom)

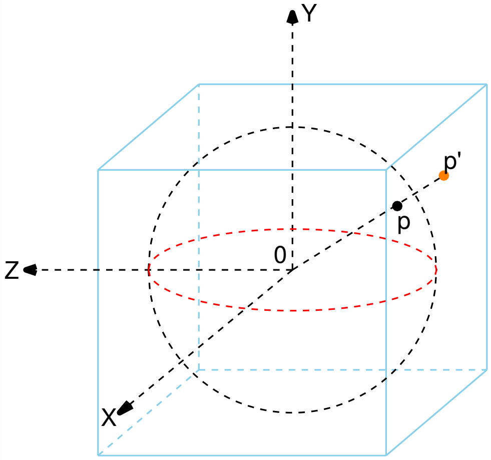

Fig. 2. Equi-Angular Cubemap

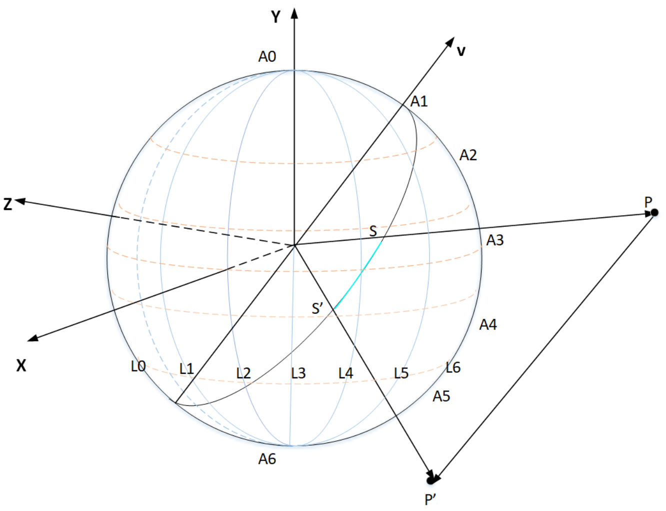

Fig. 3. Illustration of the trajectory of projection on the sphere due to camera motion

translational motion of the camera, surrounding static objects are perceived as moving on the sphere along geodesics that intersect at the poles of the camera motion vector. The camera motion also introduces perspective related deformations. As an object approaches a camera velocity pole, due to camera translation, there is shrinkage in its perceived shape (and vice versa, expansion with increasing distance from the camera velocity poles).

The current approach for encoding spherical video is shown in Fig. 4 and discussed in detail in [12]. The original spherical video is commonly represented by a high resolution projection to a plane via ERP. It is then re-projected onto a low resolution projection format such as cubemap. The projected video is then encoded using a standard video coder. At the decoder, the projected video is decoded, up-sampled and re-projected to ERP at the original resolution, as a representation of the original sphere. As explained in Section I, employing standard video coder results in highly suboptimal performance as it fails to characterize motion in spherical video, due to the warping introduced by the projection to planes.

Instead, we propose motion compensated prediction that fully accounts for camera motion. The proposed method assumes that the direction of camera motion is known, as most smart phones and 360 cameras include sensors such as accelerometer, gyroscope, etc., to detect and estimate motion, which can be fed to the video encoder. When such information is not available, it can be estimated directly from the video. Given the direction of the camera motion, we define geodesics which intersect at the point given by the unit vector in the direction of camera motion. With this setup, the specific steps are described below:

- Sphere Mapping: Let us consider a block of pixels in current frame that need to be predicted with motion compensation. We first project them onto the sphere. For pixel $(i, j)$ in the prediction block, let $(\theta_{ij}, \phi_{ij})$ be the spherical coordinates relative to the polar axis defined by the camera velocity vector.
- Geodesic Translation: Given a motion vector $(m, n)$, we move a pixel on the sphere along its geodesic to arrive at the spherical coordinates of the reference pixel as,

$$\theta'_{ij} = \theta_{ij} + m\Delta\theta, \phi'_{ij} = \phi_{ij} + n\Delta\phi \qquad (1)$$

  where $\Delta\theta$ and $\Delta\phi$ are predefined step sizes. Note that if the video motion field is entirely determined by translational motion of the camera, we only expect motion along the geodesics with no "lateral" motion, i.e., $\phi'_{ij} = \phi_{ij}$. However, we use 2D motion vectors to account for actual object motion unrelated to camera translation.
- Projection and Interpolation: The reference pixels are then projected to the reference frame. The projected coordinates may not be on the sampling grid of the reference frame, and we perform interpolation as required to obtain the value of the prediction signal at the projected coordinate.

Note that, in conjunction with the proposed motion-compensated prediction, the encoding operation itself can be performed either with respect to the original video sphere, or rotated to align its polar axis with the camera velocity vector.

## IV. Experimental Results

The proposed encoding procedure was tested with HM-16.15 [13] as the video codec. Geometry conversion and the sample rate conversion were performed using the projection conversion tool 360Lib-3.0 [14]. The proposed method was tested over five video sequences [15], [16] and [17], which are dominated by translational motion of the camera. The first one second of these videos were encoded at four QP values of 22, 27, 32 and 37 in random access profile. We provide results with ERP and EAC as the low resolution projection formats. ERP is encoded at 2K resolution. The face width for EAC is chosen to be 576 so that the total number of samples is approximately the same as ERP 2K. Rotational motion model proposed earlier by us is also implemented in HM-16.15. We measured the distortion in terms of end-to-end weighted spherical PSNR [18], as recommended in [12]. Bitrate reduction is calculated as per [19] over standard HEVC encoding technique for all the approaches. In [9], we showed that the rotational model outperforms other existing approaches. Table I compares the proposed method and our earlier rotational motion model [9] in terms of bitrate reduction for Y component with ERP as the low resolution projection format in all the approaches. Table II gives the bitrate reduction for the proposed method and rotational motion model [9] with EAC as the low resolution format in all the approaches. It is clear from the tables that the new motion model tailored to the translation motion of camera gives significant gains when compared to models that do not account for camera motion. The rate-distortion (RD) curve for the *bicyclist* sequence is shown in Fig. V. This clearly demonstrates the consistent significant performance gains at all bitrates.

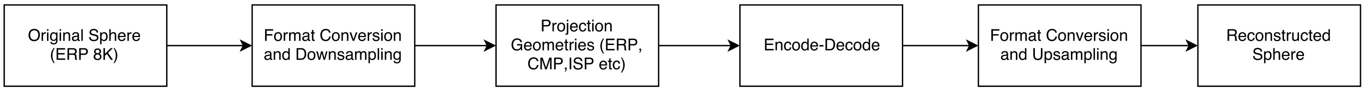
Original Sphere (ERP 8K)
Format Conversion and Downsampling
Projection Geometries (ERP, CMP,ISP etc)
Encode-Decode
Format Conversion and Upsampling
Reconstructed Sphere

Fig. 4. Standard spherical video encoding procedure

TABLE I
BIT RATE SAVINGS OVER HEVC WHERE ERP IS THE PROJECTION GEOMETRY (EVALUATED ON THE Y COMPONENT)

| Sequence | Rotational motion model in [9] | Proposed Method |
|---|---|---|
| Bicyclist | -12.7 | -17.3 |
| Chairlift | -7.5 | -13.4 |
| Broadway | -1.8 | -22.3 |
| Balboa | -3.2 | -29.1 |
| Harbor | -4.6 | -35.5 |
| Average | -5.9 | -23.5 |

TABLE II
BIT RATE SAVINGS OVER HEVC WHERE EAC IS THE PROJECTION GEOMETRY (EVALUATED ON THE Y COMPONENT)

| Sequence | Rotational motion model in [9] | Proposed Method |
|---|---|---|
| Bicyclist | -1.1 | -7.5 |
| Chairlift | -1.9 | -7.7 |
| Broadway | -0.5 | -5.8 |
| Balboa | -1.2 | -6.8 |
| Harbor | -0.9 | -1.8 |
| Average | -1.12 | -5.92 |

## V. CONCLUSIONS

This paper proposes a novel encoding technique for spherical videos dominated by translational motion of the camera. The proposed approach perfectly captures the perceived motion of objects on the sphere and the perspective distortion resulting from the translation motion of the camera. The motion model is agnostic of the projection format and extension of the approach to different geometries is straightforward. Experimental results yield substantial bit rate reduction and demonstrate the effectiveness of the proposed framework.

## 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.

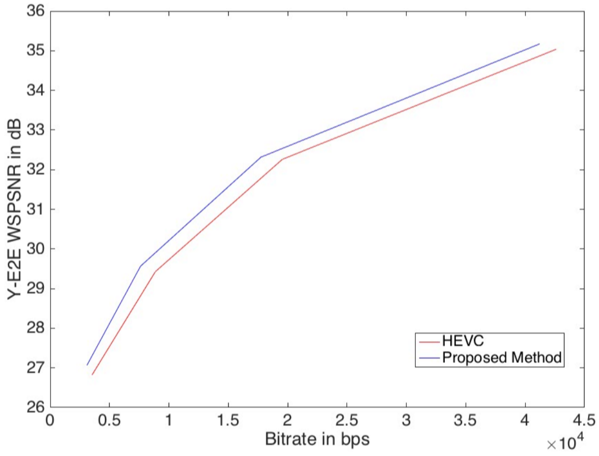

Fig. 5. RD curve for the *bicyclist* sequence where ERP is the projection geometry

[6] 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.

[7] F. De Simone, N. Birkbeck, B. Adsumilli, and P. Frossard, "Deformable block based motion estimation in omnidirectional image sequences," in *IEEE 19th International Workshop on Multimedia Signal Processing*, 2017, number EPFL-CONF-229997.

[8] J. Boyce and Q. Xu, "Spherical rotation orientation indication for hevc and jem coding of 360 degree video," in *Applications of Digital Image Processing XL*. International Society for Optics and Photonics, 2017, vol. 10396, p. 103960I.

[9] 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.

[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] M. Zhou, "AHG8: A study on equi-angular cubemap projection," *Document JVET-G0056*, 2017.

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

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

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

[15] A. Abbas, "Gopro test sequences for virtual reality video coding," *Document JVET-C0021*, 2016.

[16] A. Abbas and B. Adsumilli, "New gopro test sequences for virtual reality video coding," *Document JVET-D0026*, 2016.

[17] E. Asbun, Y. Ye, P Hanhart, Y. He, and Y. Ye, "Test sequences for virtual reality video coding from interdigital," *Document JVET-G0055*, 2017.

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

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