Advanced Spherical Motion Model and Local Padding for 360° Video Compression

Li Li* Member, IEEE, Zhu Li† Senior Member, IEEE, Xiang Ma, Haitao Yang†, and Houqiang Li†, Senior Member, IEEE

Abstract—The 360° video compression has two main challenges due to projection distortions, namely, the geometry distortion and the face boundary discontinuity. There are some tradeoffs between selecting equi-rectangular projection (ERP) and polyhedron projection. In ERP, the geometry distortion is severer than the face boundary discontinuity; while for the polyhedron projections, the face boundary discontinuity is severer than the geometry distortion. These two distortions will have side effects on the motion compensation and undermine the compression efficiency of the 360° video. In this paper, an integrated framework is developed to handle these two problems to improve coding efficiency. The proposed framework mainly has two key contributions. First, we derive a unified advanced spherical motion model to handle the geometry distortion of different projection formats for the 360° video. When fitting the projection between the various projection formats and the sphere into the unified framework, a specific solution can be obtained for each projection format. Second, we propose a local 3D padding method to handle the face boundary discontinuity between the neighboring faces in various projection formats of the 360° video. The local 3D padding method can be applied to different projection formats through setting different angles between neighboring faces. These two methods are independent of each other and can also be combined into an integrated framework to achieve a better rate-distortion performance. The proposed framework can be seamlessly integrated into the latest video coding standard high-efficiency video coding. The experimental results demonstrate that introducing proposed coding tools can achieve significant bitrate savings compared with the current state-of-the-art method.

Index Terms—360-degree video compression, projection distortion, advanced spherical motion model, local 3D padding, high efficiency video coding.

I. INTRODUCTION

THE 360-degree video [1], which can bring immersive experiences to the viewers, is typically recorded using multiple cameras, or a camera that contains multiple camera lenses. The resulting scenes from various cameras or different lenses are then stitched to form the 360-degree video. Recently, the 360-degree video is becoming more and more popular due to the emergence of various commercial Head-mounted displays [2] such as HTC Vive, Samsung Gear VR, and Oculus Rift. However, since the 360-degree video provides the viewers with the scenes from 360 degrees, 8K or even higher resolution is needed to guarantee a satisfactory visual experience. Therefore, there is an urgent need to develop efficient compression algorithms for the 360-degree video for both transmission and storage purposes.

To better utilize the dominant block-based video coding standards such as H.264/Advanced Video Coding [3] and H.265/High Efficiency Video Coding (HEVC) [4], the 360-degree video is usually projected from sphere to the 2D formats before the compression. As shown in Fig. 1, there are many methods to project a sphere to the 2D formats such as Equi-rectangular projection (ERP), Cube Map projection (CMP), and Octahedron projection (OHP). The red and blue rectangles represent the obvious geometry distortions and discontinuous face boundaries due to the various projections, respectively. If we consider the ERP as a projection from the sphere to a specific one-face polyhedron, different kinds of projections mentioned above can be considered as projections from sphere to polyhedrons with a different number of faces. We can obviously see from Fig. 1 that, with the increase of the number of faces, the geometry distortion between the sphere and the polyhedron will become smaller while there will be more discontinuous boundaries. Therefore, there are two contradictory projection distortions in various projection formats: the geometry distortion and the face boundary discontinuity.

The main influence of the geometry distortion to the compression is that the motion model of the distorted blocks will become irregular. The motions of all the pixels in a local block will not be the same after projection. Therefore, the simple translational motion model adopted in HEVC will be unable to efficiently characterize the irregular motions.
Some recent researchers tried to integrate high-order motion models [6], [7] such as the affine motion model and the perspective motion model into the video coding framework to try to characterize the complex motions existing in natural videos. However, since such general motion models have not taken the characteristics of the geometry distortions of the 360-degree video into consideration, they are unable to fully characterize the irregular motions in the 360-degree video. There are also some advanced motion models [8], [9] designed specially for characterizing the irregular motion models in the fish-eye captured videos [10]. Since those algorithms also have not utilized the projections which lead to the irregular motions, it is difficult to extend those algorithms to the 2D projection formats of the 360-degree video.

While the geometry distortion mainly has an influence on the motion model, the face boundary discontinuity has a very significant influence on the reference frame construction. If the actual motion vector (MV) of the current block happens to cross the face boundary, the reference frame without change will provide a prediction block with quite large texture discontinuities. Such a prediction block with texture discontinuities is absolutely unreasonable and will potentially lead to some serious quality degradation. In the standard-based video coding framework, a simple padding scheme using the nearest neighbor is applied to pad the unavailable pixels when the actual MV crosses the picture boundary [11]. However, such a simple scheme used in the picture boundary is unsuitable to be extended to the face boundary of the 360-degree video. There are also some works [12]–[16] trying to handle the face boundary discontinuity in the 360-degree video. All those works are with very similar ideas compared with the proposed method in this paper to extend the current face using the pixels of the neighboring faces. However, all those methods extend the face boundaries when constructing the reference frames which may increase the memory requirements.

As we know, the reference frame and the motion vector together will determine the motion compensation of each pixel. If any of these two parts happens to be inaccurate, the accuracy of the motion compensation may be influenced. Therefore, we propose an integral framework in this paper to handle the geometry distortion and the face boundary discontinuity, which may lead to inaccurate reference frame and motion vector, respectively. The framework can be seamlessly integrated into the modern video coding standards, e.g. HEVC. In summary, the proposed framework mainly has the following key contributions.

- We introduce an advanced spherical motion model to characterize the geometry distortions. Through projecting the blocks in the 2D plane back to the sphere without any geometry distortion, we derive a unified advanced spherical motion model framework for all projection formats to improve the coding efficiency.
- We propose to use a local 3D padding method to solve the problem of face boundary discontinuity. We try to project the pixels from the neighboring faces to the current face to guarantee texture continuity when the MV of the current block crosses the face boundary.
- The proposed advanced spherical motion model and the local 3D padding method can be combined into an integrated framework to deal with the two contradictory projection distortions. The integrated framework can always lead to obvious bitrate savings under various projection formats.

We perform a number of experiments to verify the efficiency of the proposed framework integrated with HEVC. Compared to the HEVC main profile, our proposed techniques altogether can achieve significant bitrate savings. Note that the basic ideas of the proposed coding tools have been provided for the cubic 4 × 3 format in [17] and [18]. In this paper, a more general and comprehensive solution is provided to adapt the method to various projection formats. More specifically, there are mainly the following differences.

- For both the advanced spherical motion model and the 3D padding methods, we have proposed a more general solution to adapt them to various projection formats. Also, we have integrated these two algorithms into a uniform motion compensation framework.
- For the 3D padding methods, we have provided a local 3D padding method to provide similar R-D performance improvement but with obviously lower encoding and decoding complexity compared with the global padding method in [18].
- We have performed more experiments in this paper to demonstrate the effectiveness of the proposed algorithm. First, more quality metrics such as S-PSNR [19] and WS-PSNR [20], which are more suitable for the 360-degree video are used to measure the performance of the proposed algorithm. Second, we have verified that the combination of the two algorithms can lead to a better performance.

This paper is organized as follows. In section II, we will give an overview of the related works. The proposed integrated...
encoding and decoding framework will be introduced in section III. The above mentioned advanced spherical motion model and local 3D padding method will be introduced in detail in section IV. In section V, we will introduce how these algorithms can be adapted to different projection formats. Section VI will give the detailed experimental results. Section VII concludes the whole paper.

II. RELATED WORK

A. Geometry Distortion

The advanced motion model to characterize the complex motions can be divided into two groups: the generalized high-order motion model and the specific advanced motion model. In as early as 1993, Seferidis and Ghanbari [21] pointed out that the high-order motion models including the affine motion model, the bilinear motion model, and the perspective motion model are more efficient to characterize the complex motions compared with the translational motion model. Among all the high-order motion models introduced in [21], the affine motion model has received the most attention of research as an advanced motion model in video compression areas. For example, Wiegand et al. [3] proposed to integrate several global affine motion models into the video coding framework to generate several warped reference frames to obtain a better prediction. Cheung and Siu [22] tried to use the neighboring information to estimate the affine motion parameters of the current block and added an affine mode into the motion decision process to improve the performance.

Especially, due to the introduction of the large blocks [23] into HEVC, the researches on affine motion model are becoming more and more. Narroschke and Swoboda [24] first pointed out that the affine motion model was more suitable for the large blocks. Huang et al. [7] designed a quite complex framework including the affine skip, the affine merge, and the affine inter modes to try to improve the coding efficiency. Chen et al. [25] further developed the affine merge mode in [7] to incorporate with the merge mode of the translational motion model. Moreover, Heithausen and Vorwerk [26] found that the four-parameter affine motion model was more efficient compared with the six-parameter affine motion model due to bits savings of two affine motion parameters. Most recently, Li et al. [27] designed a four-parameter affine motion framework with a gradient-based affine motion estimation (ME) algorithm and several fast affine motion compensation (MC) tools to obtain a better trade-off between the R-D performance and the complexity. Based on the framework, Nam et al. [28] further simplified the motion vector accuracy and filter precision while keeping the performance of the affine motion model. Zou et al. [29], [30] pointed out that the six-parameter affine motion model combined with the four-parameter affine motion model can provide some additional gains. In summary, the generalized high-order motion models can improve the coding efficiency for most natural sequences. However, they cannot exploit fully the projection information from sphere to specific projection format, thus cannot achieve satisfiable R-D performance for various projection formats.

Besides the generalized high-order motion models, there are also many works focusing on the compression of the fish-eye based videos. For example, Jin et al. [8] proposed to use the equidistant mapping to characterize the warping in the fish eye lens. An efficient MV prediction scheme was also proposed to transmit the model parameters efficiently. Besides, Ahmed et al. [9] introduced the elastic model into HEVC to better describe the object motions in the fish eye cameras. Compared with the method in [8], a more sophisticated method was applied in [9] to more accurately characterize the geometry distortion. Moreover, Eichenseer et al. [31] proposed a re-mapping method to handle some extreme cases in which the field of view is larger than 180-degrees. However, all these motion models specified for fish eye cameras are unable to efficiently characterize the geometry distortions in various projection formats of the 360-degree video. Recently, Vishwanath et al. [32] proposed to use the 3D rotation model to characterize the irregular motion of the 360-degree video.

B. Boundary Discontinuity

There are actually two situations where the padding method is needed to fill in the unavailable pixels in the picture boundaries. One case is that the MV crosses the picture boundary. In this case, the unavailable pixels will be replaced using their nearest neighbors, which is standardized by HEVC. The other case is that the picture size is not the times of the size of Macroblock in H.264 or coding unit (CU) in HEVC. In this case, we can determine the suitable pixels to fill in since it is an encoder only issue. A simple solution is also to fill the unavailable pixels with their nearest neighbors. Besides, Li et al. [33] tried to optimize the process using the fundamental rate distortion optimization (RDO) theory [34]. However, all these solutions are unsuitable for the filling of the face boundary since no texture continuity can be guaranteed.

Except for the picture boundary, there are also some works focusing on the face boundary discontinuity in the 360-degree video. Sauer and Wien [12], [13] and Sauer et al. [14] proposed to use camera parameters to model the relationship between neighboring faces and extend the pixels in the neighboring face to the current face. The advantage of this work is that it can be easily extended to all kinds of polyhedron projection formats. Besides, He et al. [15] and Coban et al. [16] attempted to extend the face boundary using the projection relationships between the sphere and polyhedron formats which is the same as the proposed algorithm in this paper. However, their methods will construct a larger reference frame which will increase the memory requirements.

III. THE PROPOSED CODING FRAMEWORK

The framework of the proposed advanced spherical motion model and the local 3D padding method combined with the HEVC encoder and decoder are shown in Fig. 2 and Fig. 3, respectively. In the following, we will briefly introduce how the proposed methods can be fit in the HEVC context in a high level.
1) Advanced Spherical Motion Model: In the encoder as shown in Fig. 2, the advanced spherical motion model provides the other choice of the motion model to better characterize the geometry distortion besides the translational motion model. The RDO is used to determine whether the translational motion model or the advanced spherical motion model will be applied for MC, and one flag is signaled to the decoder to indicate which motion model is used. For the advanced motion model, one MV is also signaled to the decoder to derive all the MVs of the current block. The MV can be encoded using merge mode or advanced motion vector prediction (AMVP) mode similar to the translational MV.

In the decoder as shown in Fig. 3, when the MV crosses the face boundary, the corresponding reference frame will be extended to guarantee the continuous texture of the prediction block in both the ME and MC processes. In the decoder as shown in Fig. 3, the same padding process will be performed when the MV crosses the face boundary in the MC process.

IV. THE FUNDAMENTAL PRINCIPLE OF THE PROPOSED ALGORITHMS

A. The Advanced Spherical Motion Model

In the proposed advanced spherical motion model, motivated by the fact that the 360-degree videos are captured in the 3D sphere, we try to project the 2D pixel coordinates to the 3D sphere and derive the motions of each pixel in 2D space according to the motions on the sphere. The real motions of each pixel between neighboring frames in the sphere can be rather complex due to the combination of the camera and object motions. However, since the change of the neighboring frames is usually quite small, we approximate the complex motions in the 3D sphere as translational motion like we usually do for the general 2D videos. In this way, only one MV is needed to determine the MVs of all the pixels for the current block.

Let \( f(x, y) \) be the function that maps the 2D coordinate to the 3D coordinate. We first project the pixels with 2D coordinate \((x_0, y_0), (x_1, y_1), \text{ and } (x_2, y_2)\) to the pixels on the sphere space with 3D coordinate \((x_0, y_0, z_0), (x_1, y_1, z_1), \text{ and } (x_2, y_2, z_2)\).
The MV of the center of the left block is calculated as

\[(x_2, y_2, z_2) = f(x_0, y_0, z_0)\]

The detailed forms of \(f\) vary along with the change of the 2D projection formats, which will be explained in more details in the next section.

Then utilizing the approximation that the 3D motion model on the sphere is translational, we can derive the corresponding pixel \((x_3, y_3, z_3)\) of \((x_0, y_0, z_0)\) as follows.

\[
\begin{align*}
(x_3 - x_2) &= x_1 - x_0 \\
y_3 - y_2 &= y_1 - y_0 \\
z_3 - z_2 &= z_1 - z_0
\end{align*}
\]

After that, we can use \(f^{-1}\) to project the \((x_3, y_3, z_3)\) back to the \((x_0, y_0, z_0)\) as follows.

\[
(x_0, y_0, z_0) = f^{-1}(x_3, y_3, z_3)
\]

The \(x_0\) and \(y_0\) may be fractional pixels after the inverse projection. Therefore, we need to interpolate those pixels during the MC process. Similar to the affine motion model based MC framework in [27], we will round the fractional pixels to 1/64 pixel precision. Then the discrete cosine transform based interpolation filter (DCTIF) [35] is used to interpolate the prediction block. To obtain a good balance between the performance and the complexity, the basic unit for MV is set to a 4 × 4 block instead of a pixel. The MV of the top left pixel of a 4 × 4 block is used as the MV of the 4 × 4 block. After obtaining the prediction block using MC for each 4 × 4 block, the residue block will finally be obtained through subtracting the prediction block from the original one.

Besides the advanced spherical motion model based MC process, the MV prediction from the neighboring blocks also has significant influences on the coding efficiency. Taking the left block as an example as shown in Fig. 5, assuming that the MV of the center of the left block is \(MV_{0}\), then the MV predictor \(MV_{1}\) of the center of the current block can be derived using the same process as the previous subsection according to their relative distance. It should be noted that the MV predictor is rounded to 1/4 pixel precision to be consistent with the HEVC translational motion model and reduce the overhead bitrate.

For the HEVC translational motion model, two MV prediction schemes including merge and AMVP are used to fully exploit the MV dependency among neighboring blocks. Under the proposed scheme, we also involve the merge and AMVP schemes into the advanced spherical motion model to better improve the coding efficiency.

For the merge scheme, as shown in Fig. 6 (a), we will traverse all the blocks A, B, C, D, and E to get the merge candidates. If no valid merge candidates are found, the advanced spherical motion model is disabled. The maximum number of the merge candidates is set as 1 to avoid too many duplications with the merge candidates in the translational motion model. Besides, the proposed motion model and the traditional motion model will be the same when the MV is zero. Therefore, the zero motion is also considered as invalid when searching for a merge candidate.

For the AMVP scheme, as shown in Fig. 6 (b), we will first search the left block A0 and A1, and then search the above predictor B0, B1, and B2. Since in the AMVP scheme, we will perform motion estimation to find the optimal MV, the zero MV is used as the MV predictor when the MVs of all the neighboring blocks are invalid. The maximum number of the AMVP candidates is also set as 1. Besides, under the AMVP mode, the neighboring blocks may point to different reference frames from the given one. In this case, the MV scaling operations are applied.

### B. Local 3D Padding

The essence of the local 3D padding method is to project the pixels in the other faces to the extension of the current face to guarantee exact texture continuity. In this way, if the MV of the current block points out of the face boundary, we can still obtain a prediction block with continuous texture. For different kinds of polyhedron projection formats, the main differences are the number of faces and the angle between different faces. In the following, we will give a general form of the face extension.
Fig. 7 gives a typical example to show the face extension with the angle \( \theta \) between two faces. \( O \) is the center of both the polyhedron and the sphere, and \( ABCD \) and \( BDF \) are the current and its corresponding neighboring face. Then for a given point \( T \) in the extension of the current face, our target is to find the intersection point \( H \) of \( OT \) and \( BEDF \) in the neighboring face. For a specified projection format, we can assume that the lengths of \( OO' \) and \( O'J \) are \( L_{OO'} \) and \( L_{O'O'} \), respectively. Also, for a given point \( T \) in the extension of the current face, we can easily obtain the lengths of \( TK, KJ, KO' \) as \( L_{TK}, L_{KJ}, L_{KO'} \) according to the coordinate of \( T \). Then since \( \triangle OO'K \) and \( \triangle GG'K \) are similar triangles, we can obtain an equation as follows.

\[
\frac{L_{GG'}}{L_{OO'}} = \frac{L_{KJ}}{L_{KO'}} = \frac{L_{KJ} - L_{JG'}}{L_{KO'}} \tag{4}
\]

As the angle between the two neighboring faces is \( \theta \), we can calculate the lengths of \( GG' \) and \( JG' \) as follows.

\[
L_{GG'} = L_{JG} \times \sin(\pi - \theta) \tag{5}
\]

\[
L_{JG'} = L_{JG} \times \cos(\pi - \theta) \tag{6}
\]

Then we substitute (5) and (6) into (4), we can calculate \( L_{JG} \) as follows.

\[
L_{JG} = \frac{L_{OO'} \times L_{KJ}}{L_{KO'} \times \sin(\pi - \theta) + L_{OO'} \times \cos(\pi - \theta)} \tag{7}
\]

\( L_{JG} \) is one component of the coordinate of \( H \) in the neighboring face.

For the other component of the coordinate of \( H \), we first observe that \( \triangle O'JS \) and \( \triangle O'G'H' \) are similar triangles. Thus \( L_{G'H'} \) can be derived as follows.

\[
L_{G'H'} = \frac{L_{JS} \times L_{O'G'}}{L_{O'J}} = \frac{L_{JS} \times (L_{O'J} + L_{JG'})}{L_{O'J}} \tag{8}
\]

The only unknown variable in (8) is \( L_{JS} \). According to the similar relationship between \( \triangle O'JS \) and \( \triangle O'K'T \), \( L_{JS} \) can be calculated as

\[
L_{JS} = \frac{L_{KT} \times L_{O'J}}{L_{KO'}} \tag{9}
\]

Using (8) and (9), the other component of the coordinate of the point \( H \) can be obtained. The above formulas can be adapted to different polyhedron formats with different \( \theta \).

If we simply extend the face border in the original reference frames, the information of the original reference frame will be destroyed. There are mainly two methods to solve this problem, one way is to store multiple versions of the reference frames. For example, if we have six faces, we can store a group of reference frames for each face [15], [16]. As a result, such a scheme will increase the reference buffer size. The other way is to switch the reference frame between the original one and the extended one when encoding each PU [18]. The method is inevitably very time-consuming if the switch is performed multiple times. Besides, most of the zones calculated are not meaningful at all. Therefore, in this paper, a local 3D padding method is proposed to decrease the decoding complexity significantly. We try to extend only a local zone near the current block when performing ME and MC. Note that we extend the block by extra several pixels in horizontal or vertical directions when the fractional MVs are applied. If the fractional MVs are applied, the Luma component will be further extended by 4 pixels since the 8-tap DCTIF [35] is used, and the Chroma component will be further extended by 2 pixels since the 4-tap DCTIF [35] is used.

V. ADAPTATION OF THE PROPOSED ALGORITHMS TO VARIOUS PROJECTION FORMATS

A. The Advanced Spherical Motion Model

To adapt the proposed advanced spherical motion model to various projections, the main problem is to determine the function \( f \) in (1). The function \( f \) is always invertible since there is always a one-to-one correspondence between various 2D projections and the 3D sphere. To make the manuscript more concise, we mainly introduce the form of \( f \) under various projection formats.

1) ERP: Under the ERP format, the frame width \( W_{frame} \) is always two times of the frame height \( H_{frame} \). As shown in Fig. 8, we first calculate the sphere coordinates \( \theta, \phi \), and the sphere radius \( R \) according to the pixel coordinate \((x_u, y_u)\) [36].

\[
\begin{aligned}
\theta &= y_u / H_{frame} \cdot \pi \\
\phi &= x_u / W_{frame} \cdot \pi \\
R &= H_{frame} / \pi = W_{frame} / 2\pi
\end{aligned}
\tag{10}
\]
Then the sphere coordinate of a specified pixel will be calculated as

\[
\begin{align*}
x_s &= R \cdot \sin \theta \cdot \cos \phi \\
y_s &= R \cdot \sin \theta \cdot \sin \phi \\
z_s &= R \cdot \cos \theta
\end{align*}
\] (11)

According to Eqs. (10) and (11), \((x_u, y_u)\) can be projected to \((x_s, y_s, z_s)\).

In this way, the 3D translation motion model expressed in (2) can be converted to the form expressed in (12), as shown at the bottom of the next page. First, the equation in z direction indicates that if two points are with the same latitude \((\theta_2 = \theta_0)\), the corresponding points in the reference frame will be also with the same latitude \((\theta_3 = \theta_1)\). Second, if we add the squares of the left terms and right terms together, we can have the equation as in (13), as shown at the bottom of the next page. Eq. (13) can be simplified as

\[
sin^2 \theta_1 + \sin^2 \theta_2 - 2 \sin \theta_1 \sin \theta_2 \cos(\phi_3 - \phi_2) = \sin^2 \theta_1 + \sin^2 \theta_0 - 2 \sin \theta_1 \sin \theta_0 \cos(\phi_1 - \phi_0)
\] (14)

For the points with the same latitude \((\theta_2 = \theta_0 \text{ and } \theta_3 = \theta_1)\), we can have

\[
\phi_3 - \phi_2 = \phi_1 - \phi_0
\] (15)

This equation indicates that the points with the same latitude will be with the same longitude differences. In conclusion, the MVs for the pixels with the same y coordinates are the same in a local block under the ERP format. Third, in a more general case where the latitudes of the points are different, we can derive from the above equations that the differences in both x and y directions will be larger when the points closer to the polar areas. This is in correspondence to the geometry distortions of the ERP format, which is the essence why the proposed advanced spherical motion model can bring significant R-D performance improvements. A typical \(4 \times 4\) block correspondence can be seen from Fig. 9.

2) Polyhedrons: For the polyhedron projection formats with multiple faces such as CMP and OHP, we always need to first convert the 2D coordinates to the 3D coordinates in the polyhedrons, and then the coordinates in the polyhedrons will be projected to the sphere. Here, CMP is used as an example to explain the detailed form of projection \(f\). The six faces of the CMP format can be packed with various forms, such as \(4 \times 3\) and \(3 \times 2\) as shown in Fig. 10. The \(4 \times 3\) format is the direct unfold of the cube map that the frame width is 4 times of the face width and the frame height is 3 times of the face height. The \(3 \times 2\) format is a compact unfold of the cube map that the frame width is 3 times of the face width and the frame height is 2 times of the face height. Here, the most commonly used compact \(3 \times 2\) format is used as an example.

Under the compact \(3 \times 2\) format, the 2D coordinate \((x_u, y_u)\) is first converted to the coordinate on the surface of a cube \((x_c, y_c, z_c)\) as shown in Fig. 11. Note that the coordinate of the top left pixel is set as \((0, 0)\) in the unfold cube map, the coordinate of the center pixel is set as \((0, 0, 0)\) in the cube in 3D space. The width and height of each face are \(W_{\text{face}}\) and \(H_{\text{face}}\), respectively. The right, rear, left, bottom, front, and top faces are named as \(F_{\text{right}}, F_{\text{rear}}, F_{\text{left}}, F_{\text{bottom}}, F_{\text{front}}, \) and \(F_{\text{top}}\), respectively. Then the 2D coordinate \((x_u, y_u)\) can be converted to \((x_c, y_c, z_c)\) through the following formula depending on difference faces [36].

\[
x_c = \begin{cases} 
  y_u - 3 \times H_{\text{face}}/2 & \text{if } face = F_{\text{top}} \\
  y_u - 3 \times H_{\text{face}}/2 & \text{if } face = F_{\text{front}} \\
  y_u - 3 \times H_{\text{face}}/2 & \text{if } face = F_{\text{bottom}} \\
  3 \times W_{\text{face}}/2 - x_u & \text{if } face = F_{\text{right}} \\
  -H_{\text{face}}/2 & \text{if } face = F_{\text{left}}
\end{cases}
\] (16)

\[
y_c = \begin{cases} 
  5 \times W_{\text{face}}/2 - x_u & \text{if } face = F_{\text{top}} \\
  H_{\text{face}}/2 & \text{if } face = F_{\text{front}} \\
  x_u - W_{\text{face}}/2 & \text{if } face = F_{\text{bottom}} \\
  W_{\text{face}}/2 - x_u & \text{if } face = F_{\text{right}} \\
  -H_{\text{face}}/2 & \text{if } face = F_{\text{left}} \\
  x_u - 5 \times W_{\text{face}}/2 & \text{if } face = F_{\text{left}}
\end{cases}
\] (17)

\[
z_c = \begin{cases} 
  H_{\text{face}}/2 & \text{if } face = F_{\text{top}} \\
  x_u - 3 \times H_{\text{face}}/2 & \text{if } face = F_{\text{front}} \\
  -H_{\text{face}}/2 & \text{if } face = F_{\text{bottom}} \\
  H_{\text{face}}/2 - y_u & \text{if } face = F_{\text{right}} \\
  H_{\text{face}}/2 - y_u & \text{if } face = F_{\text{left}} \\
  H_{\text{face}}/2 - y_u & \text{if } face = F_{\text{left}}
\end{cases}
\] (18)

Then the coordinate on the surface of a cube \((x_c, y_c, z_c)\) is projected to the coordinate on the sphere \((x_s, y_s, z_s)\).
According to Fig. 12, there are two constraints for the coordinate \((x_s, y_s, z_s)\). One constraint is that the \((x_s, y_s, z_s)\) is on the sphere as shown in the following formula.

\[
x_s^2 + y_s^2 + z_s^2 = W^2_{\text{face}}/4
\]  

(19)

The other constraint is that O, M, and N are on the same line.

\[
\frac{x_s}{x_c} = \frac{y_s}{y_c} = \frac{z_s}{z_c}
\]

(20)

Combining these two constraints, the coordinate \((x_s, y_s, z_s)\) can be solved as follows.

\[
x_s = \frac{W_{\text{face}}/2}{\sqrt{1 + \frac{x_c^2}{y_c^2} + \frac{y_c^2}{z_c^2}}}
\]

(21)

\[
y_s = y_c \times \frac{x_s}{x_c}
\]

(22)

\[
z_s = z_c \times \frac{x_s}{x_c}
\]

(23)

There are some special situations where \(x_c, y_c, \) or \(z_c\) may be 0. From Fig. 12, such situations can be easily handled. In summary, after the above two steps, the pixels in the unfold cube map can be projected to the sphere.

B. The Local 3D Padding

1) ERP: The local 3D padding method is not applied to the ERP format in our proposed scheme since there is only one face in the ERP format. We only need to perform one padding for all the CUs in the picture boundary. The results of the extended picture boundary using the 360-degree information instead of copying the pixels in the boundary are shown in Fig. 13. To be more specific, the left boundary is extended using the pixels near the right boundary and the right boundary is extended using the pixels near the left boundary. For both the top and bottom boundaries, the left side pixels are extended using the right side and the right side pixels are extended using the left side. As can be obviously seen from Fig. 13, compared with the original padding method in HEVC, all the boundaries become continuous under the proposed padding method.

\[
\begin{align*}
R \sin \theta_3 \cos \phi_3 - R \sin \theta_2 \cos \phi_2 = R \sin \theta_1 \cos \phi_1 - R \sin \theta_0 \cos \phi_0 \\
R \sin \theta_3 \sin \phi_3 - R \sin \theta_2 \sin \phi_2 = R \sin \theta_1 \sin \phi_1 - R \sin \theta_0 \sin \phi_0 \\
R \cos \theta_3 - R \cos \phi_2 = R \cos \phi_1 - R \cos \phi_0
\end{align*}
\]

(12)

\[
(R \sin \theta_3 \cos \phi_3 - R \sin \theta_2 \cos \phi_2)^2 + (R \sin \theta_3 \sin \phi_3 - R \sin \theta_2 \sin \phi_2)^2
\]

\[
= (R \sin \theta_1 \cos \phi_1 - R \sin \theta_0 \cos \phi_0)^2 + (R \sin \theta_1 \sin \phi_1 - R \sin \theta_0 \sin \phi_0)^2
\]

(13)
algorithms individually using some selected sequences. The characteristics of the selected sequences are shown in Table I. From Table I, we can see that the selected sequences are with various spatial resolutions, bit depths, and motions. Note that all the sequences shown in Table I are with ERP format. They will be automatically converted into the other formats during encoding.

Since the bitrates of various algorithms are not the same, the Bjontegaard Delta-rate (BD-rate) [39] is employed in our experiments for a fair R-D performance comparison. In terms of the objective quality assessment, besides PSNR, we use lots of other quality metrics [36], [40] which are more suitable for the 360-degree video, such as S-PSNR-NN, WS-PSNR, S-PSNR-I, CPP-PSNR, and WS-PSNR (End to End, EtE). A brief introduction on these quality metrics are shown in Table II. Moreover, we also measure the performance of the proposed algorithms according to the quality of some specified view ports PSNR-ViewPort0 (VP0) and PSNR-ViewPort1 (VP1). The view port is the rendered 2D image based on the viewer’s current viewing direction. There are two predefined viewing directions under the 360-degree video compression viewer's current viewing direction. There are two predefined view ports PSNR-ViewPort0 (VP0) and PSNR-ViewPort1 (VP1). The view port is the rendered 2D image based on the viewer’s current viewing direction. There are two predefined viewing directions under the 360-degree video compression standardization group. The view port size of the 8K resolution is 1816 × 1816, while that of the 4K resolution is 856 × 856. More detailed information can be found in [38].

B. The Performance of the Combination of the Local 3D Padding and the Advanced Spherical Motion Model

Table III, Table IV, and Table V show the performances of the combination of the local 3D padding and the advanced spherical motion model for the ERP, CMP, and OHP formats in RA10 and LD10 cases, respectively. From Table III, Table IV, and Table V, we can see that the combination of the proposed algorithms can achieve very consistent R-D performance improvements compared with the HEVC anchor. Different view ports may present different bitrate savings since they are pointing to totally different video contents. For all the other quality metrics, very consistent bitrate savings can be observed. Note that since the combined algorithm presents similar R-D performance under different quality metrics, the S-PSNR-NN will be used as a representation for all the other 360-degree video specified quality metrics when we show the performance of the algorithms individually.

Taking S-PSNR-NN as an example, the combination of the proposed algorithm can achieve an average of 4.4% and 3.4% bitrate savings for the ERP format in RA10 and LD10 cases, respectively. Especially, for the Chairlift sequence with very obvious motions, the combination of the proposed algorithms can bring as high as 8.3% R-D performance improvement in RA10 case. For the CMP format, the combination of the proposed algorithms can achieve averagely 3.7% and 4.8% R-D performance improvements in RA10 and LD10 cases, respectively. For the DrivingInCountry sequence, the proposed algorithms can achieve over 6.0% bitrate savings in both test conditions. For the OHP format, the combination of the proposed algorithms can save averagely 4.7% and 6.9% bitrates in RA10 and LD10 cases, respectively. For the ERP and polyhedron formats, the majority of the R-D performances comes from the advanced spherical motion model and the local 3D padding, respectively, which will be introduced in detail in the following subsections. That is why the proposed algorithms may bring quite different R-D performance improvements for various sequences in ERP and polyhedron formats.

C. The Performance of the Advanced Spherical Motion Model

1) The Performance Comparison With HEVC Anchor: The performances of the advanced spherical motion model for the ERP, CMP, and OHP formats in RA10 and LD10 cases are shown in Table VI. From Table VI, we can see that the proposed advanced spherical motion model can bring an average of 4.3% and 3.3% R-D performance improvements in RA10 and LD10 cases for the ERP format, respectively. The experimental results obviously demonstrate that the proposed algorithm is able to characterize the irregular motions caused by the geometry distortions. After a careful observation of the test sequences, we can see that all four sequences are with obvious camera motions. However, the depth of field of the Chairlift sequence is relatively smaller than the other sequences. Therefore, the irregular motions in the polar areas will be more obvious, and thus the proposed algorithms can achieve the largest performance improvement. For the AerialCity sequence, the depth of field is larger than the other sequences. So the irregular motions will be easily approximated by the translational motion model. Therefore, the performance improvement provided by the proposed algorithm is limited.

From Table VI, we can also see that the proposed advanced motion model can bring 1.4% and 0.7% R-D performance improvements in RA10 and LD10 cases for the CMP format, respectively. For the OHP formats, the performance improvements become even smaller due to the even smaller geometry distortions. Compared with the ERP format, the proposed advanced spherical motion model brings much fewer performance improvements for the polyhedron projections.

From the R-D performance shown above, we can see that the R-D performance of the proposed spherical advanced motion model in RA case is better than that in LD case. The performance difference mainly comes from the different reference settings in RA and LD settings. In the LD setting, the nearest frame is always the most frequently used reference frame. In this case, the proposed motion model could be more approximate to the translational motion model. In the RA setting, the temporal distance between the current frame and the reference frames is relatively larger. So the proposed
motion model will be more advantageous compared with the translational motion model in describing the deformation caused by rectilinear projection.

To better show the reason why the proposed advanced spherical motion model can bring some performance improvements, we also give an example of the blocks using the proposed advanced spherical motion model in a frame under the ERP format as shown in Fig. 15. As shown in Fig. 15, the blocks with a green square are those using the advanced spherical motion model instead of the translational motion model. The blocks using the advanced spherical motion model are mostly in the zones with larger geometry distortions, which are far away from the equator in the ERP format. The experimental results obviously show that the proposed advanced spherical motion model can better characterize the geometry distortions compared with the translational motion model, and thus can lead to a better R-D performance.

<table>
<thead>
<tr>
<th>Quality metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-PSNR-NN</td>
<td>Calculate PSNR based on a set of points evenly sampled on the sphere; The sample value at the projected position is taken from the nearest neighbor integer position.</td>
</tr>
<tr>
<td>WS-PSNR</td>
<td>Calculate PSNR based on all samples; The distortion is weighted by sample area on the corresponding spherical surface.</td>
</tr>
<tr>
<td>S-PSNR-I</td>
<td>Calculate PSNR based on a set of points evenly sampled on the sphere; The sample value at the projected position is interpolated with neighboring samples at integer positions.</td>
</tr>
<tr>
<td>CPP-PSNR</td>
<td>Calculate PSNR in Craters Parabolic Projection format.</td>
</tr>
<tr>
<td>WS-PSNR (ESE)</td>
<td>WS-PSNR calculated end to end.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table II</th>
<th>A BRIEF INTRODUCTION OF THE 360-Degree Video Specified Video Quality Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test case</td>
<td>Test sequence</td>
</tr>
<tr>
<td>RA10</td>
<td>DrivingInCountry</td>
</tr>
<tr>
<td></td>
<td>AerialCity</td>
</tr>
<tr>
<td></td>
<td>Chairlift</td>
</tr>
<tr>
<td></td>
<td>SkateboardInLot</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
</tr>
<tr>
<td>LD10</td>
<td>DrivingInCountry</td>
</tr>
<tr>
<td></td>
<td>AerialCity</td>
</tr>
<tr>
<td></td>
<td>Chairlift</td>
</tr>
<tr>
<td></td>
<td>SkateboardInLot</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table III</th>
<th>THE PERFORMANCE OF THE COMBINATION OF THE LOCAL 3D PADDING AND THE ADVANCED SPHERICAL MOTION MODEL FOR THE ERP FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test case</td>
<td>Test sequence</td>
</tr>
<tr>
<td>RA10</td>
<td>DrivingInCountry</td>
</tr>
<tr>
<td></td>
<td>AerialCity</td>
</tr>
<tr>
<td></td>
<td>Chairlift</td>
</tr>
<tr>
<td></td>
<td>SkateboardInLot</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
</tr>
<tr>
<td>LD10</td>
<td>DrivingInCountry</td>
</tr>
<tr>
<td></td>
<td>AerialCity</td>
</tr>
<tr>
<td></td>
<td>Chairlift</td>
</tr>
<tr>
<td></td>
<td>SkateboardInLot</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table IV</th>
<th>THE PERFORMANCE OF THE COMBINATION OF THE LOCAL 3D PADDING AND THE ADVANCED SPHERICAL MOTION MODEL FOR THE CMP FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test case</td>
<td>Test sequence</td>
</tr>
<tr>
<td>RA10</td>
<td>DrivingInCountry</td>
</tr>
<tr>
<td></td>
<td>AerialCity</td>
</tr>
<tr>
<td></td>
<td>Chairlift</td>
</tr>
<tr>
<td></td>
<td>SkateboardInLot</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
</tr>
<tr>
<td>LD10</td>
<td>DrivingInCountry</td>
</tr>
<tr>
<td></td>
<td>AerialCity</td>
</tr>
<tr>
<td></td>
<td>Chairlift</td>
</tr>
<tr>
<td></td>
<td>SkateboardInLot</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
</tr>
</tbody>
</table>

Authorized licensed use limited to: University of Missouri-Kansas City. Downloaded on February 28, 2020 at 16:52:19 UTC from IEEE Xplore. Restrictions apply.
Fig. 15. A typical example of the blocks using the advanced spherical motion model, the sequence Chairlift, POC 14, QP 27.

### TABLE VI
THE PERFORMANCE OF THE ADVANCED SPHERICAL MOTION MODEL COMPARED WITH HEVC ANCHOR UNDER S-PSNR-NN

<table>
<thead>
<tr>
<th>Test sequence</th>
<th>ERP</th>
<th>CMP</th>
<th>OHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrivingInCountry</td>
<td>-5.9%</td>
<td>-3.3%</td>
<td>-3.1%</td>
</tr>
<tr>
<td>AerialCity</td>
<td>-0.8%</td>
<td>-1.2%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>Chairlift</td>
<td>-8.1%</td>
<td>-5.8%</td>
<td>-0.9%</td>
</tr>
<tr>
<td>SkateboardInLot</td>
<td>-2.4%</td>
<td>-2.6%</td>
<td>-1.4%</td>
</tr>
<tr>
<td><strong>Avg.</strong></td>
<td><strong>-4.3%</strong></td>
<td><strong>-3.3%</strong></td>
<td><strong>-1.4%</strong></td>
</tr>
</tbody>
</table>

### TABLE VII
THE PERFORMANCE COMPARISON OF THE ADVANCED SPHERICAL MOTION MODEL WITH THE AFFINE MOTION MODEL [27]

<table>
<thead>
<tr>
<th>Test case</th>
<th>Spherical</th>
<th>Affine [27]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrivingInCountry</td>
<td>-5.9%</td>
<td>-5.2%</td>
</tr>
<tr>
<td>AerialCity</td>
<td>-0.8%</td>
<td>-1.0%</td>
</tr>
<tr>
<td>Chairlift</td>
<td>-8.1%</td>
<td>-3.6%</td>
</tr>
<tr>
<td>SkateboardInLot</td>
<td>-2.4%</td>
<td>-3.9%</td>
</tr>
<tr>
<td><strong>Avg.</strong></td>
<td><strong>-4.3%</strong></td>
<td><strong>-3.4%</strong></td>
</tr>
</tbody>
</table>

2) The Performance Comparison With Affine Motion Model:
The performance of the advanced spherical motion model compared with the affine motion model under the ERP format in RA10 case is shown in Table VII. As can be seen from Table VII, the proposed algorithm can achieve an average of 0.9% R-D performance improvement compared with the affine motion model. It should be mentioned that the proposed algorithm can achieve as high as 4.5% bitrate savings for the Chairlift sequence, which obviously demonstrates the advantage of the proposed algorithm over the affine motion model. However, there is also about 1.5% performance loss compared with the affine motion model for the SkateboardInLot sequence. Since the affine motion model in [27] is with four parameters, it is able to better characterize some motions compared with the advanced spherical motion model, which is with only two parameters.

### D. The Performance of the Local 3D Padding

1) The Performance Comparison With HEVC Anchor:
The performances of the local 3D padding method for the ERP, CMP, and OHP formats in RA10, and LD10 cases are shown in Table VIII. From Table VIII, we can see that the proposed 3D padding method can bring averagely 0.4% and 0.7% bitrate savings for the ERP format in RA10 and LD10 cases, respectively. As there is only one face in the ERP format, we just extend the picture boundary to make full use of the 360-degree information. Therefore, the performance improvement provided by the padding method is limited.

It can also be seen from Table VIII that the local 3D padding method can achieve an average of 2.1% and 4.0% bitrate savings for the CMP format in RA10 and LD10 cases, respectively. For the OHP format, the proposed algorithm can achieve averagely as high as 4.2% and 6.5% R-D performance improvements accordingly. Since there are multiple faces boundaries in the polyhedron formats, the proposed local 3D padding algorithm will have influences on lots of areas for the current frame. Therefore, the proposed local 3D padding algorithm can lead to very obvious R-D performance improvement. It is anticipated that the proposed algorithm can achieve even better R-D performance for the projection formats with more face boundaries.

As we have analyzed in Section IV-B, the performance of the local 3D padding method is determined by the motions near the face boundary. Therefore, we can achieve some performance improvements for all the test sequences with some motions in the face boundary. It should be mentioned that for the SkateboardInLot sequence, although the motions along the
presents different trade-offs between the memory requirements. The proposed algorithm will have fewer memory requirements. However, just because of this, the proposed algorithm will replace the original reference pixels and then get them back locally. However, just because of this, the proposed algorithm will have fewer memory requirements. The proposed algorithm presents different trade-offs between the memory requirements and complexity compared with [15] and [16], they can be used for different scenarios to meet the requirements of various applications.

E. The Analysis of the Performance of the Proposed Algorithm Individually and Together

Since the local 3D padding and the advanced spherical motion model can provide a texture continuous reference frame and more accurate motion model, respectively, the combination of them should be able to provide better performance than the sum of the performance improvement individually since a texture continuous reference frame is beneficial for finding a better MV.

For the CMP and OHP formats, we can see from the previous tables that the performance improvement by combining these two methods is slightly larger than the sum of the performance improvements of each individual method. For the ERP format, there seem some cannibalizing effects between these two algorithms from the performance. However, the reason is that we have already considered the influence of the extension in the advanced sphere motion model. Under the advanced sphere motion model individually, since we always make the latitude and longitude in the range of (0, 180) and (0, 360), respectively, we can find the right pixel which is the same position with the padded one when the MV crosses the picture boundary. The only difference is that the pixels used for interpolation may be different. That is exactly why the combination can bring a little performance improvement compared with the advanced spherical motion model.

F. The Performance of the Combination of the Local 3D Padding and Advanced Spherical Motion Model on More Test Sequences

The performances of the combination of the local 3D padding and advanced spherical motion model for the ERP, CMP and OHP formats on the test sequences defined in the CTC in RA10 and LD10 cases are shown in Table XI. From Table XI, we can see that the combination of the proposed algorithms can bring an average of 1.8% and 1.4% R-D performance improvements for the ERP format in RA10 and LD10 cases, respectively. For the CMP format, from Table XI, an average of 1.7% and 2.1% R-D performance improvements in RA10 and LD10 cases can be achieved. The performance improvements for the OHP format are 2.0% and 3.0% for RA10 and LD10 cases, respectively. From Table XI, we can also see that the proposed algorithm can lead to quite good R-D performance improvements for the test sequences with obvious motions while only having neglect losses for the static sequences.

We also show the performance of some test sequences with a higher frame rate to demonstrate the effectiveness of the proposed algorithm as shown in Table XII. As can be seen from Table XII, the combination of the proposed algorithms can still achieve obvious performance improvements for the sequences with frame rate of 60fps. It should be also noted that the performance improvement of the CMP format is larger than that of the ERP format. The reason is

<table>
<thead>
<tr>
<th>Test case</th>
<th>Test sequence</th>
<th>Local</th>
<th>Global [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrivingInCountry</td>
<td>-3.1%</td>
<td>-2.9%</td>
<td></td>
</tr>
<tr>
<td>AerialCity</td>
<td>-1.7%</td>
<td>-1.8%</td>
<td></td>
</tr>
<tr>
<td>Chairlift</td>
<td>-2.2%</td>
<td>-2.1%</td>
<td></td>
</tr>
<tr>
<td>RA10</td>
<td>SkateboardInLot</td>
<td>-1.4%</td>
<td>-1.2%</td>
</tr>
<tr>
<td>Avg.</td>
<td>-2.1%</td>
<td>-2.0%</td>
<td></td>
</tr>
<tr>
<td>Enc.</td>
<td>196%</td>
<td>20153%</td>
<td></td>
</tr>
<tr>
<td>Dec.</td>
<td>119%</td>
<td>644%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test case</th>
<th>Test sequence</th>
<th>Local</th>
<th>Global [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrivingInCountry</td>
<td>-3.6%</td>
<td>-3.1%</td>
<td></td>
</tr>
<tr>
<td>AerialCity</td>
<td>-2.0%</td>
<td>-1.6%</td>
<td></td>
</tr>
<tr>
<td>RA10</td>
<td>SkateboardInLot</td>
<td>-1.6%</td>
<td>-1.2%</td>
</tr>
<tr>
<td>Avg.</td>
<td>-2.5%</td>
<td>-2.1%</td>
<td></td>
</tr>
</tbody>
</table>

The performances of the proposed local 3D padding algorithm with the camera-parameter based global padding method [14]. The experimental results for the CMP format under the S-PSNRM-NN are shown in Table IX. From the experimental results, we can see that the proposed algorithm is with the similar performance compared with our previous work. However, due to the use of the local padding method, the local 3D padding in this paper will be with much less encoding complexity.

Then we compare the proposed local 3D padding algorithm with the camera-parameter based global padding method [14]. The experimental results for the CMP format under the RA10 case are shown in Table X. Since the proposed method can better reflect the essence of the projections, the proposed method can lead to about 0.4% bitrate savings compared with [14]. However, we need to emphasize that the performance improvement is not the benefits of the local 3D padding method compared with the global padding method. The benefits come from the fact that the proposed padding model used is better than the padding model in [14]. It should also be mentioned that [14] has the advantage of extending to various polyhedron formats very easily.

We finally compare the proposed local 3D padding with [15] and [16]. The essences of the proposed algorithm and the algorithms proposed in [15] and [16] are the same. Therefore, the performance improvements for all of them are quite similar. From the implementation point of view, the proposed algorithm will be with higher encoding and decoding complexity since the proposed algorithm needs to replace the original reference pixels and then get them back locally. However, just because of this, the proposed algorithm will have fewer memory requirements. The proposed algorithm presents different trade-offs between the memory requirements...
TABLE XI
THE PERFORMANCE OF THE COMBINATION OF THE LOCAL 3D PADDING AND ADVANCED SPHERICAL MOTION MODEL ON THE CTC TEST SEQUENCES

<table>
<thead>
<tr>
<th>Test sequence</th>
<th>ERP</th>
<th>CMP</th>
<th>OHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>RA10</td>
<td>0.6%</td>
<td>0.4%</td>
</tr>
<tr>
<td>SkateboardingTrick</td>
<td>0.1%</td>
<td>0.1%</td>
<td>-0.1%</td>
</tr>
<tr>
<td>SkateboardingInLot</td>
<td>-2.6%</td>
<td>-2.7%</td>
<td>-3.0%</td>
</tr>
<tr>
<td>Chairlift</td>
<td>-8.3%</td>
<td>-6.0%</td>
<td>-3.3%</td>
</tr>
<tr>
<td>KiteFlite</td>
<td>0.0%</td>
<td>0.1%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Harbor</td>
<td>0.1%</td>
<td>0.3%</td>
<td>0.1%</td>
</tr>
<tr>
<td>PoleVault</td>
<td>0.0%</td>
<td>0.0%</td>
<td>-0.1%</td>
</tr>
<tr>
<td>AerialCity</td>
<td>-0.9%</td>
<td>-1.4%</td>
<td>-2.0%</td>
</tr>
<tr>
<td>DrivingInCity</td>
<td>-0.4%</td>
<td>-0.2%</td>
<td>-1.4%</td>
</tr>
<tr>
<td>DrivingInCountry</td>
<td>-6.0%</td>
<td>-3.6%</td>
<td>-6.4%</td>
</tr>
<tr>
<td>Avg</td>
<td>-1.9%</td>
<td>-1.4%</td>
<td>-1.7%</td>
</tr>
</tbody>
</table>

TABLE XII
THE PERFORMANCE OF THE COMBINATION OF THE LOCAL 3D PADDING AND ADVANCED SPHERICAL MOTION MODEL ON THE SEQUENCES WITH HIGHER FRAME RATE

<table>
<thead>
<tr>
<th>Test sequence</th>
<th>Frame rate (fps)</th>
<th>ERP</th>
<th>CMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balboa</td>
<td>60</td>
<td>-2.2%</td>
<td>-1.3%</td>
</tr>
<tr>
<td>Broadway</td>
<td>60</td>
<td>-0.9%</td>
<td>-0.4%</td>
</tr>
</tbody>
</table>

G. The Performance Comparison of Various Projections

Table XIII shows the performance comparison among various projections using ERP as the anchor. From Table XIII, we can see that the CMP format achieves about 0.8% R-D performance loss compared with the ERP format in both RA10 and LD10 cases. The ERP format achieves better R-D performance for the static sequences such as Train and SkateboardingTrick, while the CMP format achieves better R-D performance for the sequences with relatively large motions such as DrivingInCountry. The geometry distortions may have larger influences on the inter correlations for the sequences with large motions under the ERP format, while almost have no influence on the static sequences. That can partially explain why the performance highly depends on the contents of the sequences. For the OHP format, it suffers about 8.0% performance loss compared with the ERP format. The OHP format is unable to be encoded efficiently by the standard-based coding framework since only square or rectangle partitions are supported.

H. The Subjective Quality of the Proposed Algorithm

The proposed framework can not only improve the objective quality but also the subjective quality. Since the viewers usually watch the 360-degree video from a specified view, the images shown in the following are the scenes with size 400 × 400 cropped from the view port 1 defined in the CTC. We give two examples in ERP and CMP formats to explain benefits brought by the advanced spherical motion model and the local 3D padding.

The subjective quality on Frame 16 of the sequence DrivingInCountry under ERP format tested in RA10 case using QP 37 is shown in Fig. 16. From Fig. 16, we can see that the advanced spherical motion model can keep more details under even lower bitrate. The subjective quality on Frame 3 of the sequence DrivingInCountry under CMP format tested in LD10 case using QP 37 is shown in Fig. 17. From Fig. 17, we can obviously see that the local 3D padding method can keep the texture more continuous compared with the HEVC anchor. The experimental results obviously demonstrate that the proposed framework can provide better subjective quality.

I. The Complexity and Memory Analysis of the Proposed Algorithms

The encoding and decoding time increases of the proposed algorithms are shown in Table XIV. From Table XIV, we can see that the proposed advanced spherical motion model will bring over 3 times encoding complexity compared with the HEVC anchor. The decoding time increase is around 20%.
TABLE XIV
THE AVERAGE ENCODING AND DECODING TIME INCREASE OF THE PROPOSED ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Projection format</th>
<th>Test case</th>
<th>Encoding time</th>
<th>Decoding time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced motion model</td>
<td>ERP</td>
<td>LD10</td>
<td>372%</td>
<td>116%</td>
</tr>
<tr>
<td></td>
<td>CMP</td>
<td>LD10</td>
<td>363%</td>
<td>116%</td>
</tr>
<tr>
<td></td>
<td>OHP</td>
<td>LD10</td>
<td>291%</td>
<td>115%</td>
</tr>
<tr>
<td>Local 3-D padding</td>
<td>ERP</td>
<td>RA10</td>
<td>98%</td>
<td>105%</td>
</tr>
<tr>
<td></td>
<td>CMP</td>
<td>RA10</td>
<td>251%</td>
<td>116%</td>
</tr>
<tr>
<td></td>
<td>OHP</td>
<td>RA10</td>
<td>584%</td>
<td>168%</td>
</tr>
<tr>
<td>Combined algorithm</td>
<td>ERP</td>
<td>LD10</td>
<td>371%</td>
<td>118%</td>
</tr>
<tr>
<td></td>
<td>CMP</td>
<td>RA10</td>
<td>857%</td>
<td>205%</td>
</tr>
<tr>
<td></td>
<td>OHP</td>
<td>LD10</td>
<td>1510%</td>
<td>230%</td>
</tr>
</tbody>
</table>

complexity increase mainly comes from the projections from sphere to the 2D projections and the inverse process. We will consider reducing the complexity in our future work. For the local 3D padding method, it will not lead to obvious encoding time and decoding time increase for the ERP format since only the picture level padding is used. For the CMP format, the local 3D padding method will bring about 2 to 2.5 times encoding time increase and 15% to 20% decoding time increase. For the OHP format, the complexity is even higher due to the fact that there are more face boundaries to handle. The complexity of the proposed algorithm mainly comes from the local 3D padding operations recursively. The combination of the two algorithms will bring about the multiplication of complexity increase for the algorithms individually for both the encoder and decoder.

For the memory usage, as we have analyzed above, the proposed local 3D padding algorithm will not increase the memory requirement. Since the advanced spherical motion model also does not need any extra memory, the proposed combined algorithm will not lead to any memory usage increase.

VII. CONCLUSION

In this paper, to deal with the geometry distortion and the face boundary discontinuity in different projection formats of the 360-degree video, an integrated framework is developed to handle these two distortions so as to improve coding efficiency. The proposed framework mainly has two key contributions. First, we derive a unified advanced spherical motion model to handle the geometry distortion of different projection formats for the 360-degree video. Second, we propose a local 3D padding method to handle the face boundary discontinuity between the neighboring faces in various projection formats of the 360-degree video. These two methods can be combined into an integrated framework to achieve a better rate-distortion performance. The proposed algorithms are implemented in the High Efficiency Video Coding (HEVC) reference software. The experimental results show that the proposed algorithm can achieve an average of over 4% bitrate savings compared with the HEVC anchor for the test sequences with some motions. The experiments obviously demonstrate the effectiveness of the proposed methods.

REFERENCES

Li Li (M’17) received the B.S. and Ph.D. degrees in electronic engineering from the University of Science and Technology of China, Hefei, China, in 2011 and 2016, respectively. He is currently a Post-Doctoral Researcher with the University of Missouri-Kansas City. His research interests include image/video coding and processing. He received the Best 10% Paper Award at the 2016 IEEE Visual Communications and Image Processing Conference.

Zhu Li (M’01–SM’07) received the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, in 2004. He is currently an Associate Professor with the Department of Computer Science and Electrical Engineering (CSEE), University of Missouri, Kansas City, and Director of the NSF I/UCRC Center for Big Learning (CBL) at UMKC. He was summer visiting faculty at the US Air Force Academy (USFA), 2016, 2017, and 2018 with the UAV Research Center. He was Senior Staff Researcher/Senior Manager with Samsung Research America’s Multimedia Standards Research Lab in Richardson, TX, USA, Senior Staff Researcher/Media Analytics Lead with FutureWei (Huawei) Technology’s Media Lab in Bridgewater, NJ, USA, and an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University from 2008 to 2010, and a Principal Staff Research Engineer with the Multimedia Research Lab (MRL), Motorola Labs, from 2000 to 2008. His research interests include point cloud and light field compression, graph signal processing and deep learning in the next generation visual compression, image/video analysis and understanding. He has 46 issued or pending patents, 100+ publications in book chapters, journals, and conferences in these areas. He has been Associate Editor for IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, and the Journal of Visual Signal Processing Systems (Springer), since 2015. He serves as a steering committee member of IEEE ICME (2015–), he is an elected member of the IEEE Multimedia Signal Processing (MSSP), IEEE Image, Video, and Multidimensional Signal Processing (IVMSP), and IEEE Visual Signal Processing and Communication (VSPC) Technical Committees. He is program co-chair for IEEE International Conference on Multimedia and Expo (ICME) 2019, and co-chaired IEEE Visual Communication and Image Processing (VCIP) 2017. He received the Best Paper Award at IEEE International Conference on Multimedia and Expo (ICME), Toronto, 2006, the Best Paper Award (DocCoMo Labs Innovative Paper) at IEEE International Conference on Image Processing (ICIP), San Antonio, 2007.

Xiang Ma received the B.S. and Ph.D. degrees in communication engineering from Xidian University, Xi’an, China, in 2009 and 2015, respectively. In 2015, he joined Huawei Technologies, Shenzhen, China, where he is actively contributing to video compression standardization activities. He is currently a Senior Engineer with the Media Technology Laboratory, Corporate Research Department, Huawei. His current research interests include video compression and processing algorithms for video communication.

Haitao Yang received the B.S. and Ph.D. degrees in electronic engineering from Xidian University, Xi’an, China, in 2004 and 2009, respectively. From 2009 to 2010, he was a Post-Doctoral Research Associate with The Hong Kong University of Science and Technology, where he focused on video compression. In 2010, he joined Huawei Technologies, Shenzhen, China, where he is actively participating and contributing to video compression standardization activities conducted by the ITU-T Video Coding Experts Group and ISO/IEC Moving Pictures Experts Group, including the Joint Collaborative Team on Video Coding and Joint Video Exploration Team activities. He is currently a Principle Engineer with the Media Technology Laboratory, Corporate Research Department, Huawei. His current research interests include video compression and processing algorithms for video communication.