2015 IEEE International Symposium on Mixed and Augmented Reality

A Framework to Evaluate Omnidirectional Video Coding Schemes

Matt Yu Haricharan Lakshman Bernd Girod
Department of Electrical Engineering
Stanford University

Abstract

Omnidirectional videos of real world environments viewed on head-mounted displays with real-time head motion tracking can offer immersive visual experiences. For live streaming applications, compression is critical to reduce the bitrate. Omnidirectional videos, which are spherical in nature, are mapped onto one or more planes before encoding to interface with modern video coding standards. In this paper, we consider the problem of evaluating the coding efficiency in the context of viewing with a head-mounted display. We extract viewport based head motion trajectories, and compare the original and coded videos on the viewport. With this approach, we compare different sphere-to-plane mappings. We show that the average viewport quality can be approximated by a weighted spherical PSNR.

Index Terms: H.5.1 [Multimedia Information Systems]: Virtual Reality; E.4 [Coding and Information Theory]: Data compaction and compression

1 Introduction

Historically, virtual reality (VR) with head-mounted displays (HMDs) is associated with gaming applications and computer-generated content. However, the ability to show wide field of view content to a user can be used to provide immersive visual experiences involving real-world scenes. We refer to such applications as Cinematic VR. To this end, a real-world environment has to be captured in all directions resulting in an omnidirectional video corresponding to a viewing sphere.

Modern HMDs have the ability to track head motion with low latency, which can be used to present the view that corresponds to the direction the user is facing. Also, a separate view is presented to each eye so as to simulate depth. In Cinematic VR, this translates to stereoscopic omnidirectional video with horizontal parallax between the views. With advances in camera rigs and HMDs, the delivery of Cinematic VR content may soon become the bottleneck due to the high bitrate required for representing such content. Modern video coding standards are not designed to handle spherical video. Therefore, a spherical video is mapped to a rectangular plane resulting in so-called panoramic video. There are many ways to map a sphere onto a rectangle [1]. A number of different compression schemes have been proposed in literature for coding omnidirectional videos to reduce the bitrate [2][3][4][5]. However, different mappings and different test criteria have been employed to report coding efficiency. The main goal of this paper is to design a unified framework for evaluating the coding efficiency of omnidirectional videos. Using this framework we evaluate different mappings in terms of peak signal-to-noise ratio (PSNR) of the view presented on the HMD.

This work naturally extends to applications in augmented reality (AR). Omnidirectional videos provide content for location-based AR systems. Also, omnidirectional videos can be used as overlays rather than completely replacing a user's environment.

2 Related Work

Most previous research in generating panoramas is directed toward optimizing the representation for human viewing. The impact of such mappings on the coding efficiency of a video encoder has not yet been studied in detail. The work in [6] proposed a method for content preserving projections, with the help of manual inputs, for viewing mapped panoramas. Multi-plane perspective projections were proposed in [7] to reduce distortion in foreground objects, also for viewing the resulting panorama. Mapping schemes can also be evaluated using some attributes like sampling uniformity, area deviation, shape distortion, etc.

For encoding omnidirectional videos, the approach in [8] proposes to use spherical harmonics to encode directly in the spherical domain. However, by moving away from a rectangular block-based hybrid coding architecture, a lot of recent performance improvements in modern video coding techniques (e.g., H.264/AVC, H.265/HEVC) are lost. One of the early studies on the impact of using panoramic projection on H.264/AVC encoding was conducted in [5]. However, only the areas of the viewing sphere in the vicinity of the equator were considered. In fact, the areas near the poles of the viewing sphere may incur maximum distortion in commonly used panoramic projections. Various projection surfaces can be used for mapping a sphere, e.g., cubic [2], cylindrical [3], dodecahedron [9], etc. Furthermore, even if a projection surface is fixed, there are multiple ways of mapping the sphere onto the chosen surface. After mapping and encoding, many of the proposed compression schemes compute the coding error in the panoramic domain. However, the error in the panoramic domain does not reflect the error on the original sphere because of the reverse mapping required to get back the points on the sphere. To account for this difference in the relative importance of pixels in the panoramic domain, [9] proposed to multiply the error at each pixel by its corresponding solid angle covered on the sphere. However, many aspects have not been addressed in the literature:

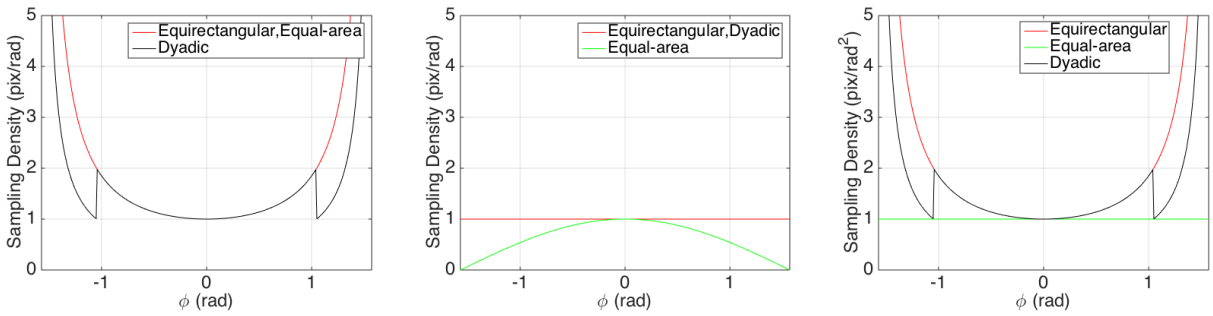
(a) For spherical videos, it is unclear how to compare high resolution ground truth videos with coded lower resolution videos, especially when the videos are represented using different panoramic projections.

(b) All points on a viewing sphere might not share the same viewing probability, e.g., we are more likely to view content in the vicinity of the equator than the poles.

(c) Furthermore, since the screen of most displays are planar, with a limited field of view, the final view presented to the user according to the current head position involves a projection from the sphere to the focal plane of the display. This fact is considered in [4] and the error on the viewport that is presented to the user is computed. However, it does not deal with all 3 degrees of rotation (see Fig. 2) possible with an HMD.

(d) In a system which streams video data from a server to a client with an HMD, the data requested by the client may arrive with a delay. In such cases, the client may employ concealment schemes (e.g., repeating last available viewport). It is desirable that the evaluation framework handles the impact of this latency as well.

978-1-4673-7660-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ISMAR.2015.12



(a) (b) (c)

Figure 1: Horizontal (a), vertical (b), and combined (c) sampling density of different projections, relative to the horizontal, vertical, and combined sampling density at the equator on an equirectangular projection, as a function of latitude ϕ.

Contributions of this paper

We first propose a method to compare the original and the coded omnidirectional videos by generating viewports corresponding to head motion data to compute the peak signal-to-noise ratio (PSNR) between the viewports. This gives an estimate of the quality of views presented to the user. We use this metric to study the impact of various panoramic projections on the coding efficiency of a video encoder. However, while designing a coding system, the actual head motion data is not known beforehand. Therefore, we propose a sphere based PSNR computation, denoted as S-PSNR, to approximate the average quality over all possible viewing directions. Then, we consider the fact that not all viewing directions are equally likely, e.g., users are more likely to view areas around the equator than the poles. We use head motion data over a set of users and estimate relative frequencies of accessing different points on the sphere. Thus, we compute weighted S-PSNR and show that this can approximate the average viewport PSNR without explicit head motion data.

The source code for all the evaluation metrics proposed in this paper can be found at `https://github.com/mattcyu1/omnieval`.

3 Review of Panoramic Projections

We consider capture systems using either a wide-angle optical setup or computational methods like stitching videos from multiple cameras to generate omnidirectional videos. Using computational methods can lead to stitching errors consisting of artifacts like tearing and image doubling. Mild tearing artifacts tend not to affect coding efficiency at low bitrates since the coarse quantization will remove sharp edges. However, such artifacts may consume a large number of bits at high bitrates since the coder will work to preserve these features. In this work, we choose a dataset which has minimal stitching artifacts. In order to store the omnidirectional video in memory, it is addressed using latitudes and longitudes, forming a panoramic projection from the sphere to a plane. Different panoramic projections (equirectangular, equal-area, Mercator, cubic, etc.) can give rise to very different sampling patterns. The horizontal, vertical, and combined sphere sampling densities of the cylindrical projections compared in this paper are shown in Fig. 1. Note that considering the combined sampling density alone ignores the large increase in horizontal sampling density of these cylindrical projections at the north and south poles.

Here we briefly review the panoramic projections used in our comparison.

- Equirectangular: This projection uses a constant spacing of latitude $\phi \in [-\pi/2, \pi/2]$ and longitude $\theta \in [-\pi, \pi]$ and addresses the vertical and horizontal positions in a panorama using ϕ and θ, respectively. Due to the constant spacing of latitude, this projection has a constant vertical sampling density on the sphere. However, horizontally, each latitude ϕ (whose circumference is given by $\cos\phi$) is stretched to a unit length to fit in a rectangle. Therefore, the horizontal sampling density at latitude ϕ is given by $1/\cos\phi$, which tends to infinity near the poles.

- Lambert Cylindrical Equal-area: This projection attempts to compensate for the increasing horizontal sampling density as we go near the poles by decreasing the corresponding vertical sampling density. Specifically, the vertical sampling density is set to $\cos\phi$ so that the combined sampling density is constant throughout the sphere, hence the name equal-area.

- Dyadic: While the equal-area projection modified the vertical sampling density to compensate for the horizontal oversampling, here we design a projection which directly decreases the horizontal oversampling of the equirectangular projection. This is achieved by halving the horizontal resolution of the panorama for $|\phi| \geq \frac{\pi}{3}$.

- Cubic: This projection places the sphere of unit diameter at the center of a cube with unit length sides. Each face of the cube is generated by a rectilinear projection with a 90° field of view in horizontal and vertical directions. This results in a sampling density that varies over each face of the cube. The sampling density is lowest at the center of the cube faces and highest where the cube faces meet.

The process of generating different panoramas from the ground truth signal is depicted in Fig. 2. We start with a target (integer) location $\mathbf{p}$ on the desired panorama. Different panoramic projections may map the same location $\mathbf{p}$ to different locations on the sphere, shown as $\mathbf{s}_1$ and $\mathbf{s}_2$. Therefore, the values at location $\mathbf{p}$ are computed from different locations on the ground truth signal $\mathbf{g}_1$ and $\mathbf{g}_2$. Bicubic interpolation is used to compute values at sub-pixel locations throughout this paper. The generated panoramas are encoded using a video encoder at various bitrates.

4 Viewport-based Quality Evaluation

Consider the visual information defined on a viewing sphere of unit radius centered at point O, as shown in Fig. 3. The viewport has a limited field of view and is modeled as a plane segment ABCD tangential to the sphere at the center O' of the viewport. In this section, we use head motion information from the HMD and compute the coding error in the viewport.

To determine the pixels in the viewport, we use the pinhole camera model, i.e., a scene view is formed by projecting 3D points onto

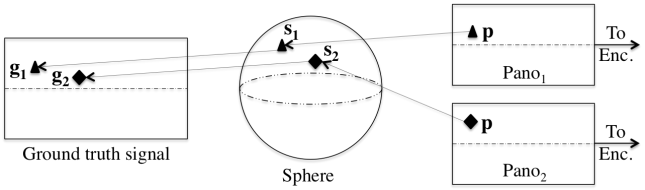


Figure 2: Generating different panoramic mappings from the ground truth signal.

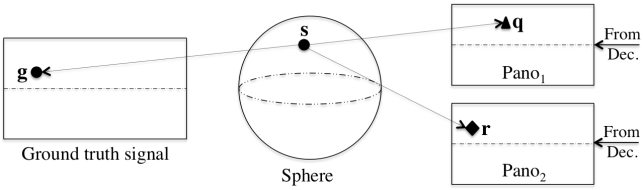


Figure 4: Comparison of the ground truth signal with coded panoramas on a set of uniformly sampled points on a sphere.

the image plane using a perspective transformation. If we uniformly span the spherical coordinates in the visible region of the sphere and pass rays from O to the points on the sphere, they will intersect the viewport plane with non-uniform spacing between the pixels. We refer to this as the *forward* projection. In order to compute a uniform grid of pixels in the viewport, we start with the desired locations (a.k.a. texture coordinates) in the viewport and *reverse* the mapping to compute corresponding locations on the sphere.

We assume that the canonical head position is such that the user is looking down the negative z-axis. Let $\mathbf{R}$ represent the rotation of the user's head relative to the canonical position. It is equivalent to keeping the user's head fixed at the canonical position and rotating the sphere by $\mathbf{R}^T$. Let the transformation from 3D coordinates to 2D homogeneous coordinates be modeled via an intrinsic matrix

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where,

- f_x and f_y is the focal length expressed in pixels For instance, if W is the width of the viewport in pixels and fov_x is the horizontal field of view per eye in the HMD, we have $\frac{W}{2f_x} = \tan(\frac{\text{fov}_x}{2})$.
- c_x and c_y are the texture coordinates of principal point O' in the viewport.

Let $\mathbf{E} = [x, y, z]^T$ denote a point on the sphere in the currently visible region and $\mathbf{e}'$ denote the 2D homogeneous coordinates of its projection on the viewport. The forward projection can then be written as

$$w \cdot \mathbf{e}' = \mathbf{K} \cdot \mathbf{R}^T \cdot \mathbf{E}, \tag{2}$$

where w denotes a scale factor. Using this formulation, we can carry out the reverse projection necessary to start from the desired texture coordinates $\mathbf{e}'$ and compute the coordinates on the unit sphere. This can be expressed as

$$\mathbf{E} = \mathbf{R} \cdot \frac{\mathbf{K}^{-1}\mathbf{e}'}{||\mathbf{K}^{-1}\mathbf{e}'||_2}. \tag{3}$$

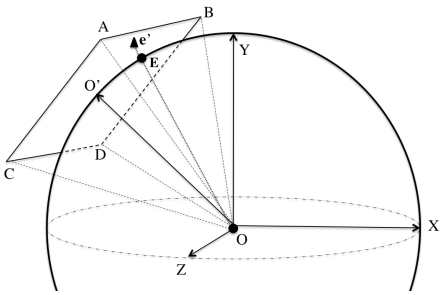


Figure 3: Example of a viewport.

We repeat this process for all the desired points on the viewport and determine the corresponding set of points on the sphere. Finally, this set of points is used to determine the coding error between the original and the coded videos.

In a system where the data requested by the client is not delivered in time for rendering, this viewport evaluation method will compute the error between the expected viewport cut out from the original and the actual viewport shown to the user. This allows our framework to evaluate various delivery schemes which may introduce delays in transmission.

5 Spherical Domain Comparison

Viewport based comparison can be used as a distortion measure if we have the knowledge of a user's head motion trajectory. However, this is not known upfront and different users may view the same video along different trajectories. Therefore, we develop S-PSNR, a spherical PSNR, to summarize the average quality over all possible viewports.

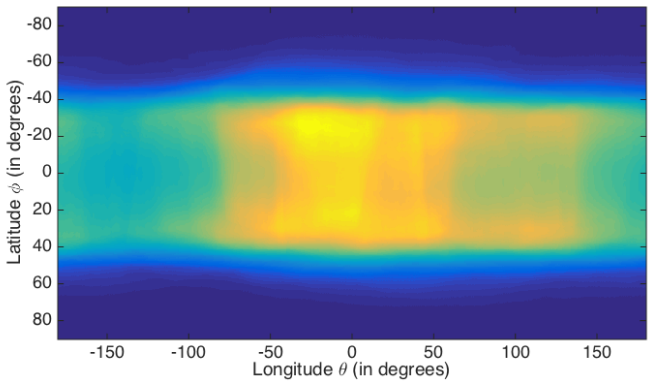
Fig. 4 depicts the proposed approach to compute the coding error. For this, instead of starting from the panoramic projection, we start with a set of uniformly sampled points on the sphere. For instance, a point on the sphere marked as **s** is mapped to corresponding locations on the ground truth **g** and a coded panorama **q**. The pixel values at these locations are computed and the error between these pixels is determined. Next, the location **r** on a different coded panorama, corresponding to the same point **s** on the sphere, is accessed and the error between the pixels at **g** and **r** is determined. The error over the entire set of points on the sphere is averaged to compute S-PSNR of different coded representations w.r.t. the ground truth.

Next, we observe that not all viewing directions are equally likely, e.g., users tend to look around the equator much more than the poles. We use the head motion data to train two types of statistics on the sphere:

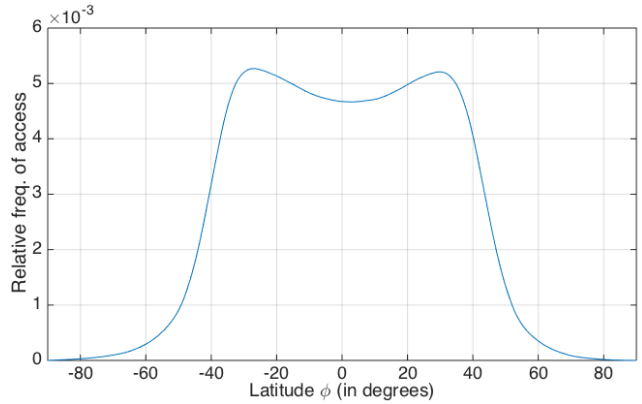
- Relative frequency of accessing different points on the sphere according to viewing probabilities. This is visualized as a heat map in Fig. 5a.
- Relative frequency of latitude-wise accesses, to succinctly capture the underlying dynamics, as shown in Fig. 5b.

We use the trained relative frequencies to weight the coding errors during the S-PSNR computation to better approximate the viewport quality a user would experience.

In Fig. 5b, note that although the viewing probability is expected to peak near the equator, the pixel access probability peaks near latitude $\phi = \pm 30^\circ$. This happens because the viewport is a perspective projection of the sphere onto a planar surface. Hence, uniform sampling of pixels on the user's viewport centered near the equator yields higher sampling density of areas on the sphere farther away from the equator.



(a)



(b)

Figure 5: Observed relative frequency of pixel accesses according to head motion trajectories. The figures show average statistics over 10 users viewing 10 omnidirectional videos each. (a) Visualization as an equirectangular heat map. Front direction corresponds to the (0,0) position. (b) Visualization marginalized over longitude. Although the viewing probability is expected to be the highest near the equator, notice that peak accesses happen near $\phi = \pm 30^\circ$ since uniform sampling on viewport positions results in a denser access of points on the sphere as we move away from the equator.

6 Experimental Results

We have described various methods in Sec. 4 and Sec. 5 to determine the quality of video data presented to users. Here we use these methods to study how the quality varies w.r.t the bitrate using an H.264/AVC codec. We consider a dataset[1] of 10 omnidirectional videos of length 10 sec each. We used a variety of scenes (e.g., biker riding around the camera, bus driving by a busy street, etc.) to capture different scenarios. While the duration of these videos is relatively short, we expect that the general statistics (e.g., users tend to watch content at the equator more than at the poles) to hold on longer videos. A group of 10 subjects were asked to watch these omnidirectional videos on an Oculus Rift DK2 using a custom video player and their head position was recorded over the entire duration. Participants were told to stand and then were given the freedom to turn around while wearing the HMD. It would be interesting to compare the difference in viewing statistics if the user was asked to sit (in both rotatable and non rotatable chairs). However, this comparison is left to future work.

The experimental evaluation is organized into two parts. In the first part, we evaluate various mapping schemes and their impact on the coding efficiency. Then, we consider the case of testing a coding system without the explicit knowledge of head motion trajectories.

[1]This dataset has been generously provided by Jaunt Inc.

Sequence	Equal-area	Cube	Dyadic
BMX	9.4%	11.4%	3.3%
Cannes	-0.2%	7.0%	-0.8%
China1	-7.3%	-4.0%	-6.1%
China2	-8.3%	7.7%	-7.1%
Kauai1	-9.4%	-10.4%	-9.0%
Kauai2	-20.1%	-16.7%	-16.4%
Kauai3	-11.3%	-8.1%	-7.9%
London	5.6%	10.7%	2.1%
Monument	-36.4%	-29.7%	-27.7%
Waterfall	-5.4%	1.3%	-3.3%
Avg	-8.33%	-3.09%	-7.29%

Table 1: BD-rate comparison of various projections relative to the Equirectangular projection using the viewport evaluation method described in Sec. 4.

Projection	WeightSph	LatSph	Sph	Quad
Equirectangular	6.85%	7.18%	16.46%	23.36%
Equal-area	5.42%	6.03%	13.10%	26.28%
Cube	6.48%	6.66%	13.55%	19.81%
Dyadic	6.08%	6.31%	13.90%	20.06%
Avg	6.21%	6.55%	14.25%	21.38%

Table 2: BD-rate comparison of various metrics described in Sec. 6.2 relative to the viewport evaluation method when using different projections.

6.1 Mapping Comparisons

We compare the mappings presented in Sec. 3, namely, Equirectangular, Lambert Equal-area, Dyadic, and Cubic. The Mercator projection was also evaluated, but the average performance was significantly worse than all other mappings, hence detailed results for the Mercator projection are not included here.

Due to limitations in processing power and memory access bandwidth in today's computing devices, we consider a resolution of 4Kx2K for the panoramas to be encoded, although higher resolutions would be beneficial for achieving higher pixels-per-degree in HMDs. One of the goals in this paper is to be able to compare an original omnidirectional video, which may be available in a certain panoramic projection, with a coded video potentially in a different panoramic projection and resolution. If the ground truth video were also at a resolution of 4Kx2K, it would unfairly bias the comparison of different panoramic projections toward the actual projection the ground truth is stored. In order to tackle this, we use a resolution of 6Kx3K for the ground truth (with equally spaced latitudes and longitudes) and remap it to different panoramic projections at 4Kx2K to be encoded, as shown in Fig. 2. We code each video at four QP settings and measure the corresponding bitrates. Viewport quality is calculated as presented in Sec. 4.

Tab. 1 summarizes the results by showing the performance of each mapping relative to the equirectangular projection. The average bitrate difference between the rate-distortion (RD) plots of the reference and the test mapping is summarized using the BD-rate metric [10]. Negative BD-rate numbers indicate bitrate savings w.r.t. the reference. It is observed that the average bitrate savings of the Equal-area projection over the Equirectangular projection is approximately 8.3%. The Dyadic projection also shows similar improvement over the Equirectangular projection. While the average

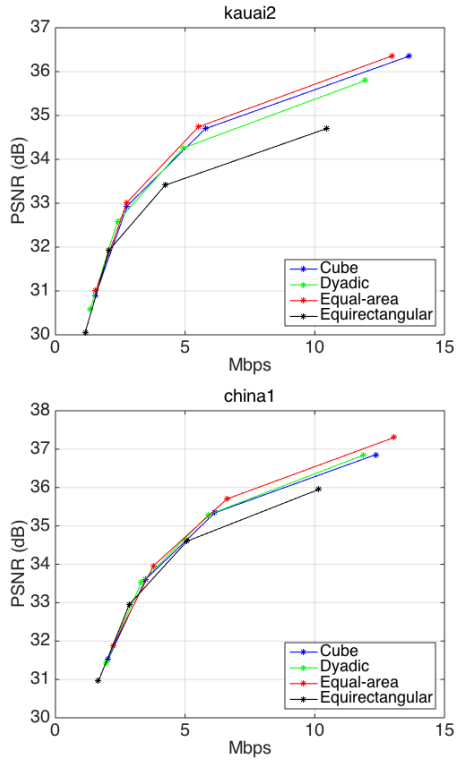


Figure 6: RD curves of two sequences for different panoramic projections using the viewport quality evaluation method.

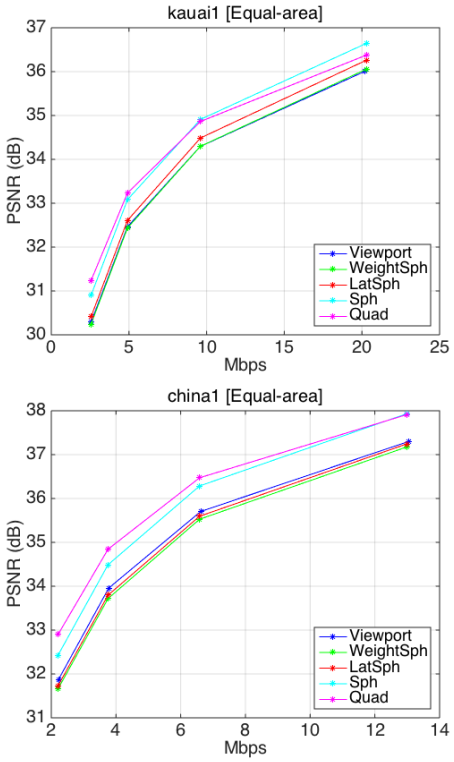


Figure 7: RD curves of two sequences coded using the Equal-area projection where the distortion is measured using various viewport quality approximation methods.

performance of the Cubic projection is lower than the Equal-area and Dyadic projections, the Cubic projection remains important since modern software like OpenGL support cube map rendering.

Fig. 6 shows the resulting RD curves for each panoramic projection for two representative sequences. In these two sequences, the viewport PSNR at a given bitrate is the lowest when using the Equirectangular projection, while it is the highest when using the Equal-area projection.

6.2 Spherical PSNR vs. Viewport PSNR

In this section, we study the capability of the methods presented in Sec. 5 to approximate the viewport quality. We perform the same coding experiments as in Sec. 6.1 and compute the resulting quality using the following quality metrics:

- WeightSph: S-PSNR with sphere points weighted by point access frequency.
- LatSph: S-PSNR with sphere points weighted by the corresponding latitude access frequency.
- Sph: S-PSNR where all points are weighted equally.
- Quad: PSNR calculated by mapping both the ground truth and the coded videos to the same 6Kx3K Equirectangular projection.

In addition to the methods discussed in Sec. 5, the Quad comparison is also included to illustrate the effects of not considering comparisons on the sphere.

We use BD-rate to compute the approximation error of each method with respect to the reference viewport evaluation method. Tab. 2 shows the average BD-rate for each mapping as well as the average BD-rate across all mappings. The WeightSph and LatSph methods differ from the reference by less than 7% on average without explicit head motion data. Interestingly, there is a large gap between the viewport method and the approximation using the Sph method, suggesting that the knowledge of general head motion statistics is required to closely approximate the viewport quality shown to a user. As expected, the Quad comparison is not able to closely approximate the viewport quality.

Fig. 7 shows the resulting RD curves for each of the quality evaluation methods for two representative sequences. It can be seen that WeightSph and LatSph methods are able closely approximate the viewport method with only general head motion statistics rather than the exact head motion data. The Sph and Quad methods yield significantly different approximations.

7 Conclusion

We have demonstrated a framework to evaluate the coding efficiency of omnidirectional videos for viewing on an HMD. This framework allows us to compare various sphere-to-plane mappings without bias toward any specific mapping or resolution. It is observed that the Equal-area mapping yields around 8.3% bitrate savings relative to the commonly used Equirectangular mapping. This framework accounts for user specific head motion trajectories when available, and otherwise falls back to general head motion statistics. We also show that it is possible to approximate the average viewport quality by exploiting general head motion statistics.

Acknowledgements

The authors wish to thank Dr. Roland Angst for valuable discussions and comments.

REFERENCES

[1] D. Zorin, and A. H. Barr, "Correction of geometric perceptual distortion in pictures," In Proc. SIGGRAPH, 1995.

[2] K. Ng, S. Chan, and H. Shum, "Data compression and transmission aspects of panoramic videos," IEEE CSVT, vol. 15, no. 1, January 2005.

[3] C. Gruenheit, A. Smolic, and T. Wiegand, "Efficient representation and interactive streaming of high-resolution panoramic views." IEEE International Conference on Image Processing, 2002.

[4] P. Alface, J. Macq, and N. Verzijp, "Interactive Omnidirectional Video Delivery: A Bandwidth-Effective Approach," Bell Labs Technical Journal, vol. 16, no. 4, March 2012.

[5] I. Bauermann, M. Mielke, and E. Steinbach, "H.264 based coding of omnidirectional video," in Proceedings of International Conference on Computer Vision and Graphics, September 2004.

[6] R. Carroll, M. Agrawala, and A. Agarwala, "Optimizing Content-Preserving Projections for Wide-Angle Images," SIGGRAPH, 2009.

[7] L. Zelnik-Manor, G. Peters, and P. Perona, "Squaring the Circle in Panoramas," IEEE International Conference on Computer Vision 2009.

[8] I. Tosic, and P. Frossard, "Low bit-rate compression of omnidirectional images," Picture Coding Symposium, 2009.

[9] C. Fu, L. Wan, T. Wong, and C. Leung, "The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video," IEEE TMM, vol. 11, no. 4, April 2009.

[10] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," ITU-T VCEG-M33, Apr. 2001.