A Geodesic Translation Model for Spherical Video Compression

Bharath Vishwanath, *Member, IEEE*, Tejaswi Nanjundaswamy, *Member, IEEE*, and Kenneth Rose, *Life Fellow, IEEE*

***Abstract*— Spherical video coding is critical to the success of many virtual reality and related applications. This paper focuses on an important class of spherical videos whose dynamics involve camera motion. A common approach to spherical video coding is to project from the sphere onto a plane (or planes), where a standard video coder is applied. The projection induces warping resulting in complex non-linear motion in the projected domain that severely comprises the performance of motion models in standard coders. To overcome this shortcoming, we propose a new motion model that captures the motion field on the sphere, and capitalizes on insights into the perceived motion on the sphere due to camera translation. Specifically, surrounding static points are perceived as moving along their respective geodesics, which all intersect at the points where the camera velocity vector intersects the sphere. We analyze the rate of translation along geodesics and its dependence on the elevation of a pixel on the sphere with respect to the camera velocity vector. The analysis leads to a motion vector modulation scheme that perfectly captures the perceived motion of each pixel. Complementary to the new motion model, we propose a search grid tailored to capture expected geodesic motion on the sphere for effective motion estimation. The proposed method yields significant bit-rate savings over employing standard HEVC after projection, which validates its efficacy.**

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

I. Introduction

SPHERICAL video, or video captured on the unit sphere, can be viewed adaptively, in any desired direction, thus enabling an immersive experience. Due to the increased field of view, a spherical video requires significantly higher resolution which translates into an enormous volume of data. An important class of spherical video signals involves motion that is dominated by camera translation. Such signals are frequently encountered in numerous applications, including in robotics and navigation, sports and outdoor activities, etc. The prevalence of this class of video signals, and the enormous amount of data generated, necessitate the design of efficient compression tools that are tailored to this scenario.

Recent research and development efforts have produced highly efficient coding tools for standard 2-D videos. In order to leverage the performance of such coders for the compression of a spherical video, it is first projected onto a plane (or planes), via one of a number of available projection geometries, where it can be directly encoded by the standard codec. There is a variety of projection formats to choose from, including equirectangular projection (ERP), cubemap, octahedron projections, and many more [1], [2]. It is common practice to sample the signal uniformly in the projection plane, which induces a spatially varying sampling density on the sphere. Moreover, the variations in sampling density depend on the projection geometry used, and often cause significant warping of the projected spherical video. An unintended consequence is the complex non-linear nature of motion of objects in the projected video signal.

A critical component of modern video coders such as AVC [3], HEVC [4] and AV1 [5] is the motion compensated prediction module (or inter-prediction module), which exploits temporal redundancies and offers massive compression gains. Object motion is locally approximated by a simple translational model, which is used to derive a motion-compensated prediction signal from the reference frame. As complex motion may not be effectively captured by a simple translational model, an extension to affine motion models was proposed in [6], [7]. However, both the translational motion model and its affine extensions fail to accurately characterize the complex motion observed in projected spherical video, due to the warping introduced by the projection to planes, and hence the coding performance is considerably compromised. An additional suboptimality, when standard coders are applied to projected spherical videos, stems from the fixed motion search pattern employed throughout the frame. Due to the projection geometry, a fixed search pattern on the projected plane induces a spatially varying search pattern on the sphere. Thus, the motion search range and the motion vector precision vary from region to region on the sphere, depend on the projection geometry, and further compromise the motion compensation effectiveness.

A few approaches have been proposed recently to either model motion in 3-D space or to manage discontinuities

Manuscript received April 19, 2021; revised November 2, 2021 and January 26, 2022; accepted January 26, 2022. Date of publication February 23, 2022; date of current version March 3, 2022. This work was supported in part by the Google Faculty Research Award. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aline Roumy. *(Corresponding author: Bharath Vishwanath.)*

Bharath Vishwanath was with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA. He is now with Bytedance, San Diego, CA 92122 USA (e-mail: bharathv@ece.ucsb.edu).

Tejaswi Nanjundaswamy was with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA. He is now with Apple Inc., Cupertino, CA 95014 USA (e-mail: tejaswi@ece.ucsb.edu).

Kenneth Rose is with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: rose@ece.ucsb.edu).

Digital Object Identifier 10.1109/TIP.2022.3152059

1941-0042 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

between cube faces. Li *et al.* proposed a 3-D translational motion model [8], [9] in which a block of pixels is mapped to the sphere and then linearly translated in 3-D space. The 3D translation vector is derived by the centers of the current block and the reference block, and doesn't relate to the camera motion, thus rendering it sub-optimal. A rotational motion model was proposed in [10], [11], in which a block of pixels, mapped to the sphere, was rotated about an axis, thus preserving shape and size of objects on the sphere. Such an approach doesn't account for the perceived motion of the objects due to camera motion and the resulting perspective distortions. The approach in [12] models motion of objects on a plane tangential to the sphere. Such an approach cannot accurately capture the motion of the objects due to camera motion. To illustrate this, let us consider a tangential plane that comprises the point where the camera velocity vector intersects the sphere. The motion field in this plane will be radial, i.e, all the motion vectors converge towards or diverge away from the point where camera velocity vector pierces the sphere. Thus, a single 2D translation motion vector in the tangential plane cannot capture the motion of all the pixels in the block, rendering it sub-optimal. More recently, Marie *et al.* proposed various 2DoF and 3DoF motion models in [13]. Their 2DoF translate-linear model suffers from same sub-optimality as in [12]. The extended 3DoF could capture the motion due to camera translation, but suffers from high encoder complexity due to search in the 3D space and high bit-rate cost to convey 3D motion vectors to the decoder. Thus, all the existing approaches that try to characterize the motion in 3-D space or on the sphere, do not directly account for the nature of the perceived motion of objects, when it is dominated by underlying camera motion or have drawbacks of high encoder complexity and high bit-rate cost to convey motion vectors. Other relevant approaches include [14] and [15], who consider projection onto multiple cube faces, and try to minimize errors due to discontinuities across face boundaries. A motion vector scaling approach is proposed in [16] to reduce the cost of motion vectors. The algorithm in [17] relies on local statistics to find a rotation angle that would help standard 2-D motion compensated prediction in the ERP domain. We note that none of these approaches account for camera motion. 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 [18], [19]. None of these approaches are applicable to the current scenario of 360° videos with dominant camera motion.

This paper proposes a motion compensation procedure to capture *on the sphere* the accurate motion field that is due to camera translation. An important basic observation is that straight lines in 3-D space map to geodesics on the sphere. Thus, in the case of camera translation, all surrounding static objects exhibit relative motion along straight lines in 3-D space (parallel to the camera velocity vector), which is mapped to perceived motion along geodesics on the sphere. More specifically, the proposed method builds on the core realization that all static points in the environment are perceived to move on the sphere along their respective geodesics, namely, geodesics that intersect at the two points where an axis aligned with the camera velocity vector "pierces" the sphere. This characterization of the motion on the sphere also accounts for the perspective deformations that are due to camera motion. Specifically, it captures the magnification effects as objects approach the camera, and vice versa.

Having established the nature of perceived motion trajectories of surrounding objects on the sphere, the approach is further refined to characterize the rate of translation of pixels along their geodesics. A mathematical analysis sheds light on the rate of geodesic translation of pixels as related to the corresponding elevation of the pixels on the sphere with respect to the camera velocity axis. Based on this realization, we propose a motion vector modulation scheme, wherein, the geodesic translation prescribed for the center of a block of pixels, is modulated to extract refined individual motion vectors for the pixels in the block, which account for their respective degrees of elevation. The motion vector modulation scheme captures the variations in perceived motion across pixels in the block to yield significantly improved prediction and consequently additional coding gains. Moreover, since a 1-D motion vector is (largely) sufficient to capture the motion that is mostly along the geodesics, unlike the general 2-D or a 3-D motion vector required by all existing approaches, the proposed approach achieves significant savings in side information bit rate to convey motion vectors to the decoder. Nevertheless, to correct for the possibility of non-stationary objects whose motion is independent of the camera motion, we allow for a second motion component to capture lateral displacement (away from the geodesic). Overall, pixels in a prediction unit are mapped to the sphere, moved along the geodesics defined by the camera motion, where the rate of translation of each pixel along its geodesic is determined by the proposed modulation scheme, and finally mapped back to the reference frame in the projected geometry to derive the ultimate prediction signal.

A complementary focus of this paper is on the motion search module. The motion model efficacy in video coders largely depends on an effective motion search procedure. In the context of spherical videos, a fixed search grid in the projected plane induces a spatially varying search grid on the sphere, which is unnatural and undesirable. To overcome this shortcoming, this paper proposes to define the search grid on the sphere, making it agnostic of the projection geometry. Further, the search grid reflects the expected geodesic motion of the objects due to camera motion, and enables accounting for independent object motions.

Thus, in contrast with standard spherical video coders that perform their motion analysis and compensation in the "warped" projected domain, the approach proposed herein effectively conducts its analysis in the natural domain of the sphere. It is important to emphasize that the proposed motion estimation and compensation is hence independent of the projection format. Note that this paper subsumes our earlier work published in conference papers [20], [21] and the contributions therein namely, i) A geodesic translation motion model that captures the perceived motion of the objects

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

due to camera motion on the sphere; ii) A motion vector modulation scheme that accurately captures the motion of each pixel in the block on the sphere. This paper subsumes the above contributions and offers further contributions and enhancements: iii) A novel search grid on the sphere is proposed for motion estimation that adapts to the camera motion; iv) Methods to reduce computational complexity in implementing the motion model; v) Extensive experiments to prove the efficacy of the described method across various projection formats and vi) Subjective results to demonstrate the impact on visual quality of the reconstructed video.

The remainder of the paper is organized as follows. Section II provides an overview of common projections, namely, equirectangular projection (ERP), equi-angular cubemap (EAC) and equatorial cylindrical projection (ECP) and introduces the standard encoding pipeline for spherical video compression. The proposed approach is described in section III. Section IV deals with efficient implementation of the proposed motion model. Section V summarizes the experimental results followed by conclusions in section VI.

II. Background

A. Overview of Common Projections

Sphere to plane mappings have been studied extensively, and a plethora of such projections are covered, e.g., in [1], [2]. In this section, we briefly review a few important projection geometries. Perhaps the most popular projection, with extensive historical use is equi-rectangular projection (ERP), which is also commonly encountered in many virtual reality applications to this day. Beside ERP, we also review two additional projections, namely, equi-angular cubemap (EAC) and equatorial cylindrical projection (ECP), which are among the best known projection formats for video compression applications. Each projection format is presented concisely. For detailed mappings, please refer to the corresponding references in each projection format.

1) Equirectangular Projection: The sampling pattern induced on the sphere by ERP, and the corresponding projection to 2-D, are shown in Fig. 1. ERP maps longitudes to vertical straight lines and latitudes to horizontal straight lines. Thus, any point p on the sphere, with an elevation (pitch) ϕ and an azimuth (yaw) θ, is mapped to the position obtained on the 2-D grid as the intersection of the vertical and horizontal lines corresponding to the constant pitch and yaw on the sphere. ERP maintains constant vertical sampling density. However, horizontal sampling density increases as we move towards the poles. For more detailed mappings of ERP, please refer to [2].

2) Equi-Angular Cubemap Projection: 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, resulting in non-uniform sampling on the sphere. However, in EAC, the sampling is done such that it achieves close to uniform sampling on the sphere, rather than on the projected cubemap faces [22].

3) Equatorial Cylindrical Projection: Equatorial cylindical projection (ECP) was proposed in [23] and is shown in Fig. 3.

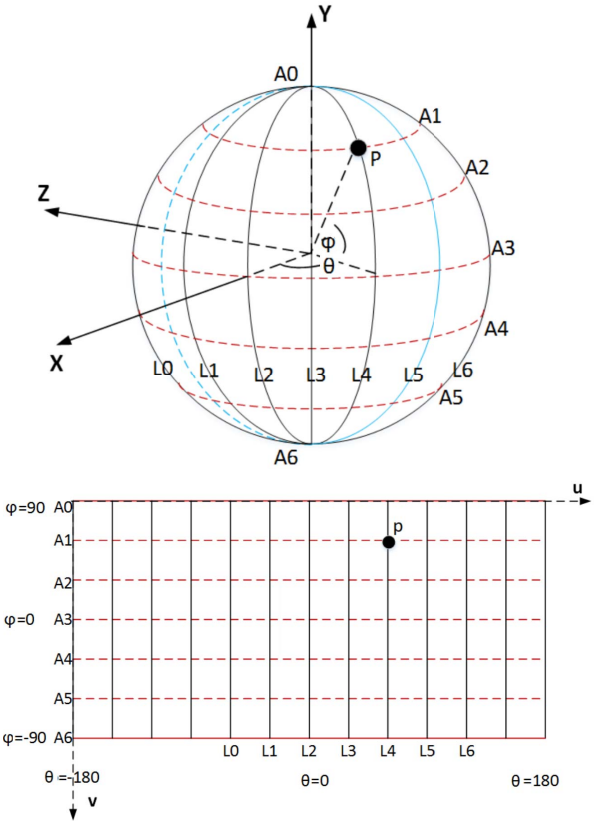


Fig. 1. Sphere sampling pattern due to equirectangular projection (top) and corresponding 2-D projection (bottom).

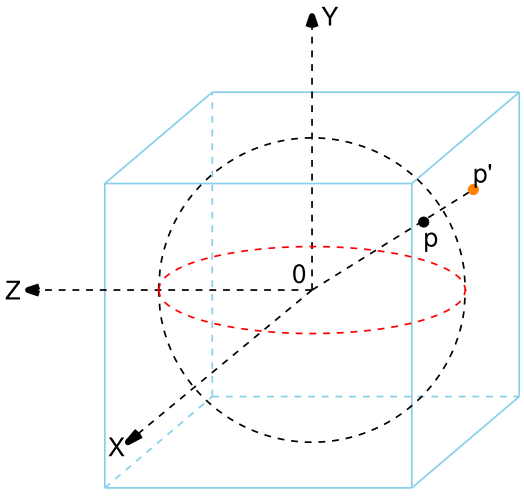


Fig. 2. Sphere mapping with equi-angular cubemap.

In ECP, the equatorial region corresponding to $\{-sin^{-1}\frac{2}{3} \leq \phi \leq sin^{-1}\frac{2}{3}\}$ is mapped to four faces of the cube via Lambert equi-area sampling [1].

The remaining two faces correspond to the polar caps which are first mapped to planar discs and then stretched to fit the square faces.

B. Standard Spherical Video Coding Pipeline

The standard spherical video coding pipeline is shown in Fig. 4 and discussed in detail in [24]. The original spherical video is projected onto a plane (or planes) via a projection

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

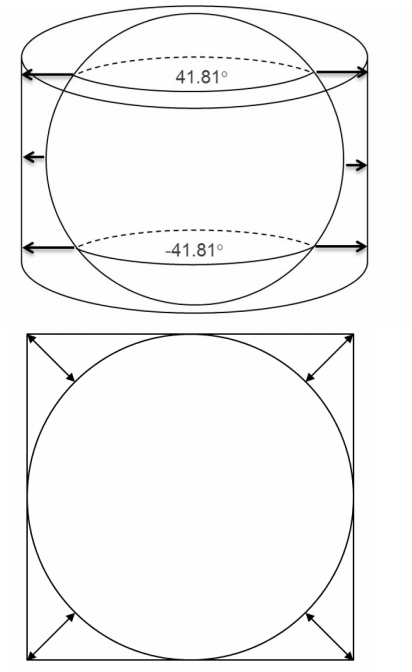


Fig. 3. Mapping of equatorial sphere region onto cylinder (top) and mapping of polar discs to square faces (bottom).

format such as ERP, cubemap, etc. The projected video is then encoded using a standard video coder. At the decoder, the projected video is decoded, and mapped back to the sphere to obtain the reconstructed spherical video. As previously explained, employing a standard video coder in this manner is highly suboptimal, as it fails to characterize accurately natural motion in spherical video, due to the warping introduced by the geometric projection.

III. Proposed Geodesic Motion-Compensated Prediction

This section presents the proposed geodesic translation motion model. We first illustrate, in a simple setting, the perceived motion of objects on the sphere, due to underlying camera motion. Based on these observations, we introduce a geodesic-based framework for motion compensated prediction. We then focus on the precise rate of translation of pixels along geodesics, and refine the standard motion vector definition for a block, by proposing a motion vector modulation framework to capture the exact motion of each pixel in the block. Finally, we introduce an approach to motion search grid adaptation that effectively circumvents shortcomings of standard motion estimation in the projected domain.

A. Motion Compensation by Geodesic Translation

1) Perceived Motion on the Sphere: In order to illustrate the perceived motion on the sphere, resulting from translational motion of the camera, consider a viewer at the origin, enclosed by a sphere, as shown in Fig. 5. The viewer sees point P, in the 3-D environment, through its projection point S on the sphere. As the camera moves forward according to 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 advances along the arc S-S′. It is important to note that the arc S-S′ is a segment of a geodesic that connects the two points where the camera velocity axis intersects the sphere. Building on this observation, we see that given constant translational motion of the camera, static surrounding points are perceived as moving on the sphere along their *respective* geodesics, which all intersect at the poles of the camera motion axis. It follows from these observations that the most natural way to capture the perceived motion of objects is by characterizing their geodesic translation on the sphere, in sharp contrast with the complex non-linear characterisation that would be necessary in the projected domain. Thus, based on the above observations, we propose a motion compensation procedure on the sphere, as discussed next.

2) Geodesic-Based Motion Compensation: As we observed, in cases of video dominated by camera motion, it is most natural to capture the perceived motion directly on the sphere. Specifically, motion compensation on the sphere will be performed as translation of a block along appropriate geodesics. 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. However, when such information is not available, it can be estimated directly from the video (see, e.g., [25], [26]). Given the camera velocity vector, we define geodesics that intersect at the two points where an axis aligned with this vector pierces the sphere. In other words, these two points are the "camera motion poles". With this setup in place, the specific three steps are specified and explained next:

a) Sphere mapping: Consider a block of pixels in the current frame, which needs to be predicted with motion compensation. Fig. 6(a) illustrates one such block in an ERP frame. We first project the block onto the sphere. For simplicity of presentation, let us define spherical coordinates with respect to the camera motion vector. Specifically, for pixel (i, j) in the prediction block, let (θ_{ij}, ϕ_{ij}) be the spherical coordinates relative to the polar axis defined by the camera velocity vector. A block of pixels mapped to the sphere and the spherical coordinate system with respect to camera translation vector is shown in Fig. 6(b). This step facilitates work on the sphere in a manner that is entirely agnostic of the projection format.

b) 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,

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

where $\Delta\theta_s$ and $\Delta\phi_s$ are predefined step sizes (more on the design choices in the experimental section). 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., $\theta'_{ij} = \theta_{ij}$. From the compression perspective, this results in notable bit-rate savings in terms of significant reduction in the side information allocated to motion vectors. Nevertheless, we allow for 2-D motion vectors to account for actual object motion, unrelated

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

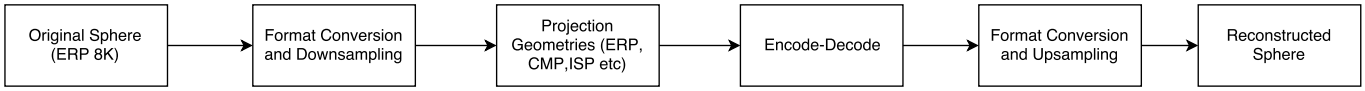


Fig. 4. Standard spherical video coding pipeline.

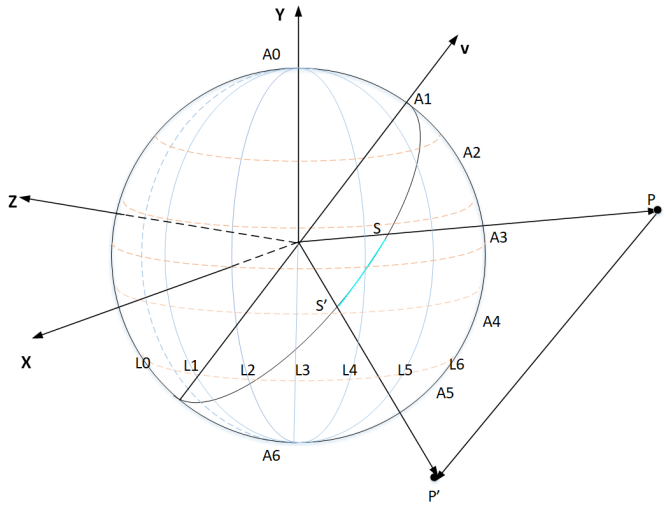


Fig. 5. A static point's perceived trajectory on the sphere due to camera motion.

to camera translation. Fig. 6(c) illustrates the geodesic translation of the block for a static object. It is evident from the figure that the proposed approach accounts for perspective distortions. Specifically, in the illustration, the object appears magnified as the camera approaches the object. Moreover, the motion vectors convey the amount of geodesic translation on the sphere, unlike motion vectors in the projected domain of standard techniques which afford little physical meaning or interpretation.

c) Projection and interpolation: The reference frame is still in the 2-D projection format. Thus, after geodesic translation of the block on the sphere, the translated pixels on sphere are projected to the reference frame. The projected coordinates may not be on the sampling grid of the reference frame. We thus perform interpolation in the projected domain to obtain the value of the prediction signal at the projected coordinate. Fig. 6(d) illustrates the reference block obtained by geodesic translation and mapping back to the projected domain.

B. Rate of Displacement: Motion Vector Modulation

The motion compensation so far exploits the nature of perceived motion on the sphere, due to camera motion, and translates all pixels in a block by the *same* distance on their respective geodesics. In this section, we further examine the rate of displacement of these pixels. Intuitively, it is easy to see that the rate of displacement of static surrounding points along their geodesics is inversely related to their depth, thereby reflecting the well known parallax effect, albeit in the context of spherical video. Moreover, even for objects at constant depth, the rate of translation depends on their position on the sphere. Mathematical analysis sheds light on the exact relationship of the rate of displacement with object depth and the elevation of the block on the sphere. This analysis leads to a motion vector modulation scheme that captures the exact motion of each pixel in a block.

1) Geodesic Displacement Analysis: In order to analyze the exact motion of each pixel along its geodesic, let us focus on the plane defined by P, P′ and the origin O, as shown in Fig. 7. Let ϕ be the elevation of point P with respect to the camera motion axis (i.e., relative to the corresponding "equator"), and let $\Delta\phi$ be the change in elevation due to camera translation. Applying the law of sines to triangle POP′ we get,

$$\frac{|\mathrm{OP}|}{\sin(\angle \mathrm{OP'P})} = \frac{|\mathrm{PP'}|}{\sin(\angle \mathrm{P'OP})} \tag{2}$$

It is easily seen that $\angle \mathrm{OP'P} = \frac{\pi}{2} - (\phi + \Delta\phi)$. OP is the depth of the point, denoted as d, and PP′ is the amount of camera translation denoted as t. We thus have the following relation,

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

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. In case the pixels do not have any constant depth, the encoder can always split the block via quad-tree partitioning and get blocks of approximately constant depth. 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 $\Delta\phi$. Armed with this observation, we extend the motion compensation procedure in III-A.2 to account for the actual rate of translation of pixels.

2) Motion Compensation With Modulated Motion Vectors: Similar to motion compensation summarized in sub-section III-A.2, a block of pixels that needs to be inter-predicted is mapped to the sphere. 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,

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

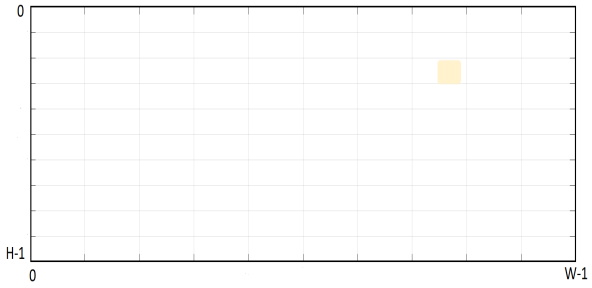
where $\Delta\theta_s, \Delta\phi_s$ are predefined step-sizes similar to (1). Let us specifically denote the change in elevation by $\Delta\phi_c$, i.e, $\Delta\phi_c = n\Delta\phi_s$. Now, for a pixel P_{ij} in the block, under the assumption of constant depth across pixels in a block, we obtain from (3):

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

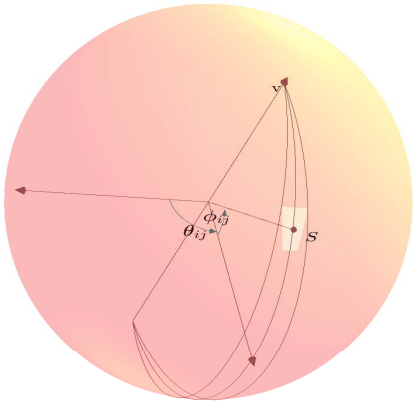
where $\Delta\phi_{ij}$ is the change in elevation of P_{ij} on the sphere and k is a constant. By simple trigonometry we obtain the relationship,

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

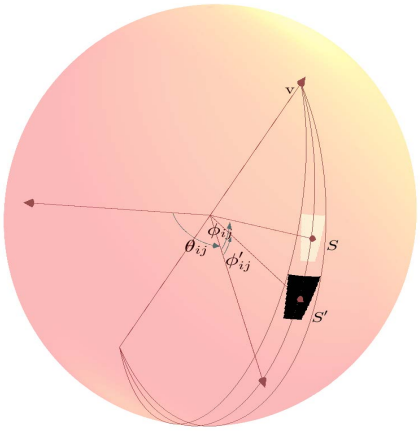
Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.



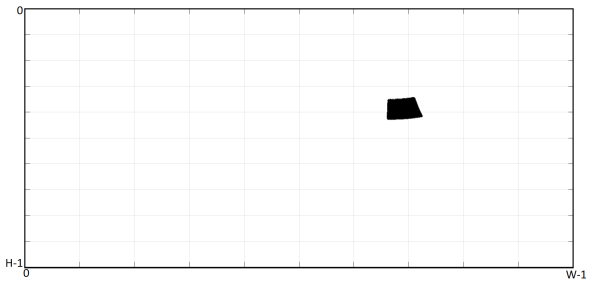
(a) Block of pixels in the projected domain



(b) Block mapped to the sphere



(c) Translation along geodesics



(d) Block mapped back to the projected domain

Fig. 6. The geodesic translation motion model.

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 (6). 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 to the reference frame to derive the prediction signal as discussed in III-A.2. The extended motion compensated prediction that accounts for the rate of displacement of individual pixels 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 to derive the prediction signal.

Note that motion vector modulation proves particularly effective in low bit-rate coding, since the encoder tends to use larger blocks, where motion vector modulation has significant impact in accurately capturing motion variations within the block.

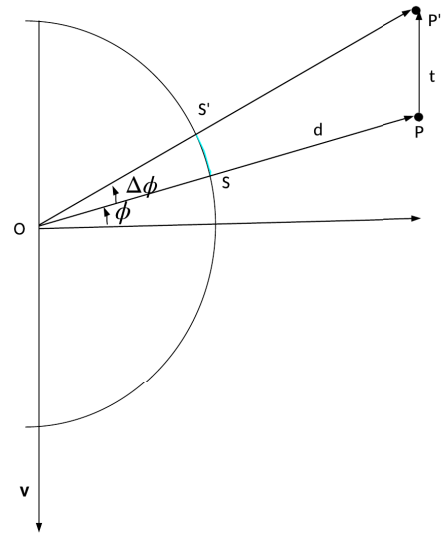


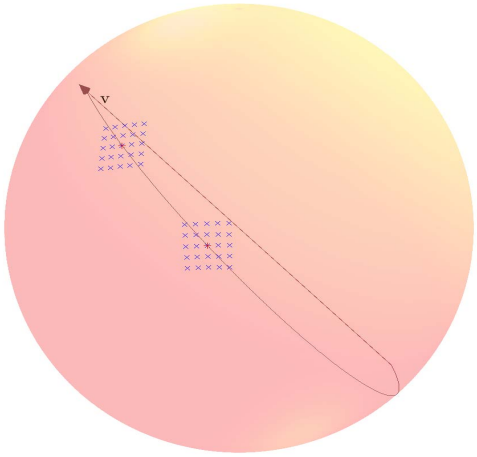
Fig. 7. Relation between geodesic displacement, elevation, depth and camera translation.

C. Motion Search Grid Adaptation

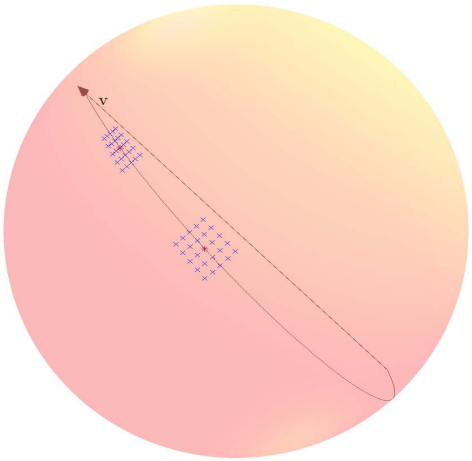
To gain the full benefit of the proposed motion model, we rely on efficient motion estimation procedures, the efficacy of which critically depends on the motion search grid. We first consider the shortcomings of the standard search grid, and then propose means to overcome these shortcomings, as well as to adapt the grid to account for camera motion.

1) Shortcomings of the Standard Search Grid: As observed in the discussion of projection formats, uniform sampling in the projection plane induces non-uniform sampling on the sphere. Thus, employing a fixed search pattern in the

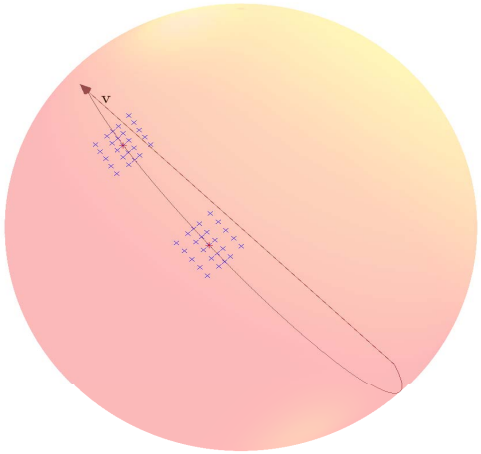
Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.



(a) HEVC search pattern defined in projection format



(b) Search pattern aligned with the expected geodesic motion of the center of the block



(c) Geodesic aligned search pattern optimized to account for object motion.

Fig. 8. Comparison of motion search patterns.

projection plane leads to spatially varying search patterns on the sphere. This observation is illustrated for the case where ERP is employed, in Fig. 8(a). Observe how the search pattern varies spatially on the sphere, in a way that depends on the arbitrary North-South pole. For the current scenario with camera motion, motion of the center of the block is expected to be along its respective geodesic. However, the search grid doesn't exploit this observation and need not have grid points along the expected geodesic. Moreover, for a mere 'approximation' of the motion of the center of the block, we need a 2-D motion vector in the projected domain. Thus, there is a clear motivation for the optimization of the search grid.

2) Proposed Search Grid Adaptations: To address the above mentioned shortcomings, we define a search grid on the sphere that directly captures the expected motion of the center of the block due to camera motion. Given a motion vector (m, n), the first component is used to capture change in yaw and the second component to capture change in elevation *with respect to the camera velocity vector*. Specifically, the spherical coordinates (θ_c, ϕ_c) are defined with the camera velocity vector as the polar axis. A motion vector (m, n) captures the change in the spherical coordinates of the center of the block as,

$$\Delta\theta_c = m\Delta\theta_s, \Delta\phi_c = n\Delta\phi_s \tag{8}$$

where $\Delta\theta_s$ and $\Delta\phi_s$ are predefined step sizes. This interpretation of motion vectors leads to the search pattern illustrated in Fig. 8(b). The proposed approach offers two benefits: It eliminates dependence on the arbitrary parameters of the projection (e.g., ERP's dependence on the North-South pole), and it explicitly captures the expected geodesic translation of the center of the block. Further, all the grid points on this geodesic corresponds to the case where $m = 0$, i.e, there is no change in yaw. Thus, for static objects, we have motion vectors that are 1-D, leading to bit-rate savings in side-information.

In case of object motion that is independent of camera motion, we use the component m to capture any "lateral" motion. Specifically, the component of the motion vector 'm' captures the change in azimuth as $m\Delta\theta$. It is important to note that, as we move away from the equator towards a camera motion pole, $m\Delta\theta_s$ corresponds to smaller lateral displacement. In terms of motion search pattern during motion estimation, this corresponds to a "shrinking" search grid, as illustrated in Fig. 8(b). This leads to suboptimality in estimating object motion that is independent of camera motion. It also results in excess penalty in side information needed to convey the lateral motion to the decoder, since small lateral displacement for blocks closer to the pole translate to large values of m. Thus, we need further search grid optimization to account for object motion with the following desired characteristics: i) The grid range should be agnostic of the elevation of the block with respect to the camera velocity vector. ii) For the scenario with dominant camera motion, we expect less 'lateral displacement', which motivates denser grid points near the center of the block, to capture the change in azimuth. However, we still must preserve the search range to handle occasional large object motions. To achieve the first desired characteristic, *only for the purpose of defining the search grid*, we proceed as if the block were at the equator, thereby eliminating the dependence of the search grid on the elevation of the block. To achieve dense grid close to the center of the block along the azimuth and yet not compromising on

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

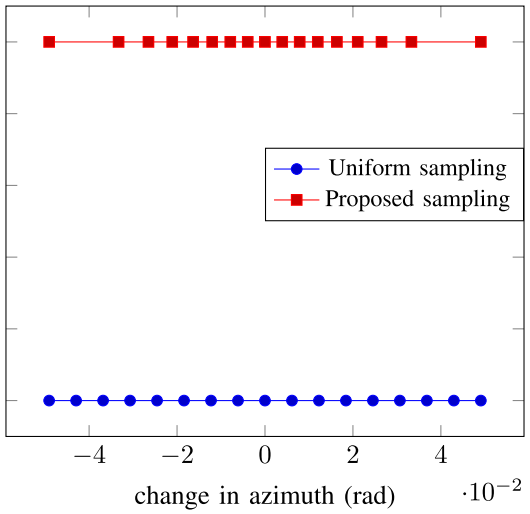


Fig. 9. Comparison of the proposed EAP based distribution of motion vectors along the azimuth and uniform distribution of motion vectors.

the search range, we define grid points in the spirit of equal area projection (EAP). In EAP, the longitudes are sampled such that, the sampling density decays as the cosine of the elevation, resulting in dense sampling close to the equator and spare sampling as we move towards the pole. We exploit this observation to have non-uniform density of motion vectors *along the azimuth* such that, we have dense sampling close the center of the block and spare sampling as we move away from center. Specifically, given a motion vector (m, n), the component m now represents the change is azimuth as,

$$\Delta\theta_c = K sin^{-1}(\frac{m}{R}), -R \leq m \leq R \tag{9}$$

where R is the search range. The choice of K determines the search range on the sphere, since $m = \pm R$ corresponds to $\Delta\theta_c = \pm K\frac{\pi}{2}$. For ERP, the width W corresponds to field of view of 2π, so K is chosen to get a search range of $\frac{2\pi R}{W}$ rad on the sphere. This yields $K = \frac{4R}{W}$. Similarly, for cube projections, the face-width W corresponds to field of view of $\frac{\pi}{2}$. Thus, $K = \frac{R}{W}$. The proposed distribution of grid points along the azimuth is illustrated in Fig. 9, in comparison with the uniform distribution. Note how the proposed grid is denser near the center and becomes sparser as we move away, while maintaining the same search range. The proposed sampling pattern in conjunction with its agnostic nature with respect to the elevation of the block on the sphere, is illustrated in Fig. 8(c).

Combining all the coding tools that we have discussed, the overall motion estimation algorithm at the encoder is summarized in Algorithm 1. The decoder essentially does this operation only for the best motion vector.

Algorithm 1 Overall Motion Estimation Algorithm

Map the block onto the sphere;
Let $(\theta_c.\phi_c)$ be the spherical coordinates of the center with respect to camera velocity vector;
Define search pattern as discussed in III-C ;
for *each point on sphere in search pattern* **do**
  (a) The center of the block is moved to (θ'_c, ϕ'_c) given by the search pattern;
  (b) Set $\Delta\theta_c = \theta'_c - \theta_c$ and $\Delta\phi_c = \phi'_c - \phi_c$;
  (c) Define $k = \frac{\cos(\phi_c+\Delta\phi_c)}{\sin(\Delta\phi_c)}$;
  for *each pixel in the block* **do**
    i) New azimuth is $(\theta_{i,j} + \Delta\theta_c)$;
    ii) New elevation is calculated as $\phi_{i,j} + \tan^{-1}(\frac{\cos\phi_{ij}}{k+\sin\phi_{ij}})$;
    iii) Map the new pixel on sphere to the projected domain;
    iv) Perform interpolation in the projected domain to get the prediction signal;
  end
  (d) Calculate the error between the original block and the predicted block;
end
Choose the best motion vector that minimizes prediction error.

IV. On Efficient Implementation of the Proposed Method

The proposed method consists of mappings between projection format and the sphere and and interpolations in reference frame that can be computationally very expensive. In this section, we present preliminary efforts to reduce the complexity. We consider each step in the proposed method and discuss the relevant optimizations:

1) Sphere Mapping: The first step in the proposed method involves mapping a block from a plane onto the sphere and computing the spherical coordinates with respect to the camera velocity vector. Given the projection format, the set of samples on the sphere are fixed. Thus, we create a look-up table of the spherical coordinates for all the pixels in the projected domain. This is a one-time computation whose results can then be reused for all frames, during motion estimation and compensation. The created look-up table greatly alleviates the burden of having to map from projection format to the sphere for a block in a given frame.

2) Geodesic Translation: For geodesic translation without modulated motion vectors, this step simply involves adding $(\Delta\theta, \Delta\phi)$ for all pixels in the block and doesn't call for much optimization. However, motion vector modulation, when enabled, requires trigonometric functions, for which appropriately devised look-up tables significantly reduce complexity.

3) Inverse Projection and Interpolation: After geodesic translation, the proposed method involves mapping pixels back to the reference frame and performing interpolation in the projected domain. Mapping between pixels on the sphere to the projection plane often involves complex trigonometric operations, which again are circumvented by look-up tables that minimize the computational burden. During motion estimation, it would be computationally expensive to perform higher order interpolation for every pixel in the block, for each choice of motion vector. In order to mitigate this computational cost, for the integer motion estimation stage, we up-sample the reference frames and use nearest neighbor interpolation, in the

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

TABLE I
BD-RATE GAINS IN % OVER HEVC (Y COMPONENT)

Geometry	Sequence	Rotational motion model in [10]	Proposed Method
ERP	Bicyclist	12.7	24.8
	Chairlift	7.5	14.5
	Broadway	1.8	22.6
	Balboa	3.2	29.7
	Harbor	4.6	35.9
	Average	**5.9**	**25.5**
EAC	Bicyclist	1.1	12.7
	Chairlift	1.9	9.4
	Broadway	0.5	7.0
	Balboa	1.2	8.3
	Harbor	0.9	3.1
	Average	**1.1**	**8.1**
ECP	Bicyclist	4.2	15.7
	Chairlift	2.5	7.2
	Broadway	0.6	7.1
	Balboa	1.6	8.9
	Harbor	0.9	2.7
	Average	**2.0**	**8.3**

up-sampled reference frame, to derive the prediction signal. An up-sampling factor of four was found to be a good trade-off between memory and performance. For successive motion vector refinements we use sinc interpolation in the reference frame at $\frac{1}{64}$ pixel precision.

We note that the central focus of the paper is to demonstrate the potential of the geodesic motion model and the above mentioned optimizations enumerate our initial efforts to reduce complexity. Some preliminary results on the complexity reduction from the above enumerated methods are presented in the following experimental results section.

V. EXPERIMENTAL RESULTS

A. Simulation Settings

The proposed encoding procedure was implemented with HM-16.15 [27] as the video codec. Geometry conversion and the sample rate conversion were performed using the projection conversion tool 360Lib-3.0 [28]. The proposed method was tested over five video sequences [29], [30] and [31], which are dominated by translational motion of the camera. The first one second of each video was encoded at four QP values of 22, 27, 32 and 37, in random access profile. We provide results with ERP, EAC and ECP as the low resolution projection formats. ERP is encoded at 2K resolution. The face width for EAC and ECP is chosen to be 576 so that the total number of samples is approximately the same as ERP, namely, 2K. The step sizes $\Delta\theta_s$ and $\Delta\phi_s$ are chosen to be $\frac{\pi}{H}$, where H is the height of the ERP video. The corresponding step sizes for EAC and ECP with face-width W are chosen to be $\frac{\pi}{2W}$ since a face-width of W corresponds to a field of view of $\frac{\pi}{2}$rad. The rotational motion model proposed earlier by us [10], is also implemented in HM-16.15. In both approaches, we use sinc interpolation at $\frac{1}{64}^{th}$ pixel accuracy to derive prediction signal from the reference frame.

B. Objective Results

For objective results, bit-rate reduction is calculated as per [32] over standard HEVC encoding technique for all the approaches. We measured the distortion in terms of end-to-end weighted spherical PSNR [33], as recommended in [24] and [34]. In [10], we had already shown that the rotational model outperforms other existing approaches, and this is the reason it was selected here as leading (nearest) competitor. Table I compares the proposed method and rotational motion model [10] in terms of bit-rate reduction over HEVC, for the Y component, and provide results in conjunction with projections ERP, EAC and ECP, respectively. It is clear from the table that the new motion model tailored to the translation motion of camera gives significant gains when compared to models that do not properly account for camera motion. The rate-distortion (RD) curves for the *bicyclist* and *balboa* sequences for different projection formats are shown in Fig. 10-12. Overall, the results demonstrate consistent performance gains at all bit-rates and across different geometries.

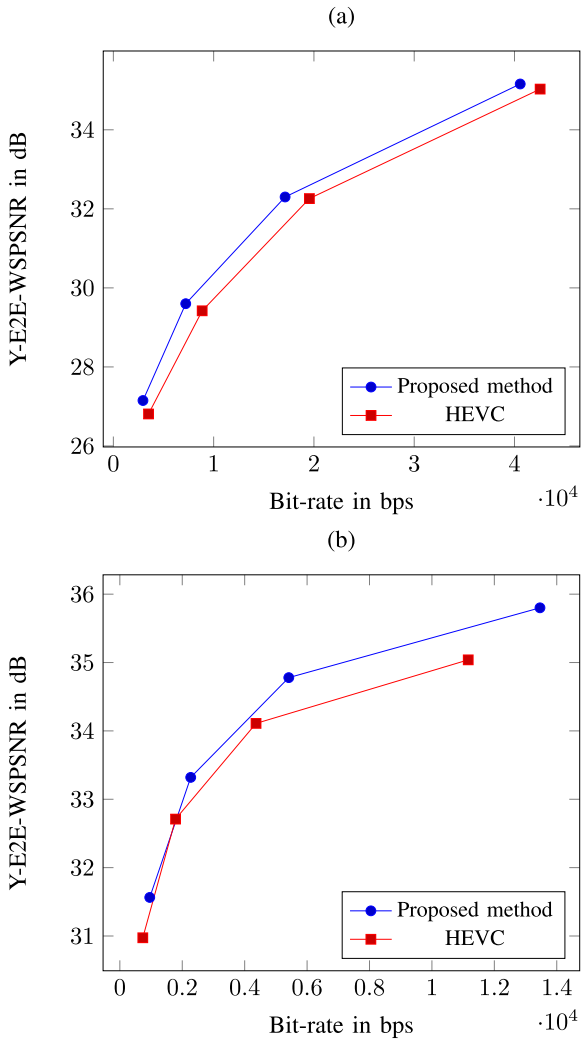


Fig. 10. RD curves for (a) *bicyclist* and (b) *balboa* sequences with ERP as the projection format.

As regards the complexity, the unoptimized encoder and decoder have complexity of over 40x and 8x, respectively, compared to the HEVC anchor. The complexity optimizations proposed in section IV bring down the complexity to 10x-15x for the encoder, and 3x for the decoder.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

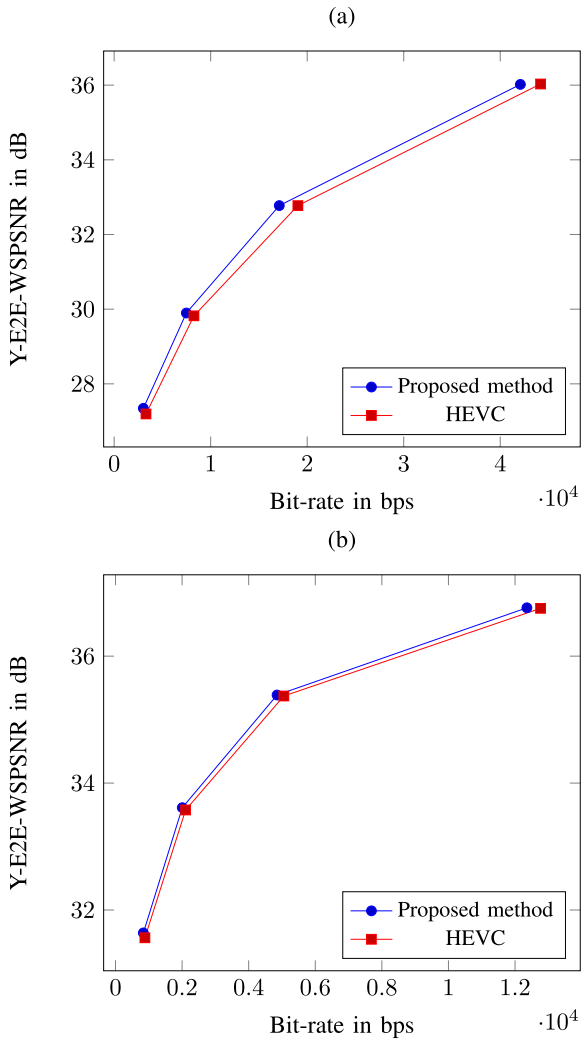


Fig. 11. RD curves for (a) *bicyclist* and (b) *balboa* sequences with EAC as the projection format.

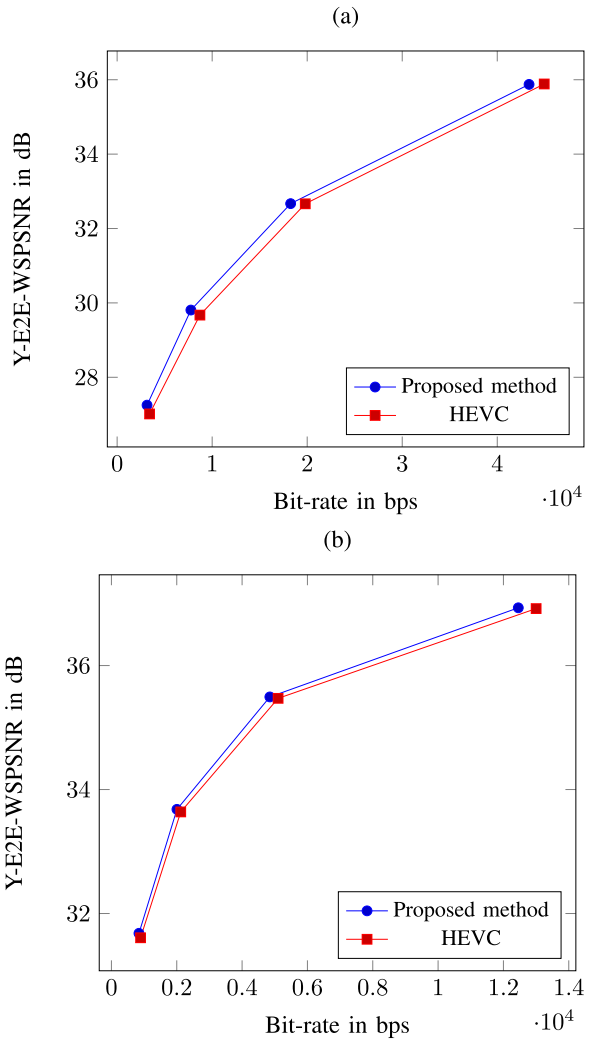


Fig. 12. RD curves for (a) *bicyclist* and (b) *balboa* sequences with ECP as the projection format.

TABLE II

BIT RATE % SAVINGS OVER HEVC FOR EQUI-ANGULAR CUBEMAP PROJECTION WITH DIFFERENT PRECISION OF INTERPOLATION FILTER (EVALUATED ON THE Y COMPONENT)

Sequence	Interpolation at $\frac{1}{4}$ pixel precision	Interpolation at $\frac{1}{64}$ pixel precision
Bicyclist	11.6	12.7
Chairlift	8.5	9.4
Broadway	5.6	7.0
Balboa	7.6	8.3
Harbor	1.8	3.1
Average	**7.0**	**8.1**

TABLE III

BIT RATE % SAVINGS OVER HEVC FOR *bicyclist* SEQUENCE FOR EQUI-ANGULAR CUBEMAP PROJECTION WITH ERRORS IN CAMERA VELOCITY VECTOR (EVALUATED ON THE Y COMPONENT)

Azimuth estimation error θ_e in degree	Elevation estimation error ϕ_e in degree	Bit-rate reduction over HEVC
0	0	12.7
1	1	11.7
2.5	-2.5	11.1
7.5	2.5	9.5
-7.5	7.5	6.7

As mentioned earlier, we use sinc interpolation filter at $\frac{1}{64}^{th}$ pixel accuracy to derive prediction signal from the reference frame. In contrast, HEVC only supports interpolation at $\frac{1}{4}^{th}$ pixel accuracy. The gains obtained by employing interpolation at $\frac{1}{64}^{th}$ pixel accuracy and interpolation at $\frac{1}{4}^{th}$ pixel accuracy for EAC projection format is presented in Table II. We observe an average 1.1% bit-rate savings by employing interpolation at $\frac{1}{64}$ pixel precision as compared to the interpolation at $\frac{1}{4}$ pixel precision in HEVC.

C. Subjective Results

To get subjective results, we compressed videos with specified target bit-rate with HEVC anchor and the proposed method. Fig. 13 shows significant improvement in the visual quality for example frames from the "balboa" sequence with the proposed motion model as compared to HEVC based encoding at the same bit rate. For example, we draw attention to edges on the building, where the proposed method offers crisp reconstruction in contrast with the highly distorted reconstruction of HEVC.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

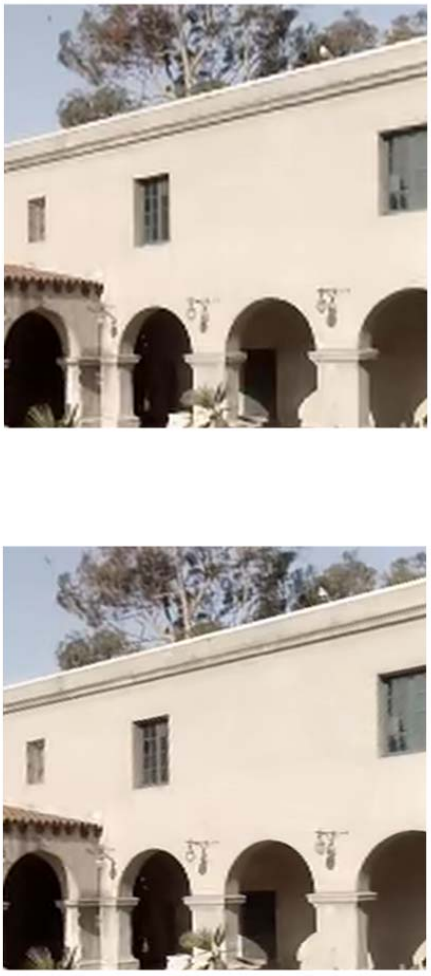


Fig. 13. Subjective comparison for balboa sequence encoded at constant bit-rate with ERP as projection format. anchor (top) and proposed method (bottom).

D. Effect of Error in Camera Pose Estimation

The proposed method relies on sensor measurements or vision algorithms to obtain the direction of camera motion. These measurements or the estimations are prone to errors that impact the performance of the proposed algorithm. To analyze this, we consider the *bicyclist* sequence with EAC projection format and encode it with erroneous directions of camera motion. The error in the azimuth and the elevation of the camera direction and the corresponding average bit-rate savings over HEVC are presented in Table III. We observe that the proposed method significantly outperforms HEVC even with errors in the camera velocity vector.

VI. Conclusion

This paper proposes a novel encoding technique for spherical videos with dynamics dominated by camera motion. The proposed approach leverages insights into the perceived motion of static objects on the sphere, and the perspective distortion due to camera motion. The motion model is agnostic of the projection format and the approach is extendable to other geometries in a straightforward manner. Experimental results yield substantial bit rate reduction and demonstrate the effectiveness of the proposed framework.

References

[1] J. P. Snyder, *Flattening Earth: Two Thousand Years Map Projections*. Chicago, IL, USA: Univ. Chicago Press, 1997.

[2] *Joint Video Exploration Team of ITU-T SG*, document JVET-F1003, Apr. 2017, vol. 16.

[3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Aug. 2003.

[4] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Sep. 2012.

[5] Y. Chen *et al.*, "An overview of core coding tools in the AV1 video codec," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 41–45.

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

[7] H. Huang, J. W. Woods, Y. Zhao, and H. Bai, "Control-point representation and differential coding affine-motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1651–1660, Oct. 2013.

[8] L. Li, Z. Li, M. Budagavi, and H. Li, "Projection based advanced motion model for cubic mapping for 360-degree video," 2017, *arXiv:1702.06277*.

[9] L. Li, Z. Li, X. Ma, H. Yang, and H. Li, "Advanced spherical motion model and local padding for 360° video compression," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2342–2356, Dec. 2019.

[10] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "Rotational motion model for temporal prediction in 360 video coding," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

[11] B. Vishwanath, K. Rose, Y. He, and Y. Ye, "Rotational motion compensated prediction in HEVC based omnidirectional video coding," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 323–327.

[12] F. De Simone, P. Frossard, N. Birkbeck, and B. Adsumilli, "Deformable block-based motion estimation in omnidirectional image sequences," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

[13] A. Marie, N. M. Bidgoli, T. Maugey, and A. Roumy, "Rate-distortion optimized motion estimation for on-the-sphere compression of 360 videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1570–1574.

[14] J. Sauer, J. Schneider, and M. Wien, "Improved motion compensation for 360° video projected to polytopes," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 61–66.

[15] Y. He, Y. Ye, P. Hanhart, and X. Xiu, "Motion compensated prediction with geometry padding for 360 video coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

[16] R. Ghaznavi-Youvalari and A. Aminlou, "Geometry-based motion vector scaling for omnidirectional video coding," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2018, pp. 127–130.

[17] J. Boyce and Q. Xu, "Spherical rotation orientation indication for HEVC and JEM coding of 360 degree video," *Proc. SPIE*, vol. 10396, Sep. 2017, Art. no. 103960I.

[18] A. Ahmmed, M. M. Hannuksela, and M. Gabbouj, "Fisheye video coding using elastic motion compensated reference frames," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2027–2031.

[19] G. Jin, A. Saxena, and M. Budagavi, "Motion estimation and compensation for fisheye warped video," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2751–2755.

[20] B. Vishwanath, T. Nanjundaswamy, and K. Rose, "Motion compensated prediction for translational camera motion in spherical video coding," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–4.

[21] B. Vishwanath and K. Rose, "Spherical video coding with motion vector modulation to account for camera motion," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.

[22] M. Zhou, *AHG8: A Study on Equi-Angular Cubemap Projection*, document JVET-G0056, 2017.

[23] G. V. Auwera, M. Coban, and M. Karczewicz, *AHG8: Equatorial Cylindrical Projection for 360-Degree Video*, document JVET-F0026, 2017.

[24] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, *JVET Common Test Conditions and Evaluation Procedures for 360° Video*, document JVET-F1030, Apr. 2017.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.

[25] D. Kim, S. Pathak, A. Moro, A. Yamashita, and H. Asama, "SelfSphNet: Motion estimation of a spherical camera via self-supervised learning," *IEEE Access*, vol. 8, pp. 41847–41859, 2020.
[26] F.-E. Wang *et al.*, "Self-supervised learning of depth and camera motion from 360° videos," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 53–68.
[27] (2016). *High Efficiency Video Coding Test Model, HM-16.15*. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/
[28] Y. He, B. Vishwanath, X. Xiu, and Y. Ye, *AHG8: InterDigital's Projection Format Conversion Tool*, document JVET-D0021, 2016.
[29] A. Abbas, *GoPro Test Sequences for Virtual Reality Video Coding*, document JVET-C0021, 2016.
[30] A. Abbas and B. Adsumilli, *New Gopro Test Sequences for Virtual Reality Video Coding*, document JVET-D0026, 2016.
[31] E. Asbun, Y. Ye, P. Hanhart, Y. He, and Y. Ye, *Test Sequences for Virtual Reality Video Coding From Interdigital*, document JVET-G0055, 2017.
[32] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33 ITU-T Q6/16, Austin, TX, USA, Apr. 2001.
[33] B. Vishwanath, Y. He, and Y. Ye, *AHG8: Area Weighted Spherical PSNR for 360 Video Quality Evaluation*, document JVET-D0072, Chengdu, China, 2016.
[34] X. Xiu, Y. He, Y. Ye, and B. Vishwanath, "An evaluation framework for 360° video compression," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

Bharath Vishwanath (Member, IEEE) received the B.E. degree in electronics and communications engineering from the National Institute of Technology Karnataka, India, in 2014, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara (UCSB), Santa Barbara, CA, USA, in 2016 and 2021, respectively. He is currently a Multimedia Research Scientist with Bytedance, San Diego, CA, USA. He has interned with Interdigital Communications Inc., San Diego, CA, USA, during the Summer of 2016 and 2017, and Dolby Laboratories during the Summer 2019. His research interests include video coding, non-convex optimization, and information theory.

Tejaswi Nanjundaswamy (Member, IEEE) received the B.E. degree in electronics and communications engineering from the National Institute of Technology Karnataka, India, in 2004, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara (UCSB), Santa Barbara, CA, USA, in 2009 and 2013, respectively. He worked at Ittiam Systems, Bengaluru, India, from 2004 to 2008, as a Senior Engineer on audio codecs and effects development. He was a Postdoctoral Researcher at the Signal Compression Laboratory, UCSB, from 2013 to 2017. He is currently a Audio Codec Engineer at Apple, Cupertino, CA, USA. He is an Associate Member of the Audio Engineering Society (AES). He has won the Student Technical Paper Award at the AES 129th Convention.

Kenneth Rose (Life Fellow, IEEE) received the Ph.D. degree from the California Institute of Technology, Pasadena, in 1991. Then, he joined the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently a Distinguished Professor. His main research activities are in the areas of information theory and signal processing and include rate-distortion theory, source and source-channel coding, audio-video coding and networking, pattern recognition, and non-convex optimization. He has published over 350 peer-reviewed papers in these fields. A long-standing interest of his is in the relations between information theory, estimation theory, statistical physics, and their potential impact on fundamental and practical problems in diverse disciplines. Recently, he was the senior coauthor of a paper for which his students received the 2015 IEEE Signal Processing Society Young Author Best Paper Award. His earlier awards include the 1990 William R. Bennett Prize Paper Award of the IEEE Communications Society and the 2004 and 2007 IEEE Signal Processing Society Best Paper Awards.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on March 14,2022 at 07:46:04 UTC from IEEE Xplore. Restrictions apply.