SPHERICAL VIDEO CODING WITH MOTION VECTOR MODULATION TO ACCOUNT FOR CAMERA MOTION

Bharath Vishwanath and Kenneth Rose

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

ABSTRACT

Emerging immersive multimedia applications critically depend on efficient compression of spherical (360-degree) videos. Current approaches project spherical video onto planes for coding with standard codecs, without accounting for the properties of spherical video, a severe sub-optimality that motivates this work. A common type of spherical video is dominated by camera translation. We recently proposed a powerful motion compensation technique for such videos which builds on the observation that, with camera translation, stationary points are perceived as moving along geodesics that meet at the point where the camera translation vector intersects the sphere. However, the approach follows standard coding procedures and translates all pixels in a block by the same amount on their respective geodesics, which is sub-optimal. This paper analyzes the appropriate rate of translation along geodesics and its dependence on the *elevation of a pixel on the sphere with respect to the camera velocity pole*. The analysis leads to a new approach that modulates the effective motion vectors within a block such that they perfectly capture the perceived individual motion of each pixel. Consistent gains in the experiments provide evidence for the efficacy of the proposed approach.

Index Terms— Spherical video, camera translation, motion compensation, HEVC, video coding

I. INTRODUCTION

Spherical video is rapidly gaining popularity in education, health care, surveillance, entertainment and related fields. In contrast to standard (planar) video, spherical video captures the entire surrounding, and creates an immersive experience. This format is gaining significant popularity, including among gamers and outdoor enthusiasts where spherical video is captured by a camera mounted on a moving subject or vehicle. In many other important applications such as robot navigation, camera translation is the dominant component of the motion in the video. The popularity of this class of videos and the enormous amount of data due to increased resolution of spherical videos pose major challenges and require efficient compression tools tailored to this scenario.

This work was supported in part by a Google Faculty Research Award

Standard compression approaches simply project the spherical data onto a plane (or planes) via one of several known geometries, according to projection formats such as equirectangular, cube-map, etc., [1]. The resulting planar video is encoded by existing planar video codecs. Modern video codecs, such as H.264 [2] and HEVC [3], perform motion compensated prediction in order to exploit temporal correlations in the signal. However, the simple translation motion model employed in the projected domain, as well as its affine extensions [4], [5], fail to capture the true motion of objects on the sphere. Moreover, the motion vector in the projected domain lacks a sound physical meaning. Some notable recent motion compensation procedures [6], [7], [8] do model the motion on the sphere. Specifically, they map a block of pixels onto the sphere, followed by either translation in the 3D euclidean space [6] or rotation on the sphere [7], [8]. However, they do not exploit models for perceived motion due to camera translation. We had earlier proposed a motion-compensation technique based on the observation that, with translational camera motion, stationary points move along respective geodesics that intersect at the poles of the camera velocity vector [9]. In our earlier approach, a block of pixels is mapped to the sphere and its pixels translated along their respective geodesics, then mapped back to the reference frame to derive the prediction signal. However, as in standard codecs, all pixels in a given block are translated by the same amount (albeit each along its geodesic), an assumption which we delve in this paper. The current work provides an analysis that sheds light on the direct dependency of the displacement rate of each pixel on its elevation on the sphere with respect to the camera velocity vector. Based on this analysis, we propose a motion vector modulation scheme, wherein the geodesic translation of the center of the block is signalled to the decoder, and from which an appropriate modulation scheme is employed to derive the individual motion vectors for pixels in the block. This perfectly captures the precise perceived motion per pixel on the sphere. The proposed method is agnostic of the projection geometries and can be easily extended to any new projection format. Its effectiveness is validated by experimental results showing consistent bit-rate reduction.

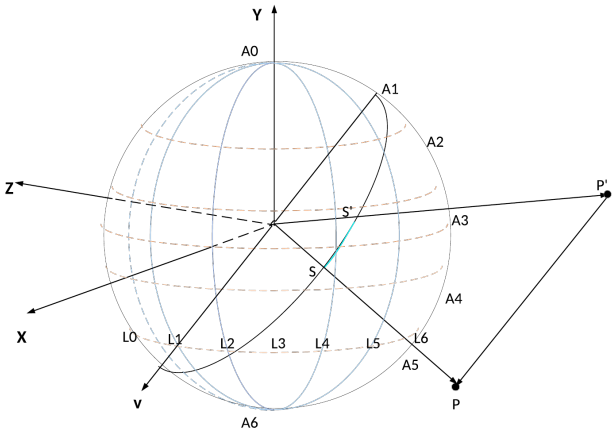

Fig. 1. Illustration of geodesic translation

II. BACKGROUND

This section briefly reviews the perceived motion of objects due to camera translation and our geodesic translation motion model proposed in [9].

II-A. Geodesic translation due to camera motion

In order to illustrate the perceived motion of objects on the sphere, due to camera translation, let us consider the simple scenario of Fig. 1, with the user at the origin observing an external point P. The user sees the point P through its projection point S on the sphere. As the camera translates along the direction of vector v, the point P is seen to be displaced to point P′. The corresponding projection on the sphere forms the arc S-S'. The central observation in our earlier work in [9] was that arc S-S′ is on a geodesic that intersects the poles defined by the camera velocity vector. It is thus easy to see that all the pixels translate along their respective geodesics, which all intersect at the same two points, namely, the camera motion poles. Based on this observation, we had earlier proposed a motion compensation technique in [9], which is briefly reviewed next.

II-B. Motion Compensation with geodesic translation

Consider the standard encoding pipeline depicted in Fig. 2. Spherical video is projected to plane(s) using a projection format of choice. Let us consider a target block of pixels, in the projection plane, which is to be inter-predicted. Motion compensation involves:

- Sphere mapping: The block of pixels is mapped back to the sphere. Let (θ_{ij}, ϕ_{ij}) be the spherical coordinates with respect to the camera translation vector.
- Geodesic Translation: Given a motion vector (m,n), the pixels on sphere are translated along geodesics as,

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

where $\Delta\theta_s$ and $\Delta\phi_s$ are predefined step sizes. If the dominant component of the motion is due to camera translation, the component n completely captures the motion with no "lateral" motion, i.e., $\phi'_{ij} = \phi_{ij}$. However, the use of 2D motion vectors allows for capturing actual object motion unrelated to camera translation.
- Mapping to reference frame: After geodesic translation, the pixels on sphere are mapped to reference frame in projection geometry to derive prediction signal.

III. PROPOSED MOTION VECTOR MODULATION

In our earlier method, all pixels in the block are translated by the *same* amount along their geodesics. A careful analysis reveals that fixed translation is sub-optimal. In this section, we derive the exact relationship between a pixel's rate of translation and its elevation on the sphere. The analysis yields a motion vector modulation scheme that accurately captures the perceived motion of each pixel.

III-A. How geodesic translation relates to elevation

In order to analyze the exact geodesic translation of each pixel, let us focus on the plane defined by P, P′ and the origin O, as shown in Fig. 3. Let ϕ be the elevation of point P with respect to the camera translation vector, and let $\Delta\phi$ be the change in elevation due to camera translation. Applying the law of sines to triangle POP′ we get,

$$\frac{|\text{OP}|}{\sin(\angle \text{OP}'\text{P})} = \frac{|\text{PP}'|}{\sin(\angle \text{P}'\text{OP})}, \qquad (2)$$

where $\angle \text{OP}'\text{P} = \frac{\pi}{2} - (\phi + \Delta\phi)$, OP is the depth of the point, denoted d, and PP′ equals the amount of camera translation, denoted t. We thus obtain the following relation,

$$\frac{d}{t} = \frac{\cos(\phi + \Delta\phi)}{\sin(\Delta\phi)} \qquad (3)$$

To motion-compensate a block of pixels, we make the simplifying assumption that all pixels in the block are approximately at the same depth from the origin. Thus, the ratio $\frac{d}{t}$ remains constant for all pixels in the block. This yields a relationship between the elevation of a pixel ϕ and the corresponding elevation change $\Delta\phi$. This result leads to an improved motion-compensation procedure.

III-B. Compensation by modulated motion vectors

A projected plane block to be inter-predicted is mapped to the sphere. For simplicity, we assume that the camera translation vector is known, as many omnidirectional camera rigs have sensors that measure camera motion (otherwise such global motion can be estimated from the video). Let (θ_{ij}, ϕ_{ij}) be a pixel's spherical coordinates with respect to the camera translation vector. Let (θ_c, ϕ_c) be the spherical coordinates of the center of the block after mapping to the sphere. Given a motion vector (m, n), the center of the block is translated along its geodesic as,

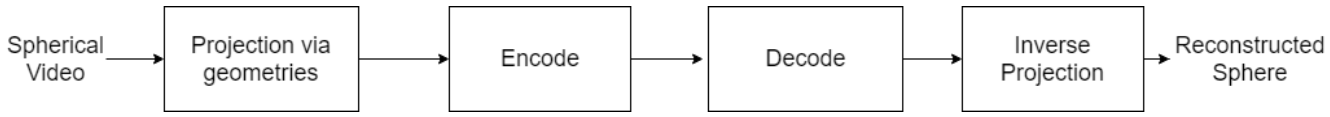

Fig. 2. Standard spherical video encoding procedure

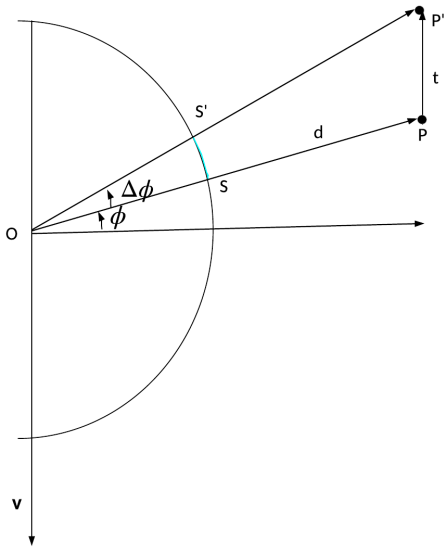

Fig. 3. Figure to analyze exact geodesic translation

$$\theta'_c = \theta_c + m\Delta\theta_s, \phi'_c = \phi_c + n\Delta\phi_s \tag{4}$$

where $\Delta\theta_s, \Delta\phi_s$ are predefined step-sizes. Let us specifically denote the change in elevation by $\Delta\phi_c$, i.e, $\Delta\phi_c = n\Delta\phi_s$. Now, for a pixel P_{ij} in the block, under the assumption of constant depth across pixels in a block, we obtain from (3):

$$\frac{\cos(\phi_{ij} + \Delta\phi_{ij})}{\sin(\Delta\phi_{ij})} = \frac{\cos(\phi_c + \Delta\phi_c)}{\sin(\Delta\phi_c)} = \frac{d}{t} = k \tag{5}$$

where $\Delta\phi_{ij}$ measures change in elevation of P_{ij} and k is a constant. Basic trigonometry yields the relationship,

$$\Delta\phi_{ij} = \tan^{-1}(\frac{\cos\phi_{ij}}{k + \sin\phi_{ij}}) \tag{6}$$

Thus, given the change in elevation of the center of the block, the elevation change for each individual pixel, or the amount of translation along its respective geodesic, is modulated according to (5). The pixels are thus translated to points with spherical coordinates given by,

$$\theta'_{ij} = \theta_{ij} + m\Delta\theta_s, \phi'_{ij} = \phi_{ij} + \Delta\phi_{ij} \tag{7}$$

The translated pixels are then mapped back to the reference frame to derive the prediction signal. The mapped pixel will not, in general, fall on the reference frame's sampling grid, and interpolation is performed to obtain the ultimate prediction signal. The proposed motion compensated prediction can thus be summarized as:

- A block of pixels is mapped to the sphere and the spherical coordinates (θ_{ij}, ϕ_{ij}) are derived with respect to the camera translation vector
- For a given motion vector (m, n), the block center on the sphere is translated according to (4).
- The change in elevation for each pixel in the block is calculated according to (6) and they are translated according to (7).
- The translated pixels are mapped to the reference frame and interpolated to derive the prediction signal.

IV. EXPERIMENTAL RESULTS

The encoding procedure including the proposed motion vector modulation was implemented and tested within HM-16.15 [10] as the underlying codec. Projection format conversion and the sample rate conversion were performed using the software tool provided in 360Lib-3.0 [11]. For testing, we chose five video sequences [12], [13] and [14], which are dominated by translational motion of the camera. Initial segment corresponding to the first one second of these videos were encoded at QP values of 22, 27, 32 and 37 in random access profile. We provide results with equi-angular cubemap (EAC) [15] as the low resolution projection format. The face width for EAC is chosen to be 576. Since each face corresponds to 90^o field of view, $\Delta\theta_s$ and $\Delta\phi_s$ are set to $\frac{\pi}{2*576}$. Interpolation in reference frame is performed at $\frac{1}{64}$ pixel precision. Our earleir motion compensation procedure with fixed geodesic translation [9] is also implemented in HM-16.15. We measured the distortion in terms of end-to-end weighted spherical PSNR [16], as recommended in [17]. Bitrate reduction is calculated as per [18] over the standard HEVC encoding technique, for both the approaches. For video sequences with substantial camera translation, our approach in [9] is shown to outperform other existing techniques. Thus, we only considered our earlier results from [9] for comparison. Table I compares the proposed method employing motion vector modulation, and our previous model of [9] with fixed geodesic translation, in terms of bitrate reduction for the Y component. The new motion compensation technique gives a significant 7% improvement, on the average, in terms of bit-rate reduction over HEVC. As compared to our previous approach in [9], we get gains as high as 4% for the *bicyclist* sequence. The proposed approach is more effective at low to medium bit-rates, where

Table I. Bit rate % savings over HEVC for Equi-angular Cubemap projection (evaluated on the Y component)

Sequence	Fixed geodesic translation in [9]	Proposed modulated geodesic translation
Bicyclist	7.5	11.4
Chairlift	7.7	8.8
Broadway	5.8	6.0
Balboa	6.8	7.7
Harbor	1.8	2.1
Average	5.9	7.2

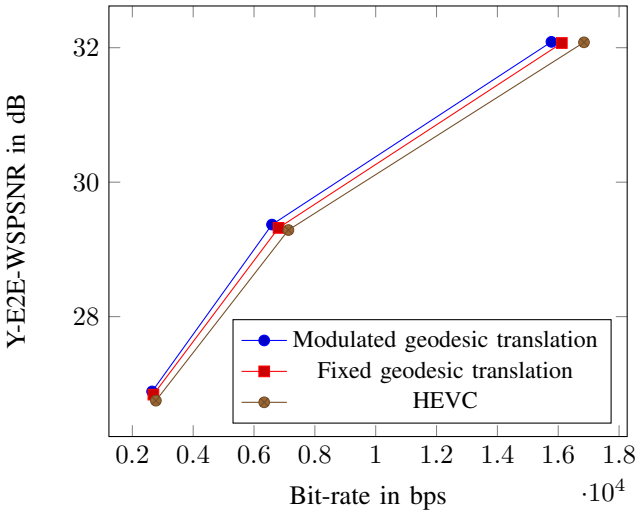

Fig. 4. RD curves for *bicyclist* sequence

HEVC chooses larger block sizes and hence the impact of motion vector modulation is higher. Fig. 4 depicts the rate-distortion (RD) curve for the *bicyclist* sequence in low to medium bit-rates.

V. CONCLUSIONS

This paper proposes a novel encoding technique with motion vector modulation for spherical videos dominated by translational motion of the camera. The proposed method perfectly captures the motion of each pixel on the sphere. The proposed motion compensation is agnostic of the projection format and its extension to any new format is straight forward. Significant gains in terms of bit-rate reduction clearly demonstrates the utility of the proposed approach.

VI. 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] 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] 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.

[8] 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.

[9] 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.

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

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

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

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

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

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

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

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

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