Pacific Graphics 2019
C. Theobalt, J. Lee, and G. Wetzstein
(Guest Editors)

Volume 38 (2019), Number 7

A Generalized Cubemap for Encoding 360° VR Videos using Polynomial Approximation

J. Y. Xiao, J. T. Tang and X. Y. Zhang†

School of Software Engineering, East China Normal University, China.
xyzhang@sei.ecnu.edu.cn
http://www.robotics.sei.ecnu.edu.cn

Abstract

360° VR videos provide users with an immersive visual experience. To encode 360° VR videos, spherical pixels must be mapped onto a two-dimensional domain to take advantage of the existing video encoding and storage standards. In VR industry, standard cubemap projection is the most widely used projection method for encoding 360° VR videos. However, it exhibits pixel density variation at different regions due to projection distortion. We present a generalized algorithm to improve the efficiency of cubemap projection using polynomial approximation. In our algorithm, standard cubemap projection can be regarded as a special form with 1st-order polynomial. Our experiments show that the generalized cubemap projection can significantly reduce the projection distortion using higher order polynomials. As a result, pixel distribution can be well balanced in the resulting 360° VR videos. We use PSNR, S-PSNR and CPP-PSNR to evaluate the visual quality and the experimental results demonstrate promising performance improvement against standard cubemap projection and Google's equi-angular cubemap.

CCS Concepts

• ***Human-centered computing*** → ***Virtual reality;*** • ***Computing methodologies*** → ***Virtual reality;*** *Image representations;*

1. Introduction

360° VR videos record a surrounding environment in every direction at the same time and enable viewers to interactively look around in the captured environment. Compared to computer-modelled 3D content, 360° VR videos can capture immersive content very quickly. Due to this unique feature, it opens a new world of potential applications such as VR education/training [HCC*17], VR games [LHLK17], VR films [BYS*17], VR museum/tour [AGB*16] and VR sports broadcasting [XZLD18]. Thanks to economic VR headsets and VR cameras launched in the past few years along with fast evolution of mobile phone and rising advancement of 5G technology, the application of 360° VR videos is growing exponentially as experience and marketing tools. The technological development for 360° VR cameras, editing, streaming and display equipment has made huge leaps.

360° VR videos face a few new challenges: 1) streaming high quality video through mobile networks will become ubiquitous and the bandwidth demands are vastly increased compared with conventional videos. 2) The 360° VR video storing and editing process is in a way similar, but also very different from conventional videos. 3) Every viewer has independent control of the viewing direction. All these challenges make efficient encoding of 360° VR video one of the top concerns. To encode 360° VR videos, spherical pixels must be mapped onto a two-dimensional domain to take advantage of the existing encoding and storage standards that are designed for conventional videos. In VR industry, Cubemap Projection (CMP) is the most widely used projection method for encoding 360° VR videos. It deforms a sphere into a cube, then unfolds the cube's six faces and packs them into a flat rectangle. However, it exhibits pixel density variation at different regions due to severe projection distortion. More specifically, the center of a cube's face is closer to the sphere, while the corners are further away. Therefore, the corners get more pixels than the center because the longer line spans more pixels on the cube's face than the shorter line during standard radial CMP projection. Recently, Google proposed Equi-Angular Cubemap (EAC) [Bro17] to handle projection distortion and therefore reduce pixel density variation. Besides, encoding 360° VR video has gained attention from many standardization collaborative teams. For instance, the Joint Video Exploration Team (JVET) and the High Efficiency Video Coding (HEVC) [SSW*17] have been working on new approaches for video encoding and delivery for years.

Main Results: In this paper, we present a generalized CMP algorithm to improve the efficiency of standard CMP using polynomial approximation. Our experiments show that our new algorithm can significantly reduce the projection distortion and generate well bal-

† X. Y. Zhang is the corresponding author.

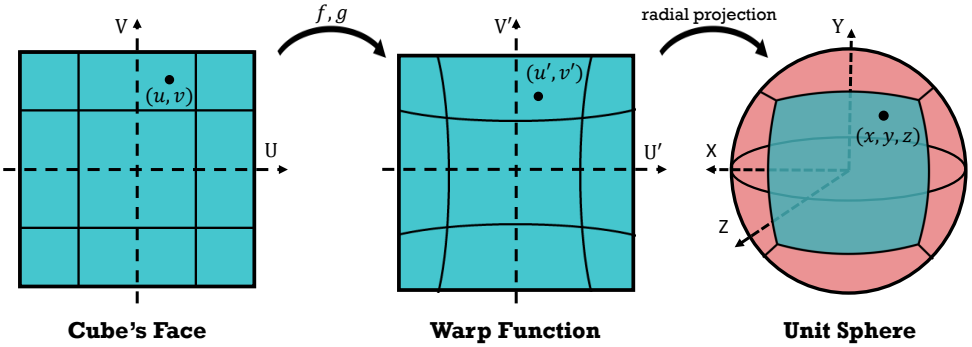

Figure 1: *Cubemap Projection (CMP) and Warp Functions*

anced pixel distribution in the resulting 360° VR videos. We use PSNR to evaluate the visual quality and the experimental results demonstrate promising performance improvement against standard CMP and even Google's EAC. Due to its simplicity, we avoid expensive trigonometric functions and their inverses used in some state of the art algorithms. Our work is expected to contribute to the ongoing standardization efforts towards efficiently encoding 360° VR videos.

The rest of the paper is organized as follows. In Section 2, we briefly survey prior work on encoding 360° VR videos. In Section 3, we introduce CMP and projection distortion. In Section 4, we present our generalized CMP using polynomial approximation. In Section 5, we provide implementation details and comparison, and highlight the performance improvement. In Section 6, we present our conclusion and future work.

2. Related Work

Although our primary motivation is encoding 360° VR videos, our work is closely related to the research in other fields, like map-making studied in cartography [Sny87], environment maps used for image-based rendering and lighting [Gre86], parameterization used in geometry processing and texture mapping [FH05, WD97, PCK04]. A brief review of a few related works is given below.

To encode 360 videos, a projection is used to map a spherical pixel onto planar rectangle or triangle. The most popular current method for projection is standard Cubemap Projection (CMP) [Gre86]. First, it projects a spherical surface onto six faces of a cube, then the six faces are packed to form a rectangle. Figure 1 depicts the map between a cube's face and $\frac{1}{6}$ spherical surface. The standard CMP uses a simple radial projection and therefore has substantial radial distortion. Recently, a joint work between YouTube and Google Daydream suggested Equi-Angular Cubemap (EAC) in order to improve equal angle pixel distribution [Bro17]. Besides CMP, many other projection methods have been proposed over the centuries. For example, Equirectangular Projection (ERP) is the oldest, yet simplest projection used in cartography [Sny87]. It is a normal cylindrical projection in which meridians are mapped to equally spaced vertical lines and circles of latitude (parallels) are mapped to horizontal lines. Though it has severe distortion near the sphere poles, it is still popular in cartography and even in 360° VR videos due to its simplicity. Recently, an improved ERP was suggested using hybrid cylindrical projection [TZ19]. Its form is relatively complex.

Some other methods use more complex domains for projections, such as Compact Octahedron Projection (COHP) [PH03, ED08, GPM*18] and Compact Icosahedron Projection (CISP) [Fis43]. COHP and CISP project spherical triangular surfaces onto planar triangular domains. COHP uses eight triangles for projection domain and eventually generates a rectangle using complex tiling scheme. A few variations like the work in [GPM*18] were also suggested to improve pixel density. CISP uses twenty triangles. Due to its sophisticated mapping and a large number of internal discontinuous seams, it is not clear if CISP is useful in practice. some trigonometric functions were used in these methods, which sometimes can be expensive for computation.

A projection based on high-order polynomial could in principle resolve the distortions exhibiting in the standard CMP, apart from avoiding trigonometric functions. In this paper, we show that a generalized CMP for encoding 360° VR videos can be very simple, yet efficient with promising performance improvement.

3. CMP & Projection Distortion

3.1. CMP

CMP is a map between a unit sphere and a cube. Each cube's face corresponds to a $\frac{1}{6}$ spherical surface. For the projection shown in Figure 1, we establish a U-V coordinate system in each cube's face and any point on the cube's face can be represented by coordinates (u,v), where $u,v \in [-1,1]$. The center of the cube is the origin of the 3D coordinate system and a cube has the side length 2. In general, the process of projecting a cube's face onto a sphere can be divided into two stages (shown in Figure 1).

First, we use warp functions

$$\begin{cases} u' = f(u,v) \\ v' = g(u,v) \end{cases} \tag{1}$$

in U-V coordinate system to transform (u,v) to point (u',v') in U′-V′ coordinate system. Note that the central point of the cube's face remains unchanged after transformation because of the symmetry

in the horizontal and vertical directions of the square. Moreover, in order to ensure the continuity at boundaries after projection, we let $f(1,v) = 1$ and $g(u,1) = 1$.

Second, without loss of generality, we directly project the vector (u',v') onto the unit sphere by normalizing the vector $(u',v',1)$ to obtain the unit vector $\vec{p} = (x,y,z)$.

$$\vec{p} = (x,y,z) = \frac{(u',v',1)}{\sqrt{u'^2+v'^2+1}} = \frac{(f(u,v),g(u,v),1)}{\sqrt{f^2(u,v)+g^2(u,v)+1}} \quad (2)$$

Here, we consider the cube's face on plane $z = 1$ and other faces can be analogously derived.

3.2. Projection Distortion

We use Area Stretching Ratio (ASR) [SLY17] to measure projection distortion. We define ASR as

$$\text{ASR} = \frac{\delta AREA(x,y,z)}{\delta AREA(u,v)}, \quad (3)$$

where $\delta AREA(*)$ is the surface area at a specified point. According to Arvo [Arv95], the corresponding ASR can be represented by

$$\text{ASR}(u,v) = \sqrt{\left(\frac{\partial\vec{p}}{\partial u}\cdot\frac{\partial\vec{p}}{\partial u}\right)\left(\frac{\partial\vec{p}}{\partial v}\cdot\frac{\partial\vec{p}}{\partial v}\right) - \left(\frac{\partial\vec{p}}{\partial u}\cdot\frac{\partial\vec{p}}{\partial v}\right)^2}, \quad (4)$$

where $\frac{\partial\vec{p}}{\partial u}$ and $\frac{\partial\vec{p}}{\partial v}$ denote the partial derivative of the unit vector $\vec{p}$ with respect to u and v. Eq. 4 can be further simplified and rewritten as follows

$$\text{ASR}(u,v) = \left(u'^2+v'^2+1\right)^{-\frac{3}{2}}\left[\frac{\partial f}{\partial u}\frac{\partial g}{\partial v} - \frac{\partial f}{\partial v}\frac{\partial g}{\partial u}\right]. \quad (5)$$

The warp functions used in standard CMP are

$$\begin{cases} u' = f(u,v) = u \\ v' = g(u,v) = v. \end{cases}$$

This simple form has been used in many applications, though it has severe projection distortion $\text{ASR}(u,v) = (u^2+v^2+1)^{-\frac{3}{2}}$.

The EAC proposed by Google reduces the projection distortion using the following warp functions

$$\begin{cases} u' = \tan\left(\frac{\pi u}{4}\right) \\ v' = \tan\left(\frac{\pi v}{4}\right). \end{cases} \quad (6)$$

Its ASR is given in Appendix.

4. Polynomial Approximation

4.1. Polynomial

As shown in Figure 1, we observe that a warp function generates distortion symmetric with respect to the axis for any face of the cube. Therefore, the warp functions must be odd, that is $f(-u) = -f(u)$ and $g(-v) = -g(v)$. Here, we use univariate odd polynomial to approximate such warp functions. Note that some algorithms may use non-polynomial functions like standard CMP and Google's EAC, and some may use bivariate functions (i.e., the warp function depends on both u and v).

Using polynomial, we denote warp functions f and g with respect to u and v as follows.

$$\begin{cases} f(u) &= k_1u + k_3u^3 + \cdots + k_{2n-1}u^{2n-1} \\ g(v) &= k_1v + k_3v^3 + \cdots + k_{2n-1}v^{2n-1}, \end{cases} \quad (7)$$

where $n = 1,2,\cdots$. Due to the symmetry, the formula of $f(u)$ and $g(v)$ are the same. Therefore, we will consider the warp function f only in the rest of the paper and the warp function g will have the same form. If we add a constraint of $f(1) = 1$, we have $\sum_{i=1}^{n} k_{2i-1} = 1$.

It is trivial to prove that the standard CMP is a polynomial approximation for $n = 1$ (i.e., $f(u) = u$ and $g(v) = v$, where $k_1 = 1$). This is also the inspiration for us to use high-order polynomial to achieve performance improvement with manageable computational costs. Therefore, we denote a generalized form of CMP using $(2n-1)$th-order polynomial as $(2n-1)$th-order CMP or simply $(2n-1)$-CMP, where $(2n-1)$ is the highest order in the polynomial. 1-CMP is standard CMP. Figure 2 shows a few polynomials after their parameters are determined.

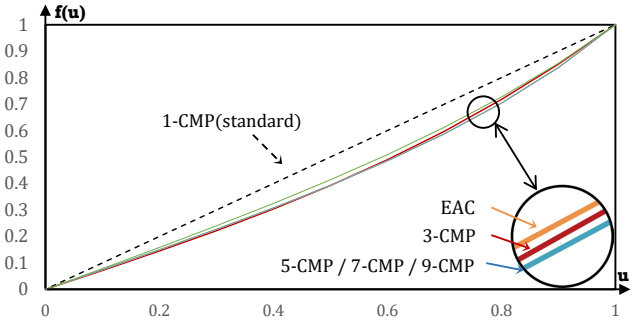

Figure 2: *1-CMP, 3-CMP, 5-CMP, 7-CMP and 9-CMP using corresponding polynomials*

4.2. Distortion Error

The ASR of an ideal area-preserving projection is a fractional area of a sphere and a cube. Without loss of generality, the ASR can be defined as

$$\text{ASR} = \frac{AREA(sphere)}{AREA(cube)} = \frac{4\pi}{24} = \frac{\pi}{6}. \quad (8)$$

Note that, the area-preserving projection is not relevant to the absolute ratio (i.e. the size of sphere or cube), but to the assumption that everywhere has the same ratio. Reducing the distortion is equivalent to finding a warp function by minimizing the relative error $|\text{ASR}(u,v) - \frac{\pi}{6}|$ for any point, where $\text{ASR}(u,v)$ is the area stretching ratio at point (u,v).

$$\text{E} = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N} \text{e}_{ij}, \quad (9)$$

where $\text{e}_{ij} = \left(\text{ASR}(u_i,v_j) - \frac{\pi}{6}\right)^2$ and $M\times N$ is the number of samples (e.g., 256×256 samples in $[0,1]\times[0,1]$).

In order to determine the polynomial defined in Eq. 7, we use

n-CMP	k_1	k_3	k_5	k_7	k_9
3-CMP	0.7136	0.2864			
5-CMP	0.7456	0.1305	0.1239		
7-CMP	0.7395	0.1814	0.0195	0.0596	
9-CMP	0.7405	0.1680	0.0688	-0.0067	0.0294

Table 1: *Parameters of nth order polynomial warp functions.*

n-CMP	k'_1	k'_3	k'_5	k'_7	k'_9
3-CMP	1.2908	-0.2908			
5-CMP	1.3432	-0.4865	0.1433		
7-CMP	1.3491	-0.5429	0.2672	0.0734	
9-CMP	1.3464	-0.4866	-0.0272	0.4925	-0.3251

Table 2: *Parameters of nth order inverse polynomial warp functions.*

Levenberg-Marquardt algorithm [Mar63] to compute their parameters $k_1, k_3, \cdots$. Levenberg-Marquardt algorithm is a damped least squares method to solve the nonlinear least squares problem. Here, we try to minimize the objective function given in Eq. 9 that expressed as the sum of squares of differences between the ASR computed using polynomial (see Eq. 7) and the ideal ASR $\frac{\pi}{6}$. Figure 2 demonstrates a few polynomials for 3-CMP, 5-CMP, 7-CMP and 9-CMP after their parameters are determined. In Figure 2, we also show the standard CMP (i.e., 1-CMP) and EAC for comparison.

Unlike standard CMP and EAC, the polynomial are not analytically invertible for $n > 2$. Therefore, we use the same method to approximate its inverse map. Since the inverse function of an odd function is still an odd function, we also choose univariate odd polynomials to approximate the inverse map. Analogously, we define the inverse map F and G as follows.

$$\begin{cases} F(u') & = k'_1 u' + k'_3 u'^3 + \cdots + k'_{2n-1} u'^{2n-1} \\ G(v') & = k'_1 v' + k'_3 v'^3 + \cdots + k'_{2n-1} v'^{2n-1} \end{cases}$$

Their parameters can be determined using the same algorithm suggested in [Mar63] by minimizing the following errors

$$\mathrm{E}' = \frac{1}{L} \sum_{i=1}^{L} \mathrm{e}'_i \tag{10}$$

where $\mathrm{e}'_i = |F(f(u_i)) - u_i|$.

Table 1 and Table 2 list the polynomial parameters of warp and inverse functions solved using Levenberg-Marquardt algorithm. Table 3 lists the error E' of the inverse functions, the corresponding maximum error e'^{max}_i and the accumulating error $\sum \mathrm{e}'_i$.

4.3. Error Analysis

Figure 3 shows the accumulating distortion errors of standard CMP, EAC and other high-order polynomial approximation. Obviously, the standard CMP has the worst performance in terms of accumulating distortion error. As the order of polynomial is increased, there is significant decrease in accumulating distortion error before 7-CMP. The change of the accumulating distortion error is not significant after 7-CMP. Our generalized CMPs using high-order polynomials show better performance against Google's EAC and standard CMP (1-CMP). Note that, in Figure 3, we do not show ERP since it does not use cube-map geometry and has worse distortion error than CMP.

n-CMP	E'	e'^{max}_i	$\sum \mathrm{e}'_i$
3-CMP	1.7300e-04	0.0207	3.4601
5-CMP	1.5613e-07	0.0011	0.0031
7-CMP	3.8014e-08	0.0004	0.0008
9-CMP	8.5771e-06	0.0098	0.1715

Table 3: E', e'^{max}_i *and* $\sum \mathrm{e}'_i$ *of the inverse functions.*

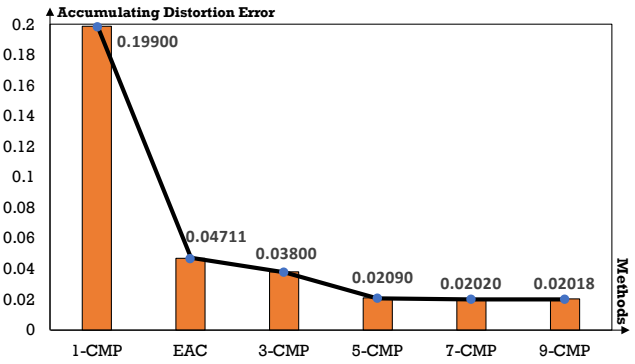

Figure 3: *Accumulating distortion errors of standard CMP, EAC and other high-order CMPs using polynomial approximation.*

5. Results & Comparison

In this section, we describe the implementation details of our approach and highlight the performance against standard CMP and EAC in terms of pixel distribution, PSNR and computational cost.

5.1. Implementation & Results

We have implemented our projection using C++ language under Windows. We use the public domain 360° VR video library, 360lib [YAB17], to implement our algorithm and to compare with other projections. We use a public domain library, Ceres Solver [AM] to solve Levenberg-Marquardt algorithm. As shown in Figure 9, we use our generalized CMPs to encoding 360° VR videos captured in real scenarios: beach, living room and gym. The performance was measured on a PC with Intel Core i7 3.4GHz CPU and 24GB memory.

5.2. Comparison

Local Distortion: In order to characterize local distortions caused by projection, we use Tissot's indicatrix to visualize the magnitude of distortion at given sample points. As shown in Figure 4, after projection, our generalized CMPs using high-order polynomial approximation exhibit better performance against other methods like standard CMP and EAC.

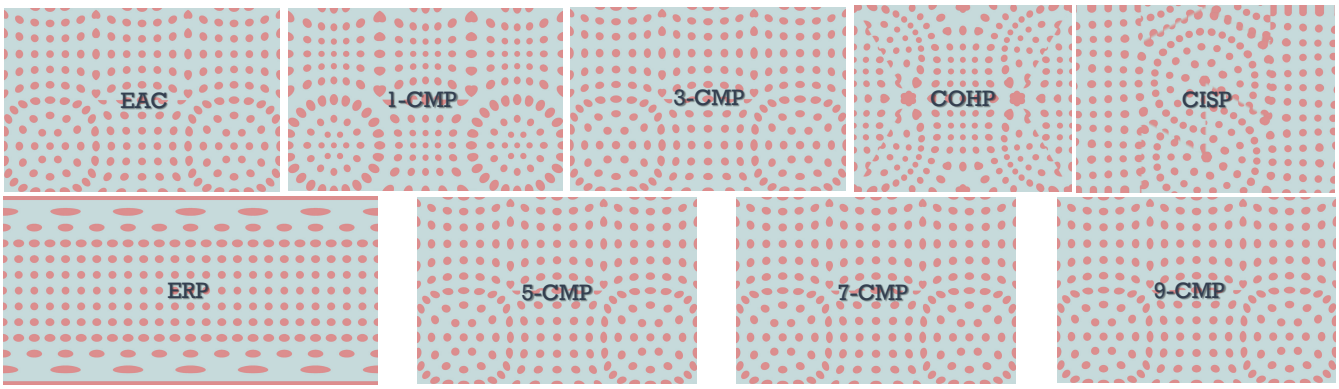

Figure 4: *The Tissot's indicatrix map, the higher order CMP, the less projection distortion.*

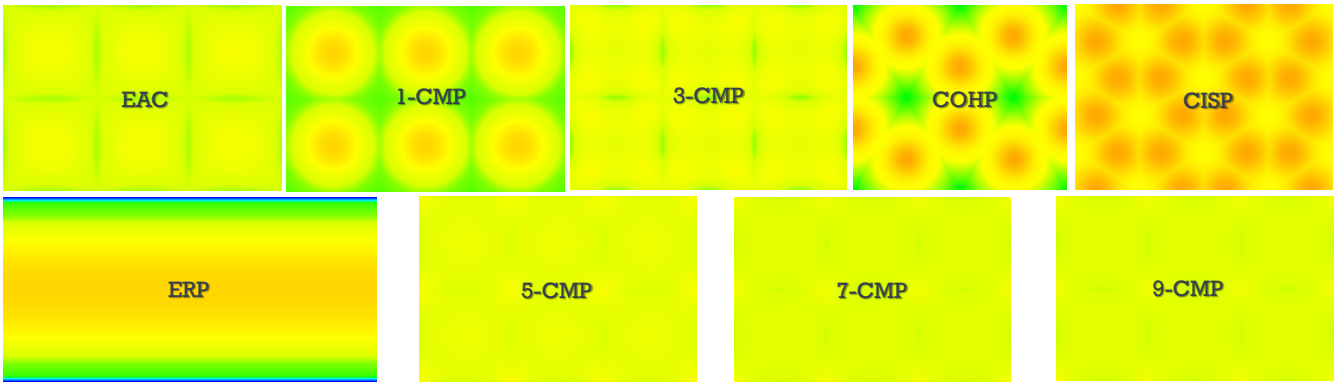

Figure 5: *The saturation maps. Uniform color distribution indicates less distortion, where blue and red represent over-sampling regions and under-sampling regions, respectively.*

Pixel Distribution: We also compare our method against others in terms of pixel distribution, shown in Figure 5. We use color-coded saturation map to visualize pixel density. The ideal projection has a saturation map that is uniform in color. Therefore, the more uniform in color, the higher quality of the encoded 360° VR videos. From the 3rd order polynomial, generalized CMP has better performance in pixel uniformity than EAC. After the 7th-order, uniformity has no significant increase.

Compared with EAC, CMP and ERP, other projections such as COHP and CISP have complex packing layouts and exhibit even worse performance (see Figure 4 and Figure 5). In particular, CISP has a large number of internal discontinuous seams and is not used in any 360 camera due to the sophisticated mapping.

S-PSNR & CPP-PSNR: Peak Signal-to-Noise Ratio (PSNR) has been widely used for objective QoE for image/video sequences. However, for 360° VR videos, since pixels are located on a spherical surface, it is non-trial to evaluate spherical images/videos using PSNR. Figure 6-(left) shows the results of PSNR for standard CMP, EAC and our generalized CMPs using high-order polynomial approximation. Not surprisingly, though ERP has severe distortion, it still outperforms other methods in PSNR merely because ERP has continuous texture. As PSNR does not account for projection distortion, it is not suitable for evaluating 360° VR videos.

Therefore, spherical-PSNR (S-PSNR) [SLY17, BAAY18] was proposed to evaluate omnidirectional video content, and Craster Parabolic Projection PSNR (CPP-PSNR) [VZ16, BAAY18] was suggested to evaluate distortions. These two metrics are more accurate than PSNR for evaluating 360° VR videos. Here, we use S-PSNR and CPP-PSNR to estimate the objective quality of 360° VR videos during the codec process. In general, a higher S-PSNR/CPP-PSNR indicates the higher quality of a video. Figure 6-(middle) and -(right) show the results of S-PSNR and CPP-PSNR for standard CMP, EAC and our generalized CMPs. Our generalized CMPs (especially 5-CMP) outperform EPR, standard CMP and EAC in both S-PNSR and CPP-PSNR.

As shown in Figure 6, COHP and CISP have worse PSNR than our high-order generalized CMPs due to internal discontinuous seams and nonuniform pixel distribution (see Figures 4 and 5).

Computational Costs: We evaluate computational costs during video encoding and projection. We use three video sequences to test the performance of encoding 360° VR. Using standard CMP method as reference and all other projection methods are compared with standard CMP. The average timings results are shown in Figure 7. We observe a reasonable and allowable increasing in computational costs with the order of polynomial.

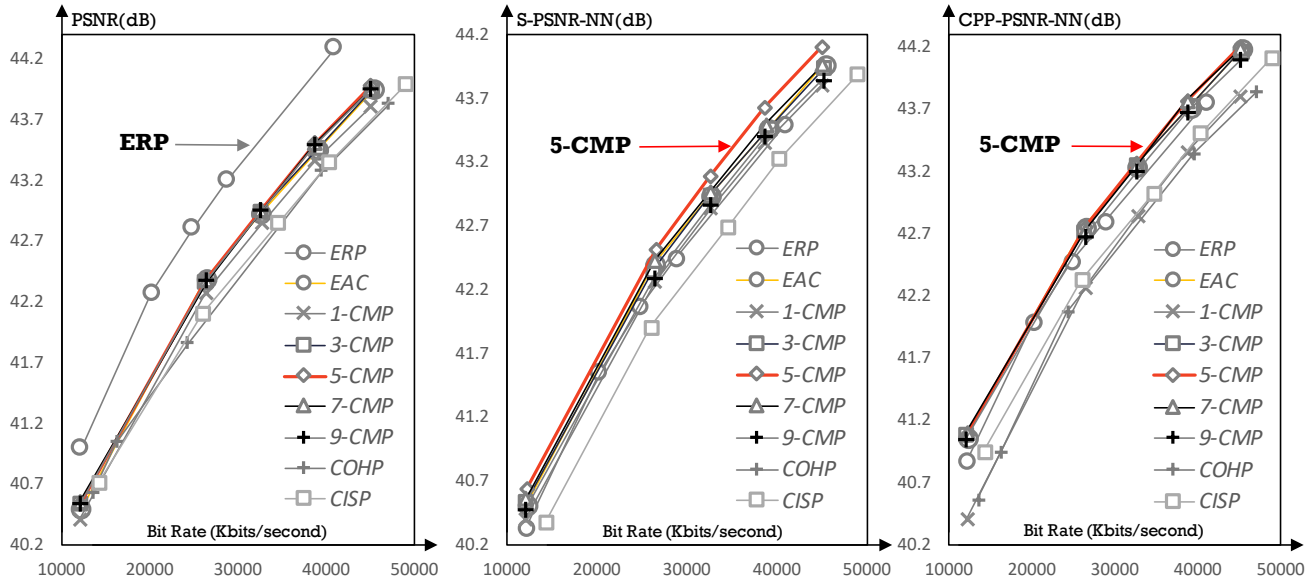

Figure 6: *Objective QoE using PSNR based metrics: (left) PSNR, (middle) S-PSNR-NN and (right) CPP-PSNR-NN.*

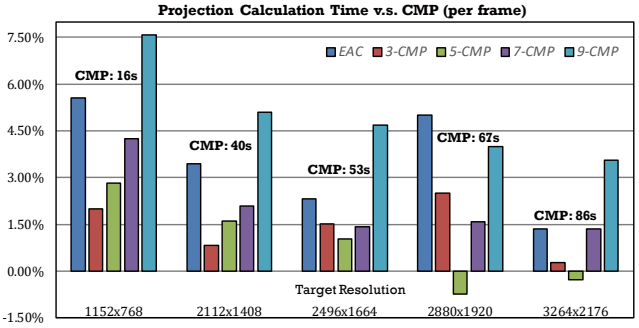

Figure 7: *Computational costs of projection type converting & encoding.*

5.3. Discussion

Based on the comparison given in Section 5.2, we can visually understand the location of dominant distortions, especially using pixel distribution. Here, we further analyze distortion using the area stretching ratio that is derived in Appendix. Figure 8 shows the ASR along a number of directions $v = 0$, $v = 0.5$, $v = 1.0$ and $v = u$. This may help us to understand the source of distortions. Observe that the ideal ASR $\frac{\pi}{6}$ is illustrated with a horizontal orange line. The ASR variance increases or decreases gradually from the center to the edge (i.e., from $u = 0$ to $u = 1$). For instance, when $v = 1.0$, the ASR variance of 5-CMP, 7-CMP and 9-CMP increases and reaches to the maximum at the cube's corner (see Figure 8-(c)). This may be caused by the pre-specified constraints $f(1, v) = 1$ and $g(u, 1) = 1$ that are used to maintain texture continuity at the cube's edges. However, these constraints can have significant impact on the ASR. It would be possible to relax these constraints along one direction (e.g., along V), so that the ASR can be further improved without influencing texture continuity.

Note that, to further improve pixel density, we may incorporate other non-polynomial functions like trigonometric functions in our generalized formulation. However, their computational costs are higher than polynomials. Moreover, computing their parameters is non-trivial.

6. Conclusion

We presented a new algorithm to improve CMP that has been widely used in 360° VR videos. We use high-order polynomial to approximate projection and achieve high performance improvement in pixel density with reasonable computational costs. Based on our approach, standard CMP can be regarded as 1-order CMP (i.e., approximation using 1st-order polynomial). We have thoroughly analyze the performance (distortion) of our algorithm for 3rd-order, 5th-order, 7th-order and 9th-order polynomial approximation. We applied our algorithm to 360° VR videos and observed the performance improvement in the quality of encoding 360° VR video. We also use PSNR, S-PSNR and CPP-PSNR to evaluate the visual quality and the experimental results show promising performance improvement against standard CMP and Google's EAC.

In our algorithm, we use univariate odd polynomials to reduce projection distortion. In future work, we would like to investigate bivariate polynomials functions. We expect to further improve projection efficiency, but it may increase computational costs. As discussed in Section 5.3, it would be possible to relax the constraints specified at the cube's edge $f(1, v) = 1$ and $g(u, 1) = 1$, so that the ASR can be further improved. Moreover, in addition to Eq. 9, other objective function may be used.

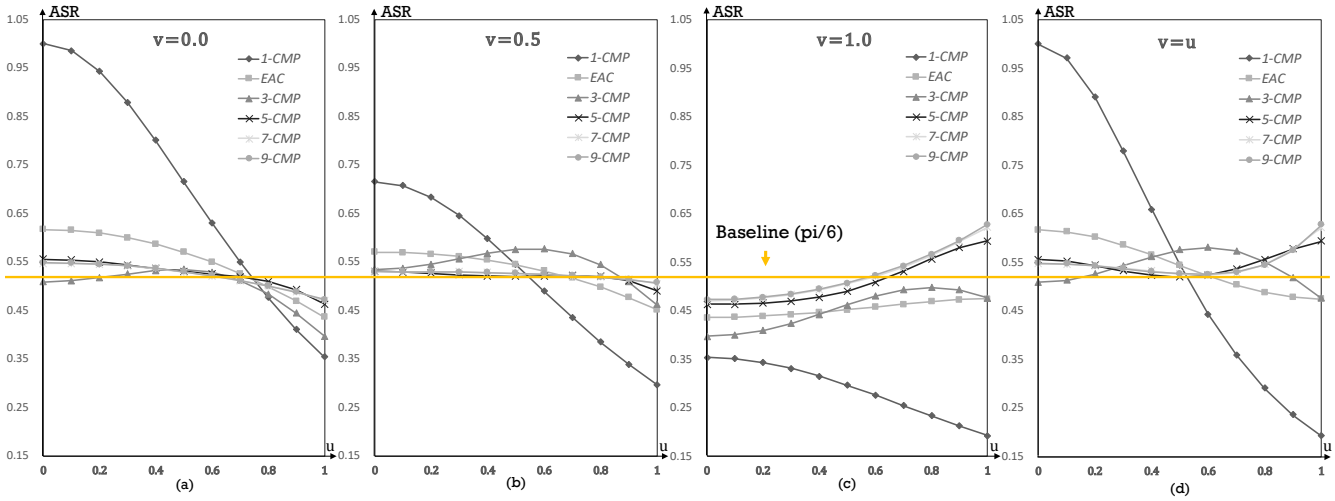

Figure 8: *The ASR variance along a number of directions $v = 0$, $v = 0.5$, $v = 1.0$ and $v = u$.*

Acknowledgments

This research work was supported by the NSFC (No.61631166002, No.61572196).

Appendix

In this appendix, we provide the deviation of ASR in EAC and ASR in our high-order CMP.

Appendix A: ASR in EAC

Given the EAC's warp functions in Eq. 6, their partial derivatives with u and v are

$$\begin{cases} \frac{\partial f}{\partial u} & = \frac{\pi}{4\cos^2\left(\frac{\pi u}{4}\right)} \\ \frac{\partial g}{\partial v} & = \frac{\pi}{4\cos^2\left(\frac{\pi v}{4}\right)} \\ \frac{\partial f}{\partial v} & = 0 \\ \frac{\partial g}{\partial u} & = 0 \end{cases}$$

According to Eq. 5, its ASR is

$$\begin{aligned} \mathrm{ASR}(u,v)_{\mathrm{EAC}} &= \left(u'^2 + v'^2 + 1\right)^{-\frac{3}{2}} \cdot \frac{\partial f}{\partial u}\frac{\partial g}{\partial v} \\ &= \left(\tan^2\left(\frac{\pi u}{4}\right) + \tan^2\left(\frac{\pi v}{4}\right) + 1\right)^{-\frac{3}{2}} \cdot \frac{\pi}{4\cos^2\left(\frac{\pi u}{4}\right)} \frac{\pi}{4\cos^2\left(\frac{\pi v}{4}\right)} \end{aligned}$$

Appendix B: ASR in $(2n-1)$-CMP

For the warp functions given in Eq. 7, their partial derivatives with u and v are

$$\begin{cases} \frac{\partial f}{\partial u} & = \sum_{i=1}^{n}\left[(2i-1)k_{2i-1}u^{2i-2}\right] \\ \frac{\partial g}{\partial v} & = \sum_{i=1}^{n}\left[(2i-1)k_{2i-1}v^{2i-2}\right] \\ \frac{\partial f}{\partial v} & = 0 \\ \frac{\partial g}{\partial u} & = 0 \end{cases} \tag{11}$$

Then we have the following ASR of $(2n-1)$-CMP using polynomial approximation

$$\mathrm{ASR}(u,v)_{n-\mathrm{CMP}} = \left(f^2 + g^2 + 1\right)^{-\frac{3}{2}} \cdot \frac{\partial f}{\partial u}\frac{\partial g}{\partial v} \tag{12}$$

where f and g are defined in Eq. 7 and $\frac{\partial f}{\partial u}$ and $\frac{\partial g}{\partial v}$ are given in Eq. 11.

References

[AGB*16] ANDERSON R., GALLUP D., BARRON J. T., KONTKANEN J., SNAVELY N., HERNÁNDEZ C., AGARWAL S., SEITZ S. M.: Jump: Virtual Reality Video. *ACM Transactions on Graphics 35*, 6 (Nov. 2016), 198:1–198:13. 1

[AM] AGARWAL S., MIERLE K.: Ceres solver. http://ceres-solver.org. 4

[Arv95] ARVO J.: Stratified sampling of spherical triangles. In *SIGGRAPH* (1995), pp. 437–438. 3

[BAAY18] BOYCE J., ALSHINA E., ABBAS A., YE Y.: Jvet-d0193: Test conditions for 360 video. 5

[Bro17] BROWN C.: Bringing pixels front and center in VR video, 2017. URL: https://blog.google/products/google-vr/bringing-pixels-front-and-center-vr-video/. 1, 2

[BYS*17] BHADSAVLE S. S., YAP X. H. S., SEGLER J., JAISIMHA R., RAMAN N., FENG Y., BIGGS S. J., PEOPLES M., BRENNER R. B., MCCANN B. C.: Immerj: A novel system for democratizing immersive storytelling. In *IEEE Virtual Reality* (2017). 1

[ED08] ENGELHARDT T., DACHSBACHER C.: Octahedron environment maps. In *VMV* (2008), pp. 383–388. 2

[FH05] FLOATER M., HORMANN K.: *Advances in Multiresolution for Geometric Modelling*. Springer, Berlin, Heidelberg, 2005, ch. Surface Parameterization: A Tutorial and Survey. 2

[Fis43] FISHER I.: A world map on a regular icosahedron by gnomonic projection. *Geographical Review 33*, 4 (1943), 605–619. 2

[GPM*18] GUO J., PEI Q., MA G., LIU L., ZHANG X.: A new uniform format for 360 VR videos. In *Pacific Graphics* (2018), pp. 245–254. 2

[Gre86] GREENE N.: Environment mapping and other applications of

world projections. *IEEE Computer Graphics and Applications 6*, 11 (1986), 21–29. 2

[HCC*17] HE T., CHEN X., CHEN Z., LI Y., LIU S., HOU J., HE Y.: Immersive and collaborative taichi motion learning in various VR environments. In *IEEE Virtual Reality* (2017), pp. 307 – 308. 1

[LHLK17] LEE H., HA G., LEE S., KIM S.: A mixed reality telepresence platform to exchange emotion and sensory information based on MPEG-V standard. In *IEEE Virtual Reality* (2017). 1

[Mar63] MARQUARDT D.: An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal for Industrial and Applied Mathematics* (1963), 431–441. 4

[PCK04] PURNOMO B., COHEN J. D., KUMAR S.: Seamless texture atlases. In *Eurographics/SIGGRAPH Symposium on Geometry Processing* (2004), pp. 65–74. 2

[PH03] PRAUN E., HOPPE H.: Spherical parametrization and remeshing. In *ACM Transactions on Graphics* (2003), vol. 22, ACM, pp. 340–349. 2

[SLY17] SUN Y., LU A., YU L.: Weighted-to-spherically-uniform quality evaluation for omnidirectional video. *IEEE Signal Processing Letters 24*, 9 (Sept 2017), 1408–1412. doi:10.1109/LSP.2017.2720693. 3, 5

[Sny87] SNYDER J. P.: *Map projections–A working manual*, vol. 1395. US Government Printing Office, 1987. 2

[SSW*17] SKUPIN R., SANCHEZ Y., WANG Y.-K., HANNUKSELA M. M., J. BOYCE M. W.: Standardization status of 360 degree video coding and delivery. In *IEEE Visual Communications and Image Processing (VCIP)* (2017), pp. 31–36. 1

[TZ19] TANG J., ZHANG X.: Hybrid projection for encoding 360 VR videos. In *IEEE Virtual Reality* (2019). 2

[VZ16] V. ZAKHARCHENKO A. DSOUZA E. A. A. S.: *AhG8: Suggested testing procedure for 360-degree video*. Tech. rep., Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Oct. 2016. 5

[WD97] WEINHAUS F. M., DEVARAJAN V.: Texture mapping 3D models of real-world scenes. *ACM Computer Survey 29*, 4 (Dec. 1997), 325–365. 2

[XZLD18] XU F., ZHAO T., LUO B., DAI Q.: Generating VR live videos with tripod panoramic rig. In *IEEE Virtual Reality* (2018), pp. 31–36. 1

[YAB17] YE Y., ALSHINA E., BOYCE J.: Algorithm descriptions of projection format conversion and video quality metrics in 360Lib. Joint Video Exploration Team of ITU-T SG, 2017. 4

Figure 9: *Encoding 360 video using various methods. From top to bottom: ERP, EAC, 1-CMP, 3-CMP and 5-CMP. Here, we did not include 7-CMP and 9-CMP since we observe no noticeable visual difference compared with 5-CMP. In most scenarios, 5-CMP can be considered as satisfactory.*