Content Adaptive Representations of Omnidirectional Videos for Cinematic Virtual Reality

Matt Yu
Stanford University
350 Serra Mall
Stanford, CA 94305, U.S.A.
mattcyu@stanford.edu

Haricharan Lakshman
Stanford University
350 Serra Mall
Stanford, CA 94305, U.S.A.
hari.lakshman@stanford.edu

Bernd Girod
Stanford University
350 Serra Mall
Stanford, CA 94305, U.S.A.
bgirod@stanford.edu

ABSTRACT

Cinematic virtual reality provides an immersive visual experience by presenting omnidirectional videos of real-world scenes. A key challenge is to develop efficient representations of omnidirectional videos in order to maximize coding efficiency under resource constraints, specifically, number of samples and bitrate. We formulate the choice of representation as a multi-dimensional, multiple-choice knapsack problem and show that the resulting representations adapt well to varying content. We also show that separation of the sampling and bit allocation constraints leads to a computationally efficient solution using Lagrangian optimization with only minor performance loss. Results across images and videos show significant coding gains over standard representations.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Artificial, augmented, and virtual realities; I.4.10 [**Image Representation**]: Multidimensional

General Terms

Algorithms, Performance

Keywords

Cinematic VR, Immersive video, Virtual reality, Omnidirectional video coding

1. INTRODUCTION

Virtual reality (VR) refers to creating an artificial environment with immersive 3D visual experience. Modern head-mounted displays (HMDs) have the ability to display wide-field-of-view content at high pixel densities to provide immersion. Furthermore, these HMDs can track user head motion and update the displayed content with low latency. In computer simulated environments like VR games, the content can be rendered from the desired perspective by employing 3D models of the environment. Since creating accurate 3D models of real-world scenes is challenging, image-based rendering techniques are typically used in Cinematic VR applications. In Cinematic VR, a real-world scene is captured in all directions (e.g., with a camera rig) resulting in an omnidirectional video corresponding to a viewing sphere. To simulate depth, a separate view is generated and presented to each eye. This leads to stereoscopic omnidirectional video with corresponding parallax between the two views.

With advances in camera rigs and stitching algorithms for post-production, systems for content creation are undergoing continuous improvement. The delivery of Cinematic VR content may soon become the bottleneck due to the high bitrate required for representing such content. Unfortunately, modern video coding standards are not designed to handle spherical content. Therefore, a spherical video is mapped onto a rectangular plane, resulting in a panoramic representation, before encoding. A sphere can be mapped onto a plane in many ways [1], but no mapping can be distortion free [2]. We refer to this as sampling distortion. Next, the video encoder may introduce coding distortions in order to reduce the bitrate. The final reconstruction quality of a spherical video is a function of both sampling and coding distortions.

In this paper, we propose content adaptive representations of omnidirectional videos by jointly optimizing the sampling and coding stages.

2. RELATED WORK

Panoramic representations such as Equirectangular (constant spacing of lattitude and longitude), Equal-area cylindrical (decreasing vertical sampling to compensate for increasing horizontal sampling near poles), and cube map (projection of sphere onto cube) are commonly used to represent spherical content. Most previous research on optimizing sphere-to-plane mappings aim to generate panoramas for human viewing. For instance, a method for content preserving projections, with the help of manual inputs, was proposed in [3]. Multi-plane perspective projections were proposed in [4] to reduce the perceived distortion in foreground objects.

Many compression schemes have been proposed in literature for coding omnidirectional videos to reduce the bitrate [5, 6]. However, these methods encode panoramic representations without optimizing the sphere-to-plane mapping. Furthermore, these methods use different metrics to report performance which make it difficult to make comparisons. The proposal in [7] uses spherical harmonics to encode directly in the spherical domain. Unfortunately, a lot of recent performance improvements in modern video coding techniques (e.g., H.264/AVC, H.265/HEVC) cannot be directly applied in the spherical domain. One of the early studies on the impact of panoramic projection on H.264/AVC encoding was conducted in [8], however without considering polar regions. Mapping a sphere onto a cube and encoding the faces of the cube can be

ImmersiveME'15, October 30, 2015, Brisbane, Australia.

DOI: http://dx.doi.org/10.1145/2814347.2814348.

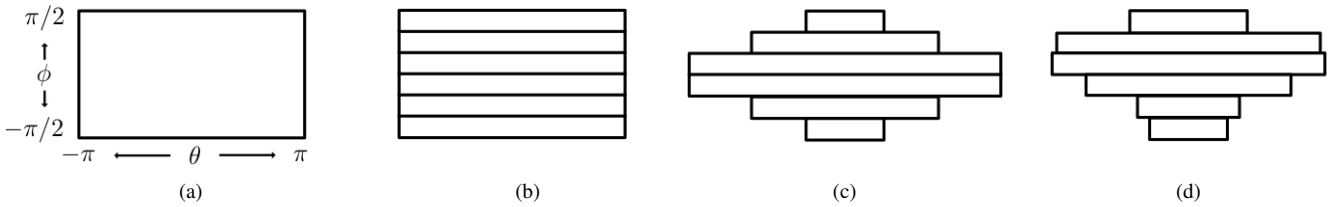


Figure 1: Illustration of panoramic representations: (a) equirectangular, (b) tiled representation, (c) adjusting sampling densities by resizing tiles, and (d) adjusting sampling densities to account for content specific statistics and user viewing probabilities.

an attractive way to reformat data to make it suitable for modern coding schemes. However, cube maps still suffer from the fact that they oversample the sphere near the edges relative to the center of the faces, due to perspective projection. One of the first schemes to adapt the encoding based on the statistics of a given omnidirectional video was proposed in [9]. However, only intra prediction was considered and the bit allocation across different parts of the panorama was optimized without taking into account the sampling problem.

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 back projection required to get the points on the sphere. Moreover, in Cinematic VR, only a viewport (i.e., cutout of the sphere) is shown to a user at any given time. In recent work [12], viewport PSNR, a metric to compare original and coded panoramic videos using head tracking data was proposed. Furthermore, it was observed that the points near the equator are more likely to be viewed than the points near the poles. To account for such user statistics, latitude weighted spherical PSNR (L-PSNR) was proposed and shown to be a good proxy for the actual viewport PSNR, since the user's viewing directions are not known beforehand.

Contributions of this paper: We propose a method to generate an adaptive representation using tiles that can exploit the statistics of both the underlying omnidirectional content and user viewing behavior. For a fair comparison of different representations, we introduce a limit on the maximum number of samples each representation may use. We optimize the representation to maximize L-PSNR while staying below the maximum number of samples and the target bitrate. Next, we show how the joint sample count and bitrate constraint can be relaxed. This enables separate optimization of the representation under the sample count constraint using Lagrangian optimization which yields faster optimization and increased scalability. Finally, we show how potential seams between tiles can be mitigated using overlapping tiles.

3. OPTIMIZED REPRESENTATIONS

Camera rigs with multiple HD cameras are designed that can acquire content at very high overall resolutions, such as 6K or 8K. However, currently most content providers distribute their videos at a lower resolution, such as 4K, due to two main reasons:

- Prohibitively high bitrate requirements, and
- Limited rendering resources at the display side. In particular, the relatively slow speed of the CPU-GPU link limits the resolution of video frames that can be transferred to the GPU for rendering in real-time.

Considering these factors, we aim to create content adaptive representations that minimize distortion under bitrate and sample count constraints.

3.1 Tiling

The equirectangular projection is widely used to generate a planar representation of omnidirectional video due to its simplicity. The constant spacing of latitudes and logitudes leads to a constant vertical sampling density. However, each latitude is stretched horizontally to fit the desired rectangle. This leads to varying horizontal sampling density, which tends to infinity near the poles.

One method to change the sampling density while retaining the rectangular shape is by breaking the equirectangular representation into multiple tiles. The width and height of these tiles can then be adjusted to change the horizontal and vertical sampling densities. More importantly, the introduction of tiles allows us to change the representation based on content specific statistics and user viewing probabilities. More samples can be allocated to tiles (i.e., by increasing the width and height) which contain content with high detail or are more likely to be viewed.

3.2 Joint Sampling-Bitrate Optimization

The generation and adjustment of tiles under bitrate and sample count constraints can be formalized as an optimization problem with multiple constraints. We start with an equirectangular projection of the spherical video, as shown in Fig. 1a. This is split into N tiles (Fig. 1b), where each tile has M options for representation, generated by varying the sizes and coding at different bitrates (Figs. 1c and 1d). We refer to the set of options for each tile as a tile group. Given a bitrate budget R_0 and a sample budget R_1, the objective is to minimize the total distortion subject to these constraints by choosing one option from each tile group. This can be mathematically expressed as,

$$\begin{aligned} \min_{x(i)\in\{1,\cdots,M\},\forall i} \quad & \sum_{i=1}^{N} d(i,x(i)) \\ \text{subject to} \quad & \sum_{i=1}^{N} r(i,x(i),k) \le R_k, \quad k=0,1 \end{aligned} \tag{1}$$

where, $d(i,x(i))$, $r(i,x(i),0)$, $r(i,x(i),1)$ are the distortion, bits used, and samples used for option $x(i)$ in tile group i, respectively. Distortions are calculated independently per tile using L-PSNR.

This optimization problem is a multi-dimensional, multiple-choice knapsack problem. A brute force search which exhaustively evaluates all combinations guarantees an optimal solution. However, the drawback is that brute force search scales poorly since the number of combinations increases exponentially. Thus, the computational complexity is $\mathcal{O}(M^N)$.

3.3 Separation of Constraints

To be able to consider a larger set of tile options, we simplify the optimization problem. We observe that the chosen tile resolutions tend not to change much when changing the target bitrate.

Thus, we propose the following simplified method:

(1) First, we optimize tile sizes considering only the sample count constraint.

(2) Then, using the chosen tile sizes, we code the tiles considering the bitrate constraint.

The tile size determination problem can now be formulated as a standard budget constrained allocation:

$$\begin{aligned} \min_{x(i)\in\{1,\cdots,M\},\forall i} \quad & \sum_{i=1}^{N} d(i,x(i)) \\ \text{subject to} \quad & \sum_{i=1}^{N} r(i,x(i)) \leq R_1 \end{aligned} \tag{2}$$

where, $d(i,x(i))$ is again the distortion introduced when choosing an option $x(i)$ in tile group i which has a resolution $r(i,x(i))$. An approximate solution to this discrete problem can be found using the Lagrangian optimization technique. Specifically, a Lagrange multiplier λ is introduced and the optimization within tile group i is rewritten as,

$$\min_{x(i)\in\{1,\cdots,M\}} \quad d(i,x(i)) + \lambda \cdot r(i,x(i)). \tag{3}$$

The key idea is that, using a common λ for all tile groups, the Lagrangian cost can be minimized separately for each tile group. This results in a problem which scales linearly with the number of tile options, instead of exponentially as in Sec. 3.2. The resulting overall sample count using Lagrangian optimization depends on the choice of λ. However, in practice, we can use heuristics or binary search to find a λ which yields an operating point close to our budget R_1. Note that it is also possible to formulate the joint optimization of (1) as a Lagrangian optimization, however we would need to determine the right vector λ in order to reach an optimum, which is much more difficult than for the scalar case.

Once the tile sizes are determined, ideally the allocation of bits to different tiles would be performed according to Pareto optimality conditions. This would involve encoding each tile with a set of quantization parameters and choosing the best combination across different tiles. However, in order to reduce computational complexity, we employ the same quantization parameter for all the tiles and show that the overall loss w.r.t. exhaustive search of Sec. 3.2 is still very small.

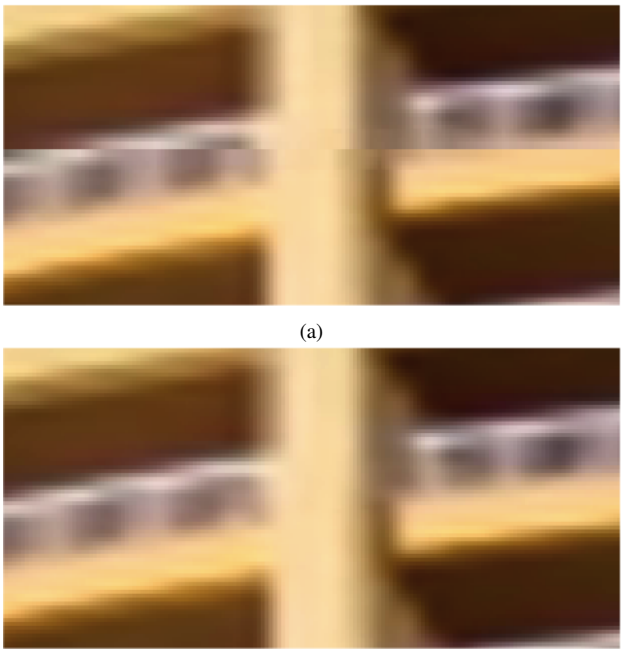
Since this algorithm runs in linear time, we can consider a larger search space than in the brute force method, i.e., more options within each tile group and larger number of tile groups.

3.4 Average Tile Configuration

In this section, we investigate whether the content adaptive scheme proposed in Sec. 3.3 can be approximated by a fixed tiling scheme learned using the entire dataset. Therefore, an average tile configuration within each tile group is computed using the entire dataset. Let the number of items in the dataset be denoted as K, and the width and height resulting from the optimization scheme of Sec. 3.3 for tile j and item k be W_{jk} and H_{jk}, respectively. Then, the average tile configuration for tile group j is calculated as,

$$\widehat{W}_j = \frac{1}{K}\sum_{k=1}^{K} W_{jk}, \quad \widehat{H}_j = \frac{1}{K}\sum_{k=1}^{K} H_{jk}, \tag{4}$$

where, $\widehat{W}_j$ and $\widehat{H}_j$ are the width and height of tile j in the average tile configuration. As in the content adaptive scheme proposed in Sec. 3.3, bit allocation for the average tile configuration is performed by using the same quantization parameter for all tiles.



(a) (b)

Figure 2: (a) Magnified images to illustrate potential boundary effects at the intersection of two tiles. (b) Using overlapping tiles with alpha blending reduces the appearance of seams.

3.5 Overlapping Tiles

Our proposed tiling scheme allows adjacent tiles to have significantly different sampling densities. This difference can sometimes lead to noticeable seams between the tiles. One way to mitigate this effect would be to disallow large changes in adjacent tile dimensions when the optimization is carried out. But this introduces dependencies between the tile groups. Therefore, we propose to generate the tiles such that they overlap with adjacent tiles by a small fraction. Then, alpha blending is used to combine the overlapping regions, resulting in a smoother boundary.

We adjust the optimization schemes proposed in Sec. 3.2 and Sec. 3.3 to account for the top and bottom overlapping regions. Specifically, in the scheme in Sec. 3.2, the bits and samples consumed, $r(i,x(i),0)$ and $r(i,x(i),1)$ in (1), increase due to overlap. Similarly, in the scheme in Sec. 3.3, the resolution of each tile, $r(i,x(i))$, increases. However, the distortion contributed by each tile is still computed using the original tile boundaries to be able to handle each tile independently. The blending of overlapping regions is done outside the optimization loop and can be seen as post-processing.

4. EXPERIMENTAL RESULTS

We evaluate the coding efficiency of the omnidirectional video representation methods described in Sec. 3 on two datasets. The first dataset consists of 10 omnidirectional images in equirectangular format at 6Kx3K resolution depicting indoor and outdoor scenes from the SUN360 database [13]. The second dataset consists of 10 omnidirectional videos in equirectangular format. This consists of 8 videos at 4Kx2K resolution and 2 videos at 6Kx3K[1]. The 4Kx2K videos contain full spherical data, while the 6Kx3K videos do not

[1]The 6Kx3K videos have been generously provided by Jaunt Inc.

Figure 3: Samples frames from the image dataset.

have content at the bottom, where the tripod was placed for recording. The duration and frame rate of each video is 10 seconds and 30 frames per second, respectively. We used a variety of scenes (e.g., concert, outdoor sports, aerial footage, etc.) to cover a wide range of scenarios.

The testing procedure is as follows. First, for each image or video, a constraint on the maximum number of samples is assigned. In all experiments, this constraint is equal to one-fourth the number of samples in the original (e.g., a 6Kx3K video would have a sample count constraint of 4.5 million samples). Second, four bitrate constraints are chosen such that the resulting rate-distortion (RD) curve spans a range between 30-40 dB. Given the maximum number of samples and bitrate constraints, the optimization schemes described in Sec. 3 are run to create content-specific representations. Note that, in the case where the content is video, only the first frame is considered in the optimization schemes to limit the computational complexity and the chosen parameters are used throughout the video. Next, the original data is mapped to this resulting representation and encoded using an H.264/AVC codec. Then, the compressed data is decoded and compared with the original data using L-PSNR. Finally, the resulting RD curves are compared using BD-rate [10].

L-PSNR was recently developed in [12] in order to compare the efficiency of various representations of omnidirectional video. The distortion metric is calculated by sampling points of omnidirectional videos corresponding to a uniform distribution of points on a sphere. The error contributed by each point is then weighted by the viewing probability of the point's latitude. The distortion metric has two main benefits for this application. One, this distortion metric can be calculated even if the original and coded data use different representations and/or resolutions. Two, this distortion metric accounts for user viewing probabilities in HMDs (e.g., users are more likely to look at equator regions than the poles).

4.1 Content Adaptive Representation

Using the image and video datasets introduced, we evaluate the performance of our algorithms with various configurations, namely,

- B10-0: Brute-force optimization using 10 resolution and 4 bitrate choices per tile. No (0%) overlap between adjacent tiles.
- B10-2: Brute-force optimization using 10 resolution and 4 bitrate choices per tile. 2% overlap between adjacent tiles.
- L10-2: Lagrangian optimization using 10 resolution choices per tile. 2% overlap between adjacent tiles.
- L50-2: Lagrangian optimization using 50 resolution choices per tile. 2% overlap between adjacent tiles.
- Avg-2: Average tile configuration. 2% overlap between adjacent tiles.

Seq.	B10-0	B10-2	L10-2	L50-2	Avg-2
0	-9.4%	-5.4%	-3.9%	-9.9%	-7.3%
1	-13.2%	-8.0%	-8.0%	-15.9%	-7.5%
2	-22.9%	-19.7%	-19.7%	-20.9%	-19.9%
3	-5.8%	-2.6%	-1.1%	-5.1%	-3.3%
4	-27.9%	-24.1%	-21.5%	-25.7%	-22.9%
5	-33.7%	-30.8%	-30.0%	-34.6%	-31.8%
6	-23.0%	-20.3%	-17.3%	-22.7%	-20.3%
7	-14.4%	-12.2%	-10.8%	-15.0%	-5.2%
8	-6.2%	-2.7%	-0.8%	-2.0%	-1.0%
9	-30.3%	-27.7%	-27.5%	-28.2%	-27.0%
Avg	-18.7%	-15.4%	-14.0%	-18.0%	-14.6%

Table 1: Results on image dataset. BD-rate comparison of various representations relative to the equal-area representation using the evaluation method described in Sec. 4.

Seq.	L50-2	Avg-2
0	-17.5%	-15.3%
1	1.1%	1.5%
2	-0.9%	1.6%
3	-11.2%	-4.1%
4	-14.5%	-10.0%
5	-9.5%	-8.0%
6	-24.6%	41.0%
7	4.2%	5.2%
8	-17.3%	-15.3%
9	-31.9%	-20.8%
Avg	-12.2%	-2.4%

Table 2: Results on video dataset. BD-rate comparison of various representations relative to the equal-area representation using the evaluation method described in Sec. 4.

The resolution choices ranged between 0.25 to 0.75 times the width and height of the original tiles.

Table 1 summarizes the average bitrate savings of the proposed methods relative to the equal-area representation on the image dataset using the BD-rate measure. While the equirectangular representation is more widely used because of its simplicity, recent work has shown that the equal-area representation generally performs better [12]. Negative BD-rate numbers indicate bitrate savings w.r.t. this baseline.

As seen in Table 1, both our brute-force and Lagrangian optimization schemes significantly outperform the baseline across all configurations. The brute-force optimization scheme with no overlap (B10-0) achieves a bitrate savings of 18.7%. Next, considering the requirement of tile overlap, the loss in coding efficiency can also be seen in Table 1. Specifically, the average gains from using our brute-force optimization scheme drops from 18.7% to 15.4% when considering tiles with 2% overlap (B10-2). Furthermore, Table 1 shows the loss in coding efficiency when using Lagrangian optimization rather than brute force search. The average gains drop from 15.4% to 14.0% when using the Lagrangian scheme (L10-2). While this loss is not negligible, the Lagrangian optimization scheme is significantly more computationally efficient. This allows more tile options to be considered, yielding larger gains than using

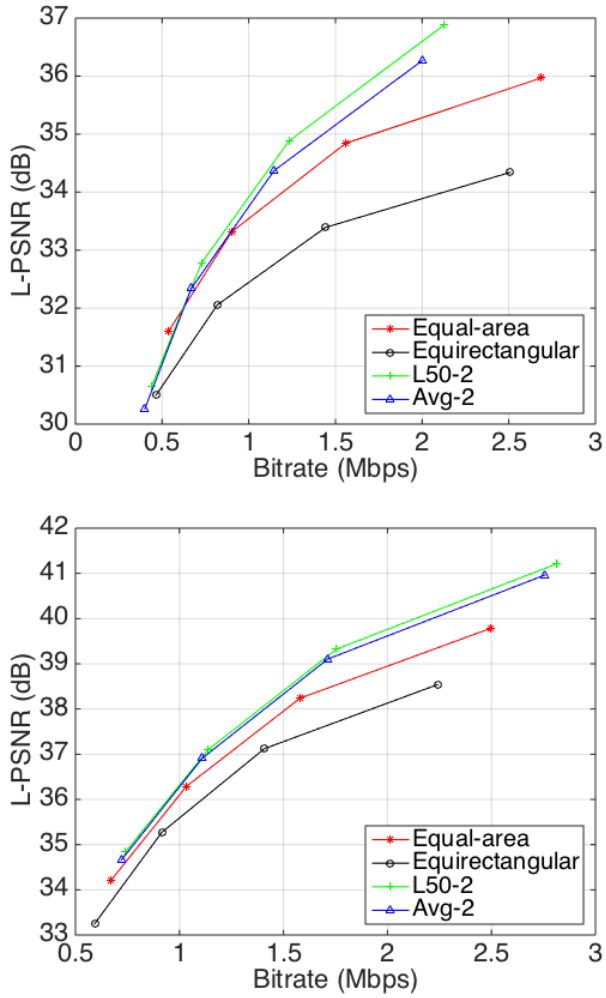


Figure 4: RD curves of two video sequences for different panoramic projections.

the brute-force scheme with fewer options. In particular, our Lagrangian optimization scheme with 50 resolution choices and 2% overlap (L50-2) has average bitrate savings of 18.0% on the image dataset compared to the baseline equal-area representation.

Table 2 summarizes the average bitrate savings on the video dataset. Here, we consider the best performing method on the image dataset with 2% overlap (L50-2) and compare it to the reference equal-area representation. The proposed method achieves average bitrate savings of 12.2% and max savings of 31.9%. It can be seen that the average bitrate savings on the video dataset is lower than on the image dataset. This can be due to the fact that the optimized representation of the first frame is used throughout the video in our experiments. This can be easily extended such that the optimization is repeated after a desired interval.

Fig. 4 shows the RD curves for two videos in our dataset, namely Seq. 4 and 5. These curves show the difference in coding efficiency between using traditional representations (i.e., equirectangular, equal-area) and our proposed methods. In particular, the popular equirectangular representation performs significantly worse than other representations. Moreover, our proposed method using Lagrangian optimization with 50 resolutions and 2% overlap outperforms all other representations.

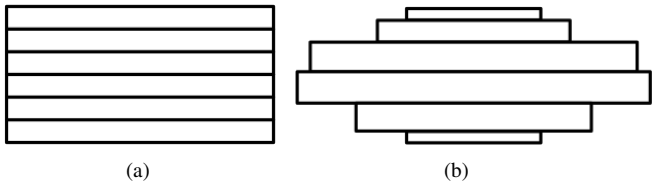


Figure 5: Relative tile sizes when using (a) the equirectangular representaiton broken into 6 horizontal tiles and (b) the average tile configuration trained on the image dataset.

4.2 Average Tile Configuration

An average tile configuration was computed separately for the image and video datasets. Fig. 5 shows the relative difference in tile sizes between the tiled equirectangular representation and the average tile configuration computed using the image dataset. Note that the illustrated tiles are drawn to scale. In the average tile configuration, the top and bottom tiles, which correspond to the north and south poles of the omnidirectional video respectively, are significantly smaller in both width and height. This large reduction in width is expected since the horizontal sampling density of the equirectangular projection is higher near the poles. The similar reduction in height is due to content typically containing less detail near the poles (e.g., sky, ceilings) than in the equatorial regions. The larger equatorial tiles also match the observation that users tend to view these areas most frequently.

For the image dataset, the coding efficiency of the average tile configuration is generally similar to the performance of our content adaptive schemes. However, the average tile configuration shows large variation in bitrate savings across the dataset. For instance, on Seq. 7 in Table 1, the average tile scheme performs 10% worse than our best content-adaptive scheme. The average savings for the video dataset in Table 2 is rather low because of Seq. 6, which is an outlier. Thus, while the average tiling configuration can generally perform better than the baseline, there exists diverse content which cannot be handled by such a fixed scheme. Hence, content adaptivity is crucial in enabling efficient representations of omnidirectional videos.

5. CONCLUSION

In this paper, we have developed an algorithm which significantly improves the coding of omnidirectional videos by optimizing the underlying representation subject to resource constraints. We showed that the joint optimization of sampling and bit allocation can be formulated as a multi-dimensional, multiple-choice knapsack problem which is solvable with brute force search. By separating the constraints on sampling and bit allocation, we also showed that the optimization problem can be solved efficiently using a Lagrangian framework. Average bitrate savings of over 18% and 12% relative to the baseline equal-area representation were observed on an image and video dataset, respectively. While a generic representation can often perform well, this work shows that a content adaptive representation can be beneficial due to the diverse nature of omnidirectional content.

Acknowledgment

The authors wish to thank Dr. Peter Vajda for valuable discussions and comments.

6. REFERENCES

[1] J. P. Snyder, “Flattening the Earth: Two Thousand Years of Map Projections,” University of Chicago Press, 1993.

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

[3] R. Carroll, M. Agrawala, and A. Agarwala, “Optimizing Content-Preserving Projections for Wide-Angle Images,” In Proc. SIGGRAPH, 2009.

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

[5] K. Ng, S. Chan, and H. Shum, “Data compression and transmission aspects of panoramic videos,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 1, January 2005.

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

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

[8] 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.

[9] 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.

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

[11] A. Ortega, K. Ramchandran, “Rate-distortion methods for image and video compression," Signal Processing Magazine, IEEE , vol.15, no.6, pp.23–50, November, 1998.

[12] M. Yu, H. Lakshman, B. Girod, “A Framework to Evaluate Omnidirectional Video Coding Schemes," IEEE International Symposium on Mixed and Augmented Reality, 2015.

[13] J. Xiao, K. A. Ehinger, A. Oliva and A. Torralba. “Recognizing Scene Viewpoint using Panoramic Place Representation" IEEE Conference on Computer Vision and Pattern Recognition, 2012.