Delivering 360-degree video with Viewport-adaptive Truncation

Tian Qiu, Department of ECE, UC San Diego, La Jolla, USA, tiq014@eng.ucsd.edu

Ish Kumar Jain, Department of ECE, UC San Diego, La Jolla, USA, ikjain@eng.ucsd.edu

Raini Wu, Department of ECE, UC San Diego, La Jolla, USA, rainiwu@ucsd.edu

Dinesh Bharadia, Department of ECE, UC San Diego, La Jolla, USA, dineshb@eng.ucsd.edu

Pamela Cosman, Department of ECE, UC San Diego, La Jolla, USA, pcosman@eng.ucsd.edu

***Abstract*—Delivering Virtual Reality (VR) content wirelessly involves projecting a 360-video into a 2D format and then encoding it to satisfy the wireless bitrate requirements. However, the popular equirectangular and cubemap projections offer little flexibility to adapt to changing bitrates and headset motion. In this work, we show that the truncated square pyramid projection offers high flexibility for network and headset motion adaptation. We adapt by tuning a truncation parameter that controls the video quality for different spatial regions in the 360-video. Depending on the video, our scheme improves average video quality by up to 1.1dB in PSNR and up to 4.6 in VMAF score compared to a non-adaptive baseline.**

***Index Terms*—Video coding, 360-video, projection**

I. INTRODUCTION

Virtual reality (VR) and augmented reality (AR) deliver immersive panoramic 360° video experiences. Due to the large bandwidth requirement [1], efficient 360° video encoding and streaming are required for delivering 360° videos wirelessly.

At any given time, a user has a limited field of view, looking at a region known as the viewport, which typically occupies only 15% of the original 360° video [2], [3]. Transmitting only the viewport region instead of the whole video can yield high-quality VR streaming while satisfying stringent wireless network constraints. However, with this naive strategy, real users will see blank screens as they move their heads and gaze in different directions. With head motion prediction, the system can fetch the new viewport in advance, so the user enjoys a seamless experience. This requires predicting user head motion over the 1-2sec needed to fetch a viewport from the server. Within this time, users can move in complex ways, so prediction accuracy is ≈58-80% [4], [5]. Thus, new 360° video streaming solutions, beyond viewport-adaptive streaming [5] are needed to provide robustness to viewport prediction error while saving bandwidth.

We present VRProj, a system that leverages the efficient truncated square pyramid (TSP) projection format for 360° video. VRProj differs from existing TSP-based 360° video streaming solutions by introducing flexible truncation and location switching to adapt to user head motion.

978-1-6654-7318-7/22/$31.00 ©2022 IEEE

The paper is organized as follows. In Section II, the problem background and design intuition are presented. In Section III, we describe the detailed design of the system. We discuss evaluation results and provide conclusions in Section IV.

II. BACKGROUND AND MOTIVATION

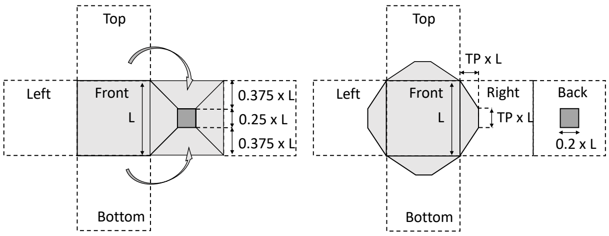
A 360° video represents a scene that covers an entire 3D sphere but is captured in multiple 2D videos and stitched into a 2D video which can be efficiently encoded by traditional 2D codecs, e.g., H.264/AVC [6], H.265/HEVC [7]. We discuss three projection methods: equirectangular projection (ERP), cubemap projection (CMP), and pyramid projections [8], [9]. Fig. 1 shows 360° video in various projections. While popular for its holistic view of the omnidirectional content, ERP inefficiently maps a 3D sphere to an unwrapped 2D plane. This results in inefficiencies which increase farther from the center of the video. CMP mitigates inefficiency by mapping the 3D spherical content to a 2D unwrapped cube; 3D spherical content is divided into six regions, one for each cube face. Inefficiencies occur along the face boundaries, a great improvement over ERP. CMP is popular with commercial video providers such as YouTube [10].

To stream 360° video over a varying channel, tiling-based systems are popular [4], [11]–[24]. These methods divide a video into smaller rectangular tiles. To adapt to varying network conditions, video tiles are encoded to various quality levels, and only the tiles in and around the predicted viewport location are streamed with high quality. The remaining tiles are streamed with low quality. Current tiling-based systems rely on ERP projection due to its simplicity.

An alternative approach to achieve viewport-adaptive quality is to process the omnidirectional content into a format where only a central viewport region has the highest quality. We focus on pyramid projection [25], a representative viewport-adaptive projection. Unlike ERP and CMP, in pyramid projection, only a section of the 3D spherical content is preserved at high quality, and other sections continuously decrease in quality. This is achieved by mapping the high-quality section to the base of a 2D unwrapped pyramid, with other sections as the pyramid sides. Pyramid projection is *intrinsically* viewport-adaptive; content not in the high-quality

(a) Equirectangular (b) Cubemap (c) TSP (TP=0.2) (d) TSP (TP=0.4)

Fig. 1: 360^o video is projected from a sphere to a 2D format in different projections.



(a) JVET-TSP construction (b) Adaptive TSP construction

Fig. 2: TSP can be constructed with different levels of flexibility.

viewport section is represented with low quality. While allowing for viewport adaptivity through projection, the pyramid lacks flexibility. For a given high-quality region size, geometric limitations remove control over the quality of the other regions. The truncated square pyramid (TSP) projection [26] improves on this aspect.

TSP projection may be understood as a modified CMP. Instead of allocating an equal number of pixels to every cube face, a single region is favored over the others. This results in an intrinsic viewport-adaptive quality while also allowing changes to the quality of non-favored regions. The *location parameter* (LP) adjusts the location of the favored region, while the *truncation parameter* (TP) changes the degree of favoritism. The TSP projection allows viewport-adaptive projection direction changes while also allowing for fine-grained control over the quality of non-favored regions using TP.

Prior work on viewport-adaptive TSP projections [26], [27] considers switching only the location parameter for viewport adaptation; the flexibility introduced by a truncation parameter was not explored. Fig. 2(a) illustrates the TSP configuration in a proposal to JVET [26] ("JVET-TSP"). The side faces are truncated in a fixed way. Fine-grained control over the quality of non-favored regions through adapting the truncation provides additional opportunities for optimization. VRProj designs a novel algorithm to find the best truncation and location parameters adapted to viewport location and motion.

III. Design of VRProj

A TSP projection, constructed from a CMP projection, inherits the six CMP faces: the front face, side faces (left, right, top, bottom), and back face as shown in Fig. 2(b). The construction requires two parameters: LP determines the location of the front face in a sphere, and TP determines the extent of truncation. For simplicity, we focus on variations along the yaw axis[1]. LP $\in [-180°, 180°]$ must be decided before projecting a sphere on a cube such that the LP lies at the center of one face of the projected cube; this is the front face of the resulting CMP frame. We then truncate the CMP side faces into trapezoids according to the TP value; given cube face edge length L and TP$=\alpha$, both the trapezoid's height and shorter parallel side will have length αL. These transformations use uniform subsampling [29]. We obtain the back face of TSP by 5X downsampling the cubic back face.

TSP provides a mechanism to adapt to a moving viewport. Changing LP allows coarse adaptation to the viewport location, while TP allows finer adaptation. Fig. 3 illustrates the mechanism for the case of 4 LPs available $(0°, 90°, 180°, -90°)$. With the initial viewport located at $0°$, i.e., centered at the TSP video's front face, we choose a low TP=0.2 to heavily truncate the video side faces. As the viewport moves towards $45°$, it significantly overlaps the side face, which would limit the quality if a low TP were used. So, we increase TP to 0.4. Next, as the viewport crosses $45°$, a large piece of the viewport lies in the right side face. So we change LP to $90°$ to make the right face become the new front face, as shown in the fourth case of Fig. 3.

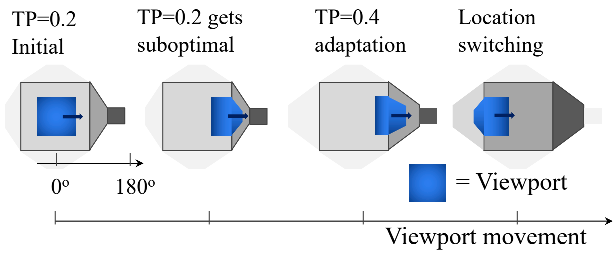


Fig. 3: TSP viewport adaptation with 4 LPs and 2 TPs.

A. Optimizing Location and Truncation Parameter

A bitrate target can be met by using TP to control an adaptive projection, or using the quantization parameter (QP) to control the quantization of a uniform encoding. A higher TP results in a raw video that can then be heavily quantized (high QP) to satisfy the bitrate constraint. Alternatively, the same

[1] A user is more likely to move her head horizontally than vertically [28].

constraint could be met with a lower TP producing a smaller raw video which then can benefit from low quantization errors (low QP), resulting in high quality for the viewport region. There is a tradeoff in selecting specific TP and QP values.

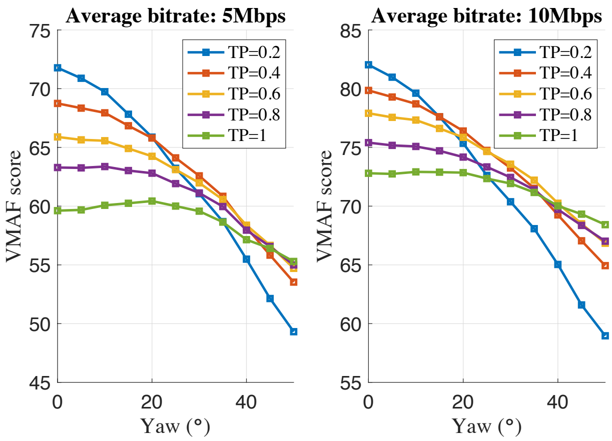


Fig. 4: VMAF versus yaw angle for different TP values (video encoded at 5 and 10 Mbps). We use this plot to formulate the mapping of viewport orientation to TP.

We analyze the TP-QP tradeoff in Fig. 4. To evaluate perceptual quality, we use Video Multi-Method Assessment Fusion (VMAF) [30], a machine learning-based metric proposed by Netflix. It ranges from 0 to 100, with 100 denoting perfect perceptual quality compared to the reference video. Fixing the TSP video's front face to $0°$ and truncating the left and right side faces with TP=0.2, 0.4, 0.6, 0.8 and 1, we extract the viewport at various static locations (yaw angles from $0°$ to $45°$). From Fig. 4, we observe that TP=0.2 provides the best VMAF when the viewport is aligned with the front face because heavy truncation of side faces allows a lower QP value, thus higher front face quality. However, when TP=0.2, VMAF degrades sharply as the viewport moves away from the front face. With higher TP values, VMAF degrades more gradually with yaw angle. In comparing TP of 0.4 and TP of 0.2, the VMAF score for TP=0.4 is lower by 3 (out of 100) for the front face, the scores are about equal at roughly $10°$ of viewport motion to the right, and beyond $15°$, TP=0.4 is consistently better. As TP above 0.4 does not provide useful trade-offs, our system uses only TP=0.2 and TP=0.4.

This mapping of viewport location to TP value is based on empirical evaluation of a $360°$ video. It can be done off-line at the server once and stored for different videos and bitrates, causing no overhead to real-time streaming. We observe through evaluation on other eight videos that the mapping is resilient to video content and to two average encoding rates (5, 10Mbps). The TP value can be selected based on the predicted viewport location by a table look-up in real time.

Another observation (Fig. 4) is that truncation sub-sampling tends to dominate the quality loss in the viewport as the viewport moves away from the front face. This is observed in other videos and at higher bit rates. This can be mitigated

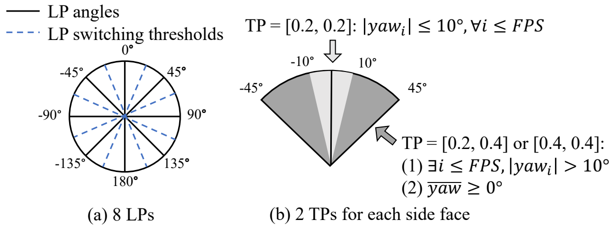


Fig. 5: VRProj's decision rules for LP and TP selection. FPS denotes the set of frames of the next video chunk.

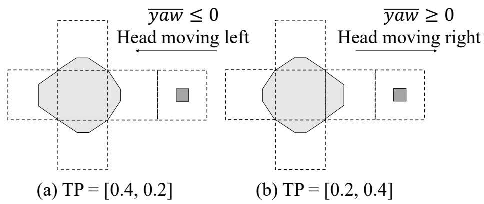


Fig. 6: Adaptive TP with head movement when TP is not [0.2, 0.2].

with a finer granularity of LP, as the viewport after location switching is less likely to reach regions heavily distorted by truncation. We use 8 LPs spaced $45°$ apart.

B. Awareness of head movement direction

Exploiting head movement direction can further reduce the file size by assigning different TPs to left and right faces. If the head is moving to one side, a higher TP could be assigned to this side. Our adaptation is summarized in Fig. 5. The front face of the next chunk is rotated by LP such that the mean value of the predicted viewport lies within $\pm 22.5°$. Then, we consider four TP combinations: [0.4, 0.4], [0.4, 0.2], [0.2, 0.4], and [0.2, 0.2]. We first decide if the head motion is static enough to use TP of 0.2 on both side faces. Using Fig. 4, we choose $\pm 10°$ as our TP decision boundary. If the viewport is predicted to be within $\pm 10°$ for every frame in the next video chunk, TP of [0.2, 0.2] will be assigned. When this is not satisfied, TP of [0.2, 0.4] or [0.4, 0.2] will be considered, depending on $\overline{yaw}$, the mean predicted viewport yaw angle over all frames in the chunk. We consider that the head is moving to the right (left) if the actual yaw angle is larger (smaller) for the last frame of the previous video chunk than for the first frame of that chunk. TP of [0.2, 0.4] will be assigned when the head is moving right and $\overline{yaw} \geq 0$; TP of [0.4, 0.2] will be assigned when the head is moving left and $\overline{yaw} \leq 0$ (see Fig. 6). Lastly, when none of the above is satisfied, TP of [0.4, 0.4] is used since we are not confident that the future viewport will be static nor that it will be on a particular side.

IV. Evaluation

We evaluate VRProj with a head movement trace-driven simulation with fixed network conditions. The key results are that when the viewport information is assumed to be perfectly known, VRProj on average has 2 (out of 100) of VMAF

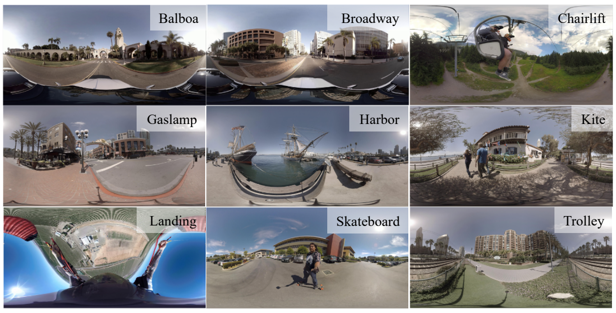


Fig. 7: Examples of video frames from our 360° video dataset.

score improvement over the JVET-TSP system. With a simple viewport prediction, the improvement in VMAF is 1.5.

Video Dataset: We use nine 360° videos provided by ITU-T [31]. Captured on various platforms while driving, walking, or skateboarding, the content includes sightseeing, nature, and sports. The videos are in uncompressed YUV format, with sizes from 10 to 30 GB. All videos have 8192x406 resolution and 60 FPS, except for "Balboa" and "Broadway" (6144x3072 60FPS), and "Chairlift" (8192x4096 30FPS). Sample frames are in Fig. 7. The dataset is in ERP format, which we convert to CMP using 360Lib [32].

User head movement dataset: We evaluate VRProj on a dataset of user head movement traces from [33], with a large variety of head motions from 0°/sec to 60°/sec. For each video, we assign one trace from the dataset as if it were the head movement of users watching that video.

End-to-end results: We implement VRProj in approximately 1000 lines of C++ and Matlab code. We emulate DASH-based [34] video streaming for both VRProj and the JVET-TSP system; a collection of TSP video is generated offline and adaptively fetched by the client during streaming. The server stores video chunks encoded at different rates. Videos are encoded with FFmpeg x265 codec with 2-pass average bitrate control. JVET-TSP stores video at 8 different LPs. For VRProj, each video is stored in 8 different LPs and 4 combinations of TPs. We evaluate VRProj at 5 and 10 Mbps in two modes, omniscient (Omni) and predictive (Pred). Fig. 8 provides an overview. System parameters (LPs and TPs) are decided each second. With Omni, parameters are based on the user's actual viewport orientation in the future. With Pred, parameters are decided from linear regression based viewport prediction. Fig. 8(a) demonstrates the viewport prediction in Pred mode. Fig. 8(b) plots the LPs assigned for each chunk with respect to the front face orientation in the original dataset. Fig. 8(c) and (d) plot the TPs chosen for each video chunk. Fig. 8(e) plots the final viewport direction the user will experience after switching the front face orientation according to LP.

VRProj outperforms JVET-TSP in both Omni and Pred modes, as shown in Table I. In Omni, the gain of VRProj over JVET-TSP is greater since LP and TP adaptation leverages the knowledge of head movement. VMAF has a strong linear correlation with subjective Differential Mean Opinion Score

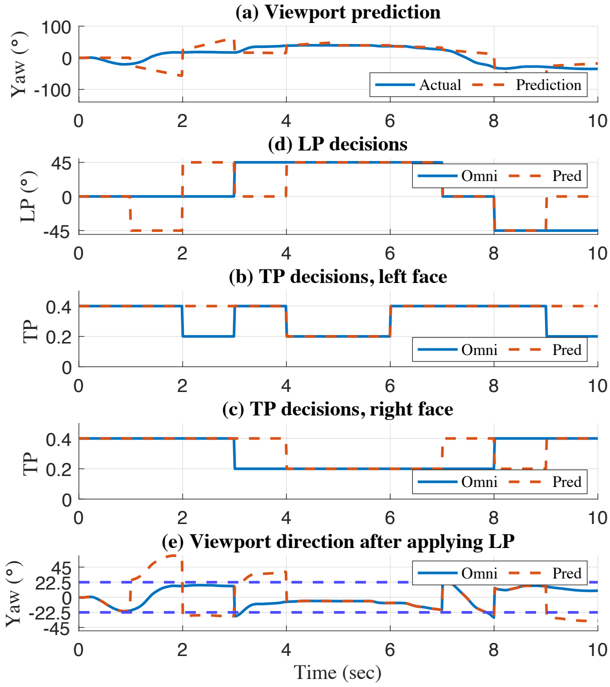


Fig. 8: VRProj adapts LPs and TPs to head movement. "Omni" and "Pred" stand for omniscient and predictive simulation.

Video	5Mbps		10Mbps	
	Omni	Pred	Omni	Pred
Balboa	1.33	1.06	1.04	1.04
Broadway	1.71	1.57	0.92	0.88
Chairlift	1.91	1.28	1.26	0.57
Gaslamp	0.88	0.20	0.42	-0.13
Harbor	1.03	0.16	0.72	-0.19
Kite	2.96	2.46	2.28	1.76
Landing	1.73	1.34	1.50	1.04
Skateboard	2.30	1.54	2.10	1.51
Trolley	4.57	4.11	2.81	2.57
Average	2.05	1.52	1.45	1.01

TABLE I: VMAF gains of VRProj over JVET-TSP.

[30], [35], so this improvement indicates that VRProj can provide better perceptual quality. VRProj also outperforms JVET-TSP in peak signal-to-noise ratio (PSNR), with average gains of 0.39dB (Omni) and 0.24dB (Pred) at 5Mbps, and 0.44dB and 0.21dB at 10Mbps. PSNR gains for individual videos ranged from 0.2dB for Balboa to 1.1dB for Trolley.

V. Conclusion

We presented an adaptive TSP projection scheme that optimizes the degree of truncation and quantization to improve video quality for a given headset motion. Both VMAF and PSNR results indicate that the gains can be significant for certain videos, while some improvement was found for all 9 videos tested. Future work could explore content-based adaptivity in the TSP system, or adding head motion adaptivity to tiling-based [4], [11]–[24] approaches.

REFERENCES

[1] M. Zink, R. Sitaraman, and K. Nahrstedt, "Scalable 360 video stream delivery: Challenges, solutions, and opportunities," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 639–650, 2019.

[2] S. Afzal, J. Chen, and K. Ramakrishnan, "Characterization of 360-degree videos," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 1–6.

[3] Y. Bao, H. Wu, A. A. Ramli, B. Wang, and X. Liu, "Viewing 360 degree videos: Motion prediction and bandwidth optimization," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 2016, pp. 1–2.

[4] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[5] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1977–1986.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[7] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[8] Z. Chen, Y. Li, and Y. Zhang, "Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation," *Signal Processing*, vol. 146, pp. 66–78, 2018.

[9] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *2017 IEEE international conference on communications (ICC)*. IEEE, 2017, pp. 1–7.

[10] "Bringing pixels front and center in VR video," https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/, Mar 2017.

[11] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: Optimizing 360 video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 394–407.

[12] C. Madarasingha and K. Thilakarathna, "Vastile: Viewport adaptive scalable 360-degree video frame tiling," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4555–4563.

[13] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Rubiks: Practical 360-degree streaming for smartphones," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 482–494.

[14] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 1–6.

[15] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 315–323.

[16] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1819–1831, 2016.

[17] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 261–271.

[18] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An http/2-based adaptive streaming framework for 360 virtual reality videos," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 306–314.

[19] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 601–605.

[20] M. Xiao, C. Zhou, V. Swaminathan, Y. Liu, and S. Chen, "Bas-360: Exploring spatial and temporal adaptability in 360-degree videos over http/2," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 953–961.

[21] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360º video streaming," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[22] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2174–2178.

[23] P. R. Alface, J.-F. Macq, and N. Verzijp, "Interactive omnidirectional video delivery: A bandwidth-effective approach," *Bell Labs Technical Journal*, vol. 16, no. 4, pp. 135–147, 2012.

[24] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "Optile: Toward optimal tiling in 360-degree video streaming," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 708–716.

[25] "Next-generation video encoding techniques for 360 video and VR," https://engineering.fb.com/2016/01/21/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/, Feb 2022.

[26] G. Van der Auwera, H. M. Coban, and M. Karczewicz, "Truncated square pyramid projection (tsp) for 360 video, jvet doc," *D0071*, 2016.

[27] A. Zare, A. Aminlou, and M. M. Hannuksela, "Virtual reality content streaming: Viewport-dependent projection and tile-based techniques," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1432–1436.

[28] S. Afzal, J. Chen, and K. Ramakrishnan, "Viewing the 360° future: Trade-off between user field-of-view prediction, network bandwidth, and delay," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2020, pp. 1–11.

[29] G. Van der Auwera, M. Coban, and M. Karczewicz, "Truncated square pyramid geometry and frame packing structure for representing virtual reality video content," Jun. 11 2019, uS Patent 10,319,071.

[30] "VMAF - Video Multi-Method Assessment Fusion," https://github.com/Netflix/vmaf, Feb 2022.

[31] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "Jvet common test conditions and evaluation procedures for 360 video," *Joint Video Exploration Team of ITU-T SG*, vol. 16, 2017.

[32] "Algorithm descriptions of projection format conversion and video quality metrics in 360Lib Version 5," https://mpeg.chiariglione.org/tags/jvet, Jan 2021.

[33] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. McMahan, R. Prakash, M. C. Q. Farias, and M. M. Carvalho, "A taxonomy and dataset for 360° videos," in *Proceedings of the 10th ACM Multimedia Systems Conference*, ser. MMSys '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 273–278. [Online]. Available: https://doi.org/10.1145/3304109.3325812

[34] "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats," https://www.iso.org/standard/79329.html, Feb 2022.

[35] R. Rassool, "Vmaf reproducibility: Validating a perceptual practical video quality metric," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2017, pp. 1–2.