CONTENT-ADAPTIVE 360-DEGREE VIDEO CODING USING HYBRID CUBEMAP PROJECTION

Yuwen He, Xiaoyu Xiu, Philippe Hanhart, Yan Ye
InterDigital Communications
9710 Scranton Road
San Diego, CA 92121, USA
{yuwen.he, xiaoyu.xiu, philippe.hanhart,
yan.ye}@interdigital.com

Fanyi Duanmu, Yao Wang
New York University
2 Metro Tech Center
Brooklyn, NY 11201, USA
{fanyi.duanmu, yaowang}@nyu.edu

***Abstract*—In this paper, a novel hybrid cubemap projection (HCP) is proposed to improve the 360-degree video coding efficiency. HCP allows adaptive sampling adjustments in the horizontal and vertical directions within each cube face. HCP parameters of each cube face can be adjusted based on the input 360-degree video content characteristics for a better sampling efficiency. The HCP parameters can be updated periodically to adapt to temporal content variation. An efficient HCP parameter estimation algorithm is proposed to reduce the computational complexity of parameter estimation. Experimental results demonstrate that HCP format achieves on average luma (Y) BD-rate reduction of 11.51%, 8.0%, and 0.54% compared to equirectangular projection format, cubemap projection format, and adjusted cubemap projection format, respectively, in terms of end-to-end WS-PSNR.**

I. INTRODUCTION

360-degree video has become popular in recent years with the advances in virtual reality (VR) and augmented reality (AR) technologies and has been rapidly commercialized in a variety of applications, such as immersive cinema, gaming, education/training, healthcare, social media and 360-degree video streaming, etc. To provide users with an immersive experience, 360-degree video requires much higher bandwidth compared with conventional 2D video, due to the increase in resolution, frame rate, and quality of experience requirements. For example, a premium quality 360-degree stereo video with 90 frames per second (fps) at 8K resolution can easily consume bandwidth up to multiple gigabits per second (Gbps). Therefore, efficient compression and delivery of ultra-high quality 360-degree video becomes important for the wide adoption of VR/AR applications.

Many companies are working towards developing more efficient 360-degree video compression and delivery systems and products. Some preliminary 360-degree video services are already provided on several major video platforms, such as Facebook and YouTube. The joint video exploration team (JVET) from MPEG and VCEG, which is exploring new technologies for the next generation video coding standard, is investigating 360-degree video coding technologies [1]. The joint exploration model (JEM) software maintained by JVET is used as the codebase for the exploration work of 360-degree video coding [2]. Recently, a call for proposals (CfP) [3] was issued on next generation video compression technologies beyond HEVC, where 360-degree video is included as an important content category.

The typical workflow of 360-degree video compression and delivery is illustrated in Figure 1. Firstly, video sequences from multiple cameras are captured and stitched into a native projection format, e.g., the equirectangular projection (ERP) format. The native projection format can be converted into another projection format, e.g., the cubemap (CMP), and frame packed before being fed into existing video codecs, such as H.264, HEVC, VP9, etc. At the client side, the decompressed video is converted into the projection format supported by the display, followed by the graphical rendering via viewport generation based on user's viewing direction before finally being projected onto the display devices (e.g., head mounted display, smartphone, desktop, etc.).

In this framework, the selection of an intermediate 360-degree video projection format is important and would potentially improve the coding performance. In the previous JVET meetings, many projection formats have been proposed in addition to ERP and CMP, and several are being investigated, such as icosahedral projection (ISP) [4], segmented sphere projection (SSP) [5], octahedron projection (OHP) [6], etc.

Among all those projection formats, CMP is widely used in the computer graphics community. Due to the intrinsic rectilinear structure of CMP, the resulting motion field (which describes the temporal correspondence between neighboring 2D projected pictures) can be efficiently represented by the translational motion model used in modern video codecs (e.g., H.264, HEVC, etc.).

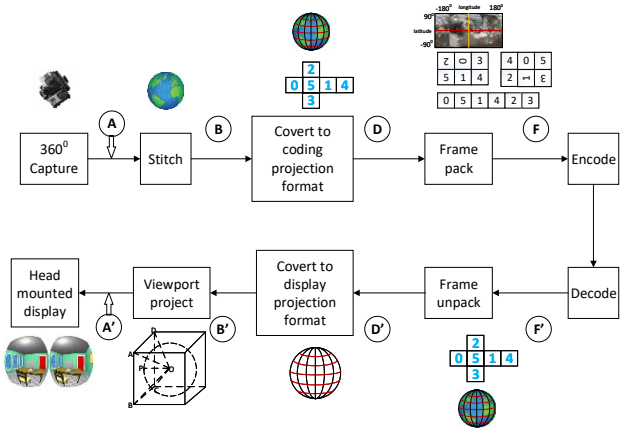


Figure 1. Workflow of 360-degree video system.

However, due to the limitation of the rectilinear projection, the samples on the sphere are unevenly sampled in the CMP format, with a higher sampling density near the face boundaries and a lower sampling density near the face centers. Such non-uniform spherical sampling could penalize the coding efficiency of the resulting 360-degree video. To achieve a better sampling efficiency for any input spherical video, it is desirable to have a projection format that allows flexible sampling function selection to accommodate different video content characteristics. Therefore, in this work, a hybrid cubemap projection (HCP) format is proposed to generalize CMP and other CMP-like projections, and allow adaptive selections of optimal transform functions for each direction over each cube face based on the specific content characteristics in that face.

The remainder of the paper is structured as follows. Section II presents related work. Section III introduces the HCP algorithm design, parameter derivation process, and parameter signaling. In Section IV, a progressive HCP parameter search algorithm is proposed to reduce the temporal variation of HCP parameters and the computational complexity. In Section V, experimental results are provided. Section VI concludes this paper with future work summarized.

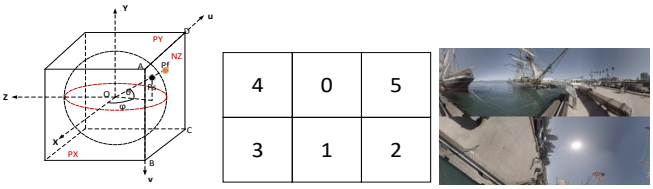
II. Related Work

Due to its simplicity and computational efficiency, CMP is widely supported by graphics hardware and video editing and processing software. As illustrated in Figure 2 (a), the CMP consists of six square faces labeled using numbers 0-5, where X, Y, and Z refers to the axes. Suppose the radius of the tangent sphere is 1, then the lateral length of each face is 2. Before being fed into a conventional video codec, the six CMP faces are frame-packed together into a single rectangular picture. Additionally, to maximize the continuity between neighboring faces, some faces may be rotated by a certain degree. Figure 2 (b) shows one frame packing scheme that places the six faces into a rectangular picture using 3×2 layout. Figure 2 (c) provides an exemplary picture.

To address the non-uniform sampling problem of CMP, some CMP-like projections have been proposed, such as adjusted cubemap projection (ACP) [7]. ACP improves the spherical sampling uniformity by adjusting the coordinates in the cube faces using transform functions whose coefficients are tailored to approximate uniform sphere sampling. For example, in ACP, a pair of transform and inverse transform functions are used, as shown in Eq. (1) and (2), respectively, where x and x' are the original coordinate defined in CMP domain and the adjusted coordinate defined in ACP domain, respectively, and $sgn(\cdot)$ is the function that returns the sign of the input value. The approximately uniform sampling grid on the sphere covering one CMP face is shown in Figure 3 (a). The corresponding sampling grid in CMP domain projected from approximately uniform sampling grid on the sphere with 3D to 2D projection is shown in Figure 3 (b). The ACP sampling grid is shown in Figure 3 (c). As can be seen, ACP greatly improves sampling uniformity compared to CMP.

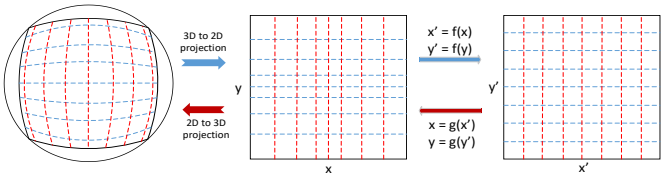
$$x' = f(x) = sgn(x) \cdot (-0.36 \cdot x^2 + 1.36 \cdot |x|) \quad (1)$$

$$x = g(x') = sgn(x') \cdot \frac{0.34 - \sqrt{0.34^2 - 0.09 \cdot |x'|}}{0.18} \quad (2)$$



(a) (b) (c)

Figure 2. CMP example. (a) 3D geometry structure; (b) the 2D layout for six faces; (c) packed CMP frame.



(a) (b) (c)

Figure 3. Illustration of samping grids in different domains. (a) Uniform sampling grid on the sphere; (b) CMP sampling grid projected from approximately uniform sampling grid; (c) ACP sampling grid after applying transform functions.

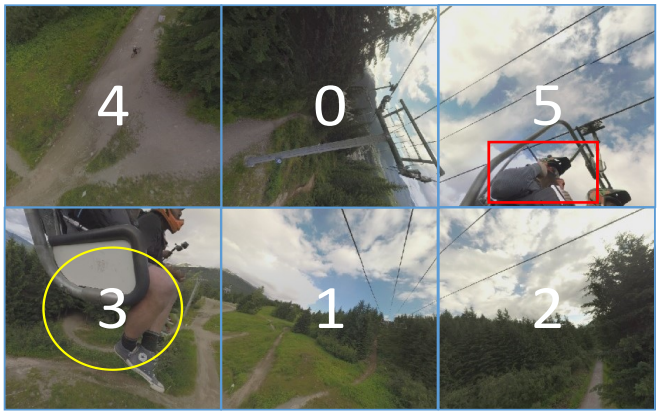


Figure 4. Packed CMP sample frame from 360-degree video sequence "ChairliftRide".

Although ACP can improve the sampling uniformity, it can be still further improved:

Firstly, as shown in Eq. (1) and (2), the sampling adjustment functions of ACP are fixed for the horizontal and vertical directions, regardless of the content inside each cube face. As shown in Figure 4, the most "attractive" content having complex texture over face 5 (as enclosed by the red square) is located at the bottom boundary of the face whereas the remaining part of the face is composed of regions with relatively simple texture. In such a case, it is preferable to design one transform function that assigns a higher spherical sampling density near the face boundaries and a lower spherical sampling density near the face center. On the contrary, in face 3, the most "attractive" content (as enclosed by the yellow circle) is located in the center of the face. Therefore, the coding performance can be improved by applying a transform function that assigns a lower spherical sampling density at the face boundaries and a higher spherical sampling density at the face center.

Secondly, the derivations of transform functions from Eq. (1) and (2) assume that each ACP face has a symmetric sampling pattern between the horizontal and vertical directions, just as CMP does. Such symmetry property may not always be optimal for each ACP face. As shown in Figure 4, the content in face 1 presents stable characteristics in the horizontal direction corresponding to either the sky region or the ground region. However, when moving along the vertical direction, the top portion corresponds to the sky region with relatively simple texture and the bottom portion corresponds to the ground region with fine texture and edges. This indicates that the content characteristics change along the vertical direction. In this case, uniform sampling in face 1 may not be optimal. Instead, a non-uniform spherical sampling density in the vertical direction (i.e., gradually increasing the sampling density from the top to the bottom of the face) with a relatively uniform sampling density in the horizontal direction would likely perform better.

III. HYBRID CUBEMAP PROJECTION

As introduced in Section II, ACP can improve the 360-degree video coding efficiency compared with the conventional CMP projection. However, the sampling distribution and transform functions of ACP are fixed. In this work, the HCP format is proposed to generalize CMP and CMP-like projections, and to better adapt to diverse content over each cube face. Similar to ACP, HCP is also related to CMP with a pair of transform and inverse transform functions. Assuming (x, y) is defined in a CMP face with side length of 2, and (x', y') is the corresponding point in an HCP face, the transform functions from CMP to HCP are defined in Eq. (3) and (4), where a and b are the HCP horizontal and vertical transform function parameters, respectively.

$$x' = f_X(x) = sgn(x) \cdot (a \cdot x^2 + (1 - a) \cdot |x|) \quad (3)$$

$$y' = f_Y(y) = sgn(y) \cdot (b \cdot y^2 + (1 - b) \cdot |y|) \quad (4)$$

Correspondingly, the inverse transform functions from HCP to CMP are defined in Eq. (5) and (6). If the parameter a or b is equal to zero, the sampling distribution in that corresponding direction becomes the same as CMP.

$$x = g_X(x') = sgn(x') \cdot \frac{-(1 - a) + \sqrt{(1 - a)^2 + 4a \cdot |x'|}}{2a} \quad (5)$$

$$y = g_Y(y') = sgn(y') \cdot \frac{-(1 - b) + \sqrt{(1 - b)^2 + 4b \cdot |y'|}}{2b} \quad (6)$$

As shown in Figure 4, the three faces within a cube face row (either top or bottom) are geometric neighbors in the 3D space. To maintain such face boundary continuity in between two neighbouring faces, the following constraints are applied over HCP parameters, as defined in Eq. (7) and (8), where b_4, b_0, and b_5 denote the vertical parameters of face 4, 0, and 5, respectively, and b_2, b_1, and b_3 denote the vertical parameters of face 2, 1, and 3, respectively. Namely, faces F_4, F_0, and F_5, which belong to the first row, will share the same vertical transform function, and faces F_2, F_1, and F_3, which belong to the second row, will share the same vertical transform function.

$$b_4 = b_0 = b_5 \quad (7)$$

$$b_2 = b_1 = b_3 \quad (8)$$

HCP parameters are selected to minimize the end-to-end weighted conversion-only error, as D calculated in Eq. (9), where the weight $W(i, m, n)$ is the same as those defined in weighted to spherically uniform PSNR (WS-PSNR) for ERP format in 360Lib [2][8]. Here F_i denotes the portion of the source ERP corresponding to face F_i in the HCP, and F_i' denotes the portion of the reconstructed ERP corresponding to face F_i in the HCP.

$$D = \sum_{F_i \in face_{row}} \sum_{(m,n) \in F_i} W(i, m, n) \, (F_i[m, n] - F_i[m, n])^2 \quad (9)$$

Considering the constraints in Eq. (7) and (8), the HCP parameter derivation process for the 3×2 frame packing scheme as shown in Figure 2 (b) is separated into two sub-processes: (a) deriving parameters for the first face row consisting of F_4, F_0, and F_5; and (b) deriving parameters for the second face row consisting of F_2, F_1, and F_3. For each face row, the following iterative parameters searching algorithm is applied over horizontal and vertical directions:

Step 1: Search parameters for the horizontal direction while fixing the parameters for the vertical direction, and update the parameters for the horizontal direction with the optimal parameters found in searching; if there is no update, searching process stops; otherwise, go to Step 2;

Step 2: Search parameters for the vertical direction while fixing parameters for the horizontal direction, and update the parameters for the vertical direction with the optimal parameters found in searching; if there is no update, searching process stops; otherwise, go to Step 1.

Usually, the proposed HCP parameter search algorithm converges within three to four iterations. The final parameters will be signalled at picture level. The parameters are estimated and updated periodically using the first picture of each intra random-access point (IRAP) and signalled once per IRAP. To support the proposed HCP design, the high-level syntax is modified. Both sequence parameter set (SPS) and picture parameter set (PPS) are extended and signalled at the beginning of each IRAP, as follows.

SPS: projection format, frame packing parameters, e.g., number of faces in horizontal and vertical directions in the frame packed picture, and each face's position and orientation in the frame packed picture.

PPS: the horizontal and vertical HCP parameters.

IV. HCP PROGRESSIVE PARAMETER SEARCH

As discussed in Section III, HCP parameter search process is computationally-expensive due to the large number of search parameter candidates (e.g., up to 64 under 6-bit precision) and iterations. Therefore, to reduce the complexity over the target I-frame of each IRAP, a progressive HCP parameter search algorithm is proposed to accelerate the search process, while simultaneously preserving the encoding and conversion efficiency. There are two major aspects to reduce the complexity: (a) reducing the search candidates with a fast search at the IRAP pictures of the video sequence; (b) refining the parameter within a limited window for the subsequent IRAP pictures when HCP parameters are periodically updated.

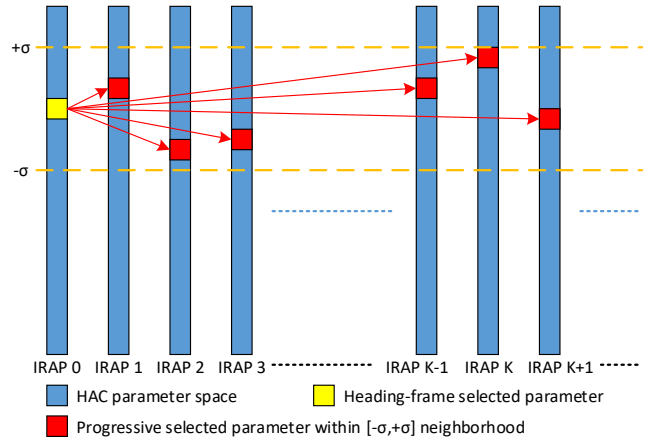
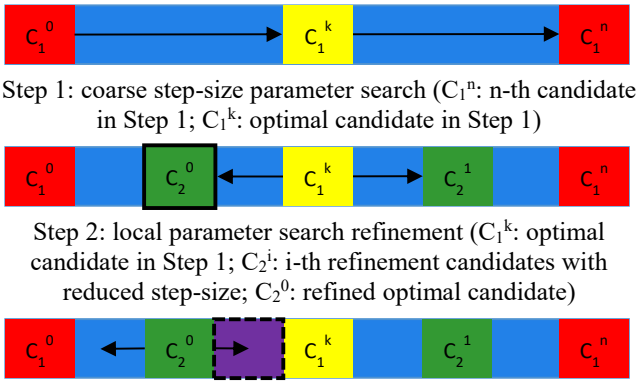


Figure 5. Illustration of progressive HCP parameter search.



Step 1: coarse step-size parameter search (C_1^n: n-th candidate in Step 1; C_1^k: optimal candidate in Step 1)

Step 2: local parameter search refinement (C_1^k: optimal candidate in Step 1; C_2^i: i-th refinement candidates with reduced step-size; C_2^0: refined optimal candidate)

Step k: recursive local search refinement until step-size reduces to 1 (purple box: final optimal candidate via CFS)

Figure 6. Illustration of coarse-to-fine search. The step-size in Step 1, Step 2, and Step 3 is set to 4, 2, and 1, respectively.

As shown in Figure 5, the first frame of the 360-degree video is defined and hereafter referred to as the "heading-fame" (f_H). Denote the HCP parameters derived for f_H as P_i^h, where "h" stands for heading-frame and i is the face index, ranging from 0 to 5. Once P_i^h is determined during encoding or conversion, the HCP parameters over the following IRAPs are only searched within a small neighborhood, i.e., $[-\sigma, +\sigma]$, with respect to P_i^h, where σ defines the progressive search range (PSR) and can be determined either empirically or based on the video characteristics, e.g., through pre-processing.

In our implementation, the parameter precision is 6 bits. Therefore, the integer parameter search range is within [-63, 0], corresponding to HCP parameter range within (-1, 0]. Note that CMP and ACP cases are covered within our search range setting. Our PSR is set to 4. The HCP search range for the subsequent IRAPs is much smaller compared to the search range of the first IRAP, and therefore significantly reduces the computational complexity.

To further improve the search efficiency, the parameter search over all IRAP frames can be further optimized, using fast algorithms such as coarse-to-fine search (CFS). CFS achieves a significant speedup, as illustrated in Figure 6. CFS algorithm firstly searches over the entire parameter space with a uniform coarse search step-size. When the initial optimal parameter is pinpointed, a local refinement is triggered in a smaller neighborhood of the previously-chosen optimal parameter and the search step-size is reduced by half. The local search refinement only searches over two neighbors next to the optimal parameter in the previous step. Such refinement process continues until a convergence criterion is met (e.g., the step-size reduces to 1).

In practice, the 360-degree video encoder or converter can trade off the complexity and quality performance by different configurations of search range and search step-size. For example, the HCP parameter progressive search range can be adaptively configured according to the distance between the current picture and the heading-frame.

V. Performance Evaluation

In this section, detailed experimental results are provided to demonstrate the coding performance and HCP parameter estimation complexity. The proposed HCP format and HCP parameter estimation algorithms are implemented upon JEM-6.0 [9] and 360Lib-3.0 [8]. The simulation results are reported using the test conditions specified in the call for evidence (CfE) document [10]. The size of each HCP face is set to 1184×1184 luma samples, which is the same face size used for ACP in the 360-degree video common test conditions (CTC) [1]. Table 1 lists those test sequences and the original format is ERP. Random access configuration is used for the encoding and HCP parameters are updated per IRAP in the simulations. The 3×2 frame packing is used for ACP and HCP as shown in Figure 4, and the coded picture resolution is 3552×2368. The end-to-end WS-PSNR (E2E WS-PSNR) [2] is applied in the objective quality comparison, where the decoded video is firstly converted back to the ERP format of the resolution as original source, then WS-PSNR is calculated in ERP format between the reference and the test.

The conversion-only performance without compression is provided in Table 2. The original ERP video is firstly converted to a projection format in a 25% downsized ratio, then the downsized video is converted back to original resolution ERP and the E2E WS-PSNR is calculated. For ERP format, the resolution of converted video is 4096×2048. The proposed HCP format outperforms ERP by 3.01dB E2E WS-PSNR improvement for luma and outperforms ACP by 0.16dB E2E WS-PSNR improvement for luma. This validates spherical sampling efficiency of the proposed HCP over both ERP and ACP.

Table 1. Test Sequences.

Sequence	ERP resolution	Frame count	Frame rate	Bit depth
SkateboardInLot	8192×4096	300	30	10
ChairliftRide	8192×4096	300	30	10
KiteFlite	8192×4096	300	30	8
Harbor	8192×4096	300	30	8
Trolley	8192×4096	300	30	8

Table 2. HCP conversion-only performance against ERP/ACP.

Sequence	E2E WS-PSNR improvement over ERP (dB)			E2E WS-PSNR improvement over ACP (dB)		
	Y	U	V	Y	U	V
SkateboardInLot	3.64	1.45	1.18	0.40	0.06	0.10
ChairliftRide	2.67	0.96	0.91	0.06	0.02	0.10
KiteFlite	2.81	0.75	0.78	0.03	-0.01	0.10
Harbor	3.11	0.76	0.78	0.23	0.02	0.10
Trolley	2.82	0.64	0.68	0.10	-0.01	0.10
Average	3.01	0.91	0.87	0.16	0.01	0.10

Table 3. HCP coding performance compared with ERP/ACP.

Sequence	BD rate saving relative to ERP			BD rate saving relative to ACP		
	Y	U	V	Y	U	V
SkateboardInLot	-15.25%	-22.94%	-21.45%	-0.90%	-0.90%	0.70%
ChairliftRide	-23.00%	-21.87%	-21.10%	-1.0%	-4.10%	-3.40%
KiteFlite	-6.05%	-8.80%	-12.76%	0.0%	0.10%	0.70%
Harbor	-7.21%	-12.31%	-12.80%	-0.3%	-0.50%	-1.90%
Trolley	-6.07%	-7.37%	-13.99%	-0.5%	-1.40%	-2.20%
Average	-11.51%	-14.66%	-16.42%	-0.54%	-1.37%	-1.22%

Table 4. HCP with progressive parameter search compared to HCP with exhaustive search and no temporal constraint.

Sequence	BD rate saving		
	Y	U	V
SkateboardInLot	-0.4%	-0.2%	0.2%
ChairliftRide	-0.1%	-0.2%	-0.3%
KiteFlite	0.0%	-0.3%	0.2%
Harbor	0.0%	0.1%	-0.2%
Trolley	-0.2%	-6.1%	-9.9%
Average	-0.13%	-1.35%	-2.01%

Table 5. HCP parameter estimation complexity comparison.

Sequence	ACP		HCP exhaustive search		HCP progressive search (PSR=4)	
	Y (dB)	Time (s)	Y (dB)	Time (s)	Y (dB)	Time (s)
SkateboardInLot	49.74	1890	50.14	24908	50.13	4208
ChairliftRide	50.00	1747	50.06	25197	50.03	4425
KiteFlite	46.41	1862	46.44	23254	46.44	4005
Harbor	48.99	1884	49.22	26491	49.22	5147
Trolley	45.29	1864	45.39	24780	45.39	4992
Average	48.09	1850	48.25	24926	48.24	4555

The BD rate [11] performance of HCP with progressive parameter search compared with ERP and ACP coding is provided in Table 3. From Table 3, the proposed HCP format demonstrates on average luma BD rate reduction of 11.51% compared to ERP format in terms of end-to-end WS-PSNR. The HCP format also achieves a luma BD rate reduction of 0.54% compared with the ACP format. Compared to CMP, the HCP achieves a luma BD rate reduction of 8.0%.

Additionally, the coding performance and parameter estimation complexity are evaluated between the progressive parameter search and exhaustive parameter search algorithms for HCP. The exhaustive parameter search method iteratively searches for the optimal parameter with a fixed search step-size of 1 (as discussed in Section III), and temporal continuity constraint for HCP parameters is not applied. Table 4 compares the coding performance of HCP using progressive parameter search with that using exhaustive search. On average, progressive search achieves 0.13% BD rate reduction and the gain is larger over fast-motion 360-degree video sequences such as "SkateboardInLot" and "ChairliftRide".

The parameter estimation complexity is evaluated with conversion-only test. Table 5 lists the quality and conversion time for ACP, HCP with exhaustive search, and HCP with progressive search. ACP is used as an anchor because there is no parameter estimation needed. HCP with progressive search is 5.5x faster than exhaustive search with practically the same conversion quality (i.e., only 0.01dB loss). Note that although HCP with progressive search is 2.5x slower than ACP during the parameter estimation stage, HCP parameter search introduces negligible complexity overhead compared to the overall encoding complexity and is only executed once per IRAP.

VI. CONCLUSION

In this work, a novel hybrid cubemap projection (HCP) solution is proposed to compress 360-degree video more efficiently. By flexibly selecting the transform functions for each cube face, more efficient sampling is achieved according to the 360-degree video content characteristics and coding performance is therefore improved. Future extension of this work may include HCP parameter estimation considering the distortion introduced during the coding process.

REFERENCES

[1] J. Boyce, E. Alshina, A. Abbas, Y. Ye, "JVET common test conditions and evaluation procedures for 360° video," JVET-F1030, Apr. 2017.

[2] Y. Ye, E. Alshina, J. Boyce, "Algorithm descriptions of projection format conversion and video quality metrics in 360Lib Version 3," JVET-F1003, Apr. 2017.

[3] A. Segall, V. Baroncini, J. Boyce, J. Chen, T. Suzuki, "Joint Call for Proposals on Video Compression with Capability beyond HEVC," JVET-H1002, Oct. 2017.

[4] V. Zakharchenko, E. Alshina, K. P. Choi, A. Singh, A. Dsouza, "AhG8: Icosahedral projection for 360-degree video content," JVET-D0028, Oct. 2016.

[5] C. Zhang, Y. Lu, J. Li, Z. Wen, "Segmented sphere projection (SSP) for 360-degree video content," JVET-D0030, Oct. 2016.

[6] H.-C. Lin, C.-Y. Li, J.-L. Lin, S.-K. Chang, C.-C. Ju, "An efficient compact layout for octahedron format," JVET-D0142, Oct. 2016.

[7] M. Coban, G. Van der Auwera, M. Karczewicz, "AHG8: Adjusted cubemap projection for 360-degree video," JVET-F0025, Apr. 2017.

[8] 360Lib-3.0
https://jvet.hhi.fraunhofer.de/svn/svn_360Lib/tags/360Lib-3.0

[9] JEM-6.0
https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-6.0

[10] M. Wien, V. Baroncini, J. Boyce, A. Segall, T. Suzuki, "Joint Call for Evidence on Video Compression with Capability beyond HEVC," JVET-F1002, Apr. 2017.

[11] G. Bjontegaard, "Improvements of the BD-PSNR model," ITU-T SG16 Q6, document VCEG-AI11, July 2008.