Performance Measurements of 360° Video Streaming to Head-Mounted Displays Over Live 4G Cellular Networks

Wen-Chih Lo, Ching-Ling Fan, Shou-Cheng Yen, and Cheng-Hsin Hsu
Department of Computer Science
National Tsing Hua University
Hsin-Chu, Taiwan

***Abstract*—Watching 360° videos using Head-Mounted Display (HMD) allows users to only see a part of the whole 360° videos. With this feature, tiled videos become a potential solution for aggressively reducing the required bandwidth for 360° video streaming, turning it into a reality in cellular networks. In this paper, we design several experiments for quantifying the performance of tile-based 360° video streaming over a real cellular network on our campus. In particular, we empirically investigate the impacts of tile streaming over 4G networks, such as coding efficiency, bandwidth saving, and scalability. Our experiments lead to interesting findings, for example, (i) only streaming the tiles viewed by the viewer achieves bitrate reduction by up to 80% and (ii) the coding efficiency of 3x3 tiled videos may be higher than non-tiled videos at higher bitrates. We believe this work will stimulate more studies in the emerging area of mobile AR/VR (Augmented Reality and Virtual Reality) over 4G networks.**

***Index Terms*—360° videos, streaming, cellular networks, experiments**

I. Introduction

Using traditional planar televisions or monitors to watch a live broadcast of an event, such as a football match or a music concert is a *passive* experience. Over the past years, Virtual Reality (VR) products, such as Head-Mounted Displays (HMDs), become widely available. Many companies release their HMDs, such as Oculus Rift DK2 [1], HTC Vive [2], and Samsung Gear VR [3]. These products offer viewers wider Field-of-Views (FoVs) and provide more immersive experience than traditional televisions and monitors. Besides, lots of 360° cameras also hit the market. For instance, Ricoh Theta S [4], Luna 360 VR [5], and Samsung Gear 360 [6]. With the growing popularity of commercial VR products, more viewers are able to watch 360° videos with HMDs. On top of that, major multimedia streaming service providers, such as Facebook [7] and YouTube [8] now support 360° video streaming for VR content.

Majority of present global Internet traffic is due to video data [9]. Streaming 360° videos further increases the Internet traffic amount, and becomes a hot research topic. In particular, streaming videos coded in 4K, 8K, or higher resolutions lead to insufficient bandwidth and overloaded decoder. As shown in Fig. 1, when watching 360° videos, a viewer wearing an HMD rotates his/her head to *actively change* the viewing orientation. Viewing orientation can be described by roll, pitch, and yaw, which correspond to rotates along x, y, and z axes. Based on the viewer's orientation, the HMD displays the current FoV, which is a fixed-size region about $100° \times 100°$ circle. A viewer with HMD only gets to see a small part of the whole video, and thus streaming the whole 360° videos is unnecessary.

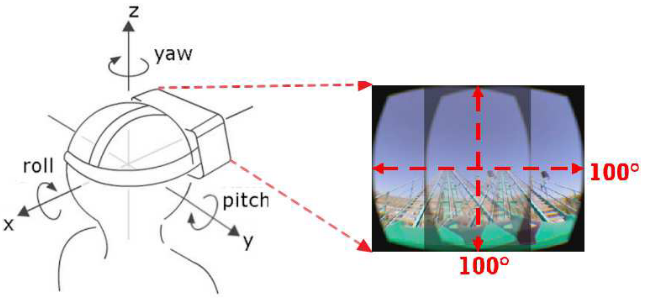


Fig. 1. A viewer watches a 360° video with an HMD. He/She rotates his/her head to change the viewing orientation and only gets to see a small part (about 100°x100° circle) of the whole video.

To save network bandwidth, a 360° video may be split into tiles of subvideos, which are encoded by modern video codecs, such as HEVC (High Efficiency Video Coding) into video bitstreams. The video streams are then streamed using MPEG Dynamic Adaptive Streaming over HTTP (DASH) [10], an adaptive streaming technology for delivering videos over the Internet. In DASH streaming systems, each tile is encoded into multiple versions at different bitrates. This provides the ability for a client to switch among different bitrates or quality levels based on current network conditions. Having multiple tiles allows each client to only request and decode those tiles that will be watched by viewers, in order to conserve resources. However, splitting a video into tiles may reduce the coding efficiency and increase the bandwidth consumption, compared to a single non-tiled video. Therefore, how to optimally split a video into tiles is a challenging issue.

In this paper, we describe and evaluate the performance of tile-based adaptive streaming over a real 4G cellular network. We aim to to answer the following questions:

978-1-5386-1101-2/17/$31.00 ©2017 IEEE

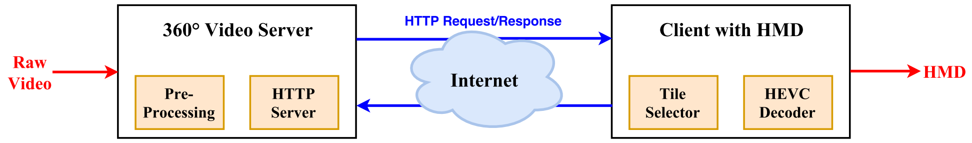


Fig. 2. Overview of our tile-based 360° video streaming platform.

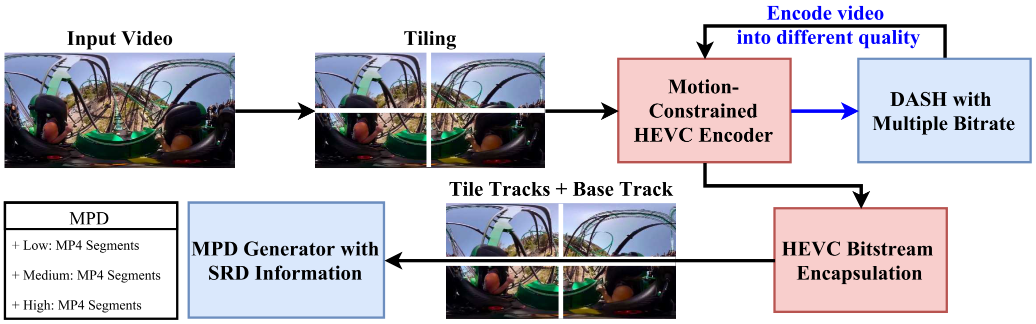


Fig. 3. The pre-processing procedure of our 360° video server.

- **How tile size affects the streaming system performance?**
- **How much bandwidth saving can we get by selectively requesting useful tiles?**
- **How many users can be supported in one 4G cell?**

The rest of this paper is organized as follows. We review the related work in Sec. II. This is followed by some background on HEVC and DASH in Sec. III. The testbed is presented in Sec. IV. We design the measurement study in Sec. V. We report the experiment results to answer the aforementioned questions in Sec. VI. Sec. VII concludes the paper.

II. Related Work

360° video streaming. Both real-time transcoding [11], [12] and tile streaming [13], [14], [15], [16] can be used to stream 360° videos. Compared to real-time transcoding, tile streaming has higher scalability, thus we consider tile streaming approach in the following. Gaddam et al. [17] propose a tiled video streaming system for panoramic videos. They propose to save the consumed bandwidth by gradually reducing the quality level from the center of the viewable region to outward. Spatial Relationship Description (SRD) [18], is an extension of DASH allowing the client to request only a part of videos. Several recent studies [15], [16] adopt SRD for tiled video streaming because of its higher flexibility. Zhou et al. [19] conduct experiments to measure the performance of offset cubic projection, which is implemented in Oculus HMD streaming system. Graf et al. [20] investigate the usage and measure the performance of different tiling strategies. El-Ganainy [21] proposes a new projection model, which introduces fewer redundant pixels compared to equirectangular projection. He further designs a rate adaptation algorithm aiming to stream all the viewed tiles with the same high quality. Ghosh et al. [22] consider QoE metrics as a function of the bitrate of tiles on the whole 360° video or on the tiles within the known FoV. They propose heuristic algorithms aiming to maximize the considered QoE metrics. The numerical results of the impacts of different factors, such as the weights of tiles within FoV and stall time, are reported as well. However, all these studies stream the 360° videos over Ethernet instead of the wireless networks, which are more vulnerable to network dynamics.

360° videos over wireless networks. Zoomability is an advantage of 360° videos. Wang et al. [14] study the problem of streaming zoomable videos using wireless multicasting, which is challenging due to heterogeneous resolutions, FoVs, and bandwidth requirements among viewers. Instead of encoding tiles with the fixed resolution at each layer, they propose to encode mixed-resolution tiles and study the acceptable quality drops of viewers. They further model the optimal resolution allocation problem by considering the available time slots for each tile. Chen et al. [23] propose to capture QoS of VR users in small cell networks. They conduct simulation and the results show that the QoS of VR users depend on performance tracking and wireless resource allocation. Qian et al. [24] study the transport issue of 360° videos by first carrying out a network-centric measurement study on two popular 360° video streaming services: YouTube and Facebook. Their subjective tests reveal some insights and several challenges when using the services over cellular networks. However, their experiments are conducted by simulations instead of through real 4G/cellular networks.

III. Background

Our 360° video streaming system consists two major components: a 360° video server and a client with HMD. Fig. 2 gives an overview on our system.

A. 360° Video Server

360° Video Server contains a pre-processing component (including an HEVC encoder and an MPEG DASH content generator) and an Apache HTTP server. Each video is spilt into tiles, and encoded using HEVC. The coded videos are then streamed independently using DASH with SRD, as shown in Fig. 3. Encoding a tiled video using HEVC encoder is unique. We summarize the key differences below.

Motion constrained. In motion prediction, a tile could refer to data of another tile in a previous or future reference frames, leading to decoding glitches. Therefore, Feldmann et al. [25] propose to constrain the tiles encoding so that each tile only refers to the same tiles in previous or future frames. This reduces the complexity on the clients and in the networks.

In-loop Deblocking Filter. HEVC uses an In-loop Deblocking Filter (DBF) and a Simple Adaptive Offset Filter (SAO) to improve the decoded video quality. Because of the motion-constrained encoding, the filtering operation may miss some information from the neighboring tiles, which leads to negative side effects. Therefore, the in-loop filter needs to be disabled at the border of tiles to avoid the artifacts. This unfortunately may cause visual discontinuities or blocking effects.

B. Adaptive Streaming over HTTP

We integrate our system with DASH. Two main features enable DASH streaming of 360° videos: *Media Presentation Description* (MPD) and *Spatial Representation Description* (SRD) [18], which are detailed below.

MPD is an Extensible Markup Language (XML) document that describes an adaptive streaming session. It contains the media segment information, such as timestamp, URL, video resolution, bitrates, and bandwidth restrictions. We use GPAC/MP4Box [26], an open-source project to package our raw HEVC bitstream and generate the MPD document.

SRD describes the relationship among tiles, which is a feature extending the MPD of MPEG DASH. It provides additional information to further help DASH clients on determining which tiles to request. SRD puts the media content in a 2D coordinate system, providing the x-axis, y-axis, width, and height attributes. SRD only describes how the content is spatially organized, but does not presume anything on how a DASH client uses this information. This enables the DASH clients to freely select and display only those tiles that are relevant.

C. Client with HMD

The client of our system contains two components: Tile Selector and HEVC Decoder. We leverage an open-source project, GPAC/MP4Client [27], to be our 360° video player. It is the only tile player publicly available at the time of writing.

Tile selector selects the useful tiles to download. With HMD, each viewer only gets to see a small part of the whole 360° video. Therefore, streaming the whole 360° video in the full resolution may lead to wasted bandwidth. We leverage a crowd-sourced 360° viewing dataset [28] to select and download only those tiles that fall in the viewers FoV.

A single **HEVC decoder** is used to decode the tiles that are received and merged first. Watching 360° videos with 4K UHD resolution or higher is quite challenging because of hardware limitations (slow CPUs or single decoder chip) on some resource-constrained end devices. The tiled video content is encapsulated into a single HEVC bitstream, and thus only requires a single standard HEVC decoder. The HEVC decoder passes the decoded videos to Oculus Software Development Kit (SDK) to generate the viewer's FoV for display.

IV. Testbed

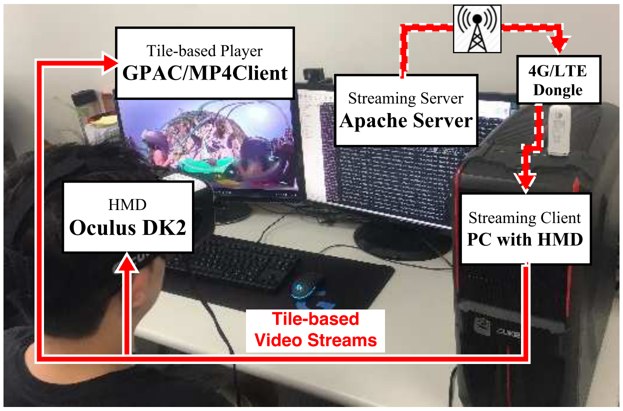


Fig. 4. A photo of our testbed for performance measurement.

We set up a 360° streaming testbed as follows to run the experiments. Fig. 4 shows our testbed with three components, including streaming server, tile-based player, and HMD. At the server side, we use 360° videos at 4K resolution and 30 frame-per-second downloaded from YouTube as our videos. We use an open-source codec, Kvazaar [29], to slice the videos into multiple tiles (subvideos). We package our raw HEVC bitstream and cut the video into small segments with a few seconds using MP4Box [26]. These streaming files are stored on our PC workstation with an Intel i7 CPU and 8 GB RAM.

At the client side, 360° video player is installed on our PC workstation with an Intel i7 CPU and 16 GB RAM. We use Oculus Rift DK2 to render videos and MP4Client as our tile-based DASH player. We adopt HUAWEI E3267 4G dongle to connect to our base station in campus, which is equipped with RBS 6601. The uplink frequency band is 1775–1785 MHz, and downlink one is 1870–1880 MHz.

V. Measurement Design

We conduct a pilot experiment and find the FoV of Oculus Rift DK2 is $100^\circ \times 100^\circ$, which is fairly close to that of other HMDs we have tried. We employ a 1-min 360° video [28], called Kangaroo Island in our experiments. Several parameters are varied in our experiments. In particular, we consider the number of tiles in $\{$1x1, 3x3, 5x5, 7x7, 9x9$\}$, the DASH segment length in $\{1, 4, 10\}$ seconds, and the video bitrate (whole 360° video) in $\{3, 6, 9\}$ Mbps. Due to the space

TABLE I
VIDEO SIZE OF UNDER DIFFERENT NUMBER OF TILES (MB)

Quality Level	Number of Tiles				
	1x1	3x3	5x5	7x7	9x9
Low	23.73	23.93	24.35	24.97	32.10
Medium	47.35	47.57	48.01	48.67	49.53
High	94.70	94.88	95.26	95.93	96.78

limitations, we only report results from 10-sec segments. We answer the following questions using the experiment design given below.

- **How tile size affects the streaming system performance?** We vary the number of tiles and video bitrates. We play each video three times, and report the medium performance results. The bandwidth consumption and network overhead are collected and analyzed using Wireshark [30].
- **How much bandwidth saving can we get by selectively requesting useful tiles?** We modify the client to only request the tiles that will be watched by viewers. The viewer's FoVs are randomly chosen from the 360° video viewing dataset [28]. We compare the network traffic amount and transfer time with and without selectively requesting useful tiles.
- **How many users can be supported in one cell?** We repeat the above experiments but with more clients to observe the capacity of one cell. More specifically, we increase the number of clients (each with a 4G dongle and a SIM card), and observe how competition affects the streaming system performance.

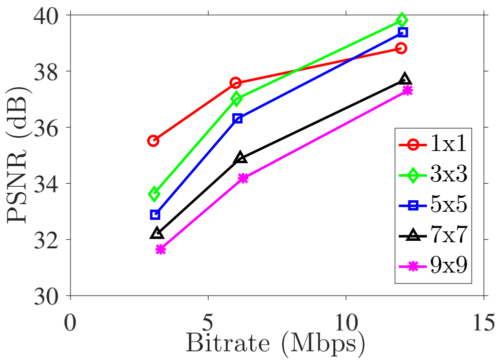


Fig. 5. Video quality under different bitrate and number of tiles.

VI. RESULTS AND FINDINGS

In this section, we present our simple experimental results and analysis.

A. Tile Size

We split each original 360° video into different numbers of tiles, encode each of them at three different quality levels, and stream them over 4G cellular networks. Table I shows the detail information of the considered tiled video, including quality levels, size, and number of tiles. Based on the current network condition, the client adaptively requests and downloads different tiles. Fig. 5 shows video quality under different bitrates and numbers of tiles. Because we restrict the motion prediction when encoding the tiles and disable the DBF. The more tiles we employ, the more quality drop we suffer. To evaluate the influence of the motion-constrained tiles, the client downloads different numbers of tiles. A sample result on requested and received sizes of 7x7 tiles is given in

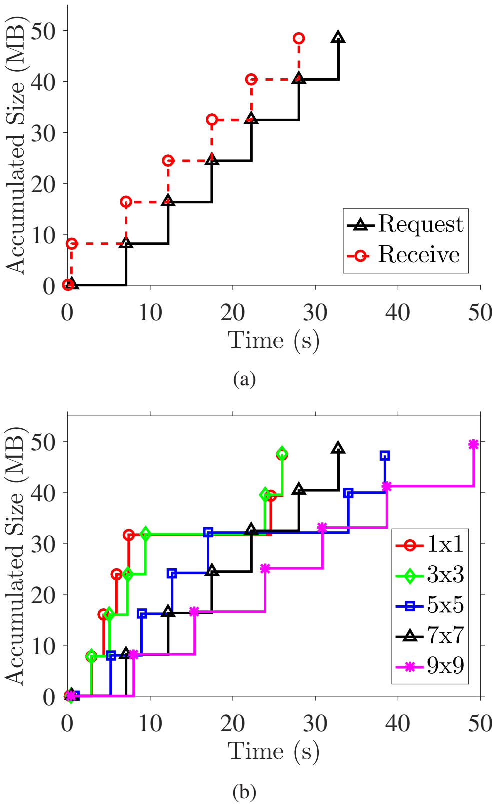


Fig. 6. Accumulated packet size over time: (a) sample requested/received results with 7x7 tiles and (b) received sizes of different tile sizes.

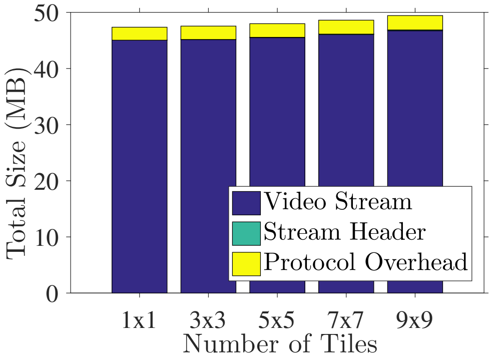


Fig. 7. The composition of transferred packet sizes under different numbers of tiles.

Fig. 6(a), which shows the dynamics. Fig. 6(b) shows that the more tiles we split (from 1x1 to 9x9), the longer a client spends to download all the tiles. This is because the client sequentially downloads all of tiles. Therefore, the protocol overhead, such as streaming protocol headers, metadata, network routing information, etc., slows down the download process when the number of tiles is larger. We report the composition of received data in Fig. 7. The data size is divided into: (i) actual video stream, (ii) HEVC stream header, and (iii) streaming protocol overhead. This figure reveals that majority of streamed data are videos, and the stream header is negligible. On the other hand, the protocol header does not increase significantly when the number of tiles increases.

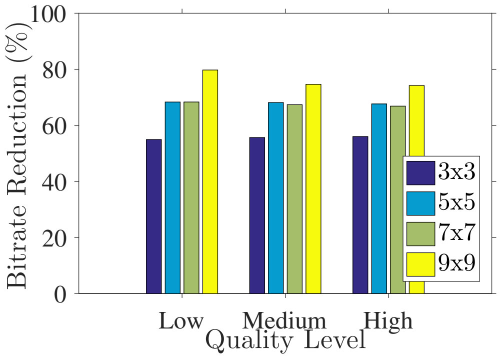


Fig. 8. Bandwidth saving of tile skipping, compared to 1x1 tile.

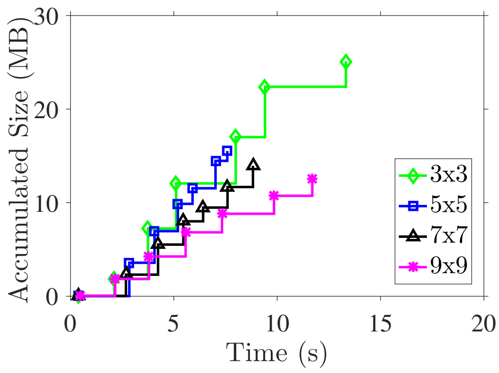


Fig. 9. Accumulated received size of different tile sizes.

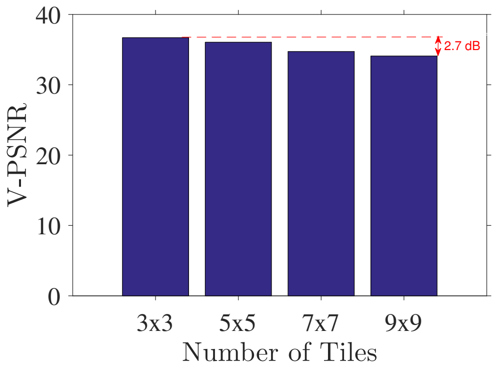
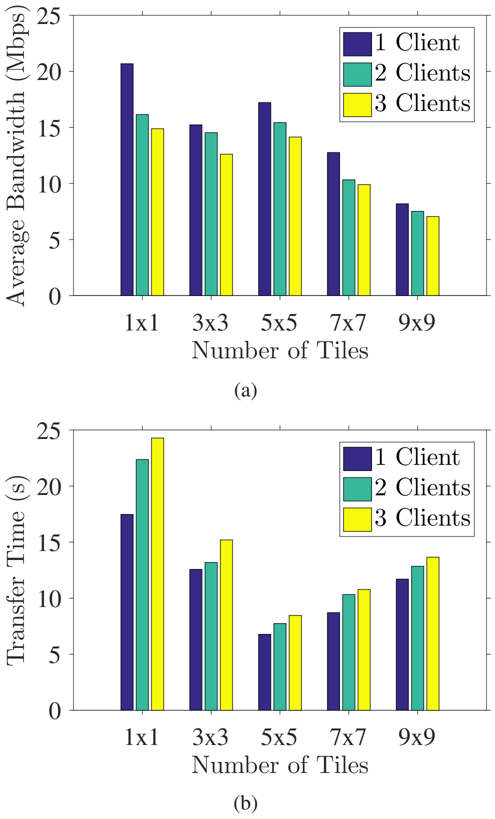


Fig. 10. The V-PSNR quality with different numbers of tiles.



(a)

(b)

Fig. 11. (a) The average bandwidth consumption and (b) total data transfer time when multiple viewers watching tile-based 360° videos at the same time.

B. Tile Skipping

To further reduce the bandwidth consumption, a client may only download and display those tiles that fall into the viewer's FoV. We refer to this strategy as tile skipping, and we next report how it affects the bandwidth consumption, video quality, and transfer time. Fig. 8 shows the normalized bitrate saving, compared to the non-tiled video. The result shows that skipping tiles based on viewer's FoV saves the bandwidth by up to 80%. Generally speaking, if a video is split into smaller tiles, client can download and display the tiles based on viewers' FoV more precisely. This leads to more bandwidth saving. Fig. 9 shows that a client downloads different number of tiles that fall into the viewer's FoV. With more tiles, the download process completes sooner, as smaller regions are downloaded when tiles are smaller. Besides, Fig. 10 plots Viewport PSNR (V-PSNR) [31], which calculates the Peak Signal-to-Noise Ratio (PSNR) for the viewer's FoV (instead of the whole 360° video). This figure confirms the coding efficiency loss due to tiled encoding; however, the video quality drop from 3x3 to 9x9 is less than 2.7 dB, which may be acceptable in some scenarios. In summary, streaming tiled 360° videos over cellular networks reduces the amount of data transfer, shortens the download time, and incurs minor video quality drop.

C. Scalability

We report the number of concurrent 360° video viewers that can be concurrently supported by our base station. Fig. 11(a) shows the average bandwidth consumption of multiple viewers downloading tiled 360° video. In general, more average bandwidth is used by our 360° video streaming system when there is a single client. On the other hand, more tiles results in smaller average bandwidth. This can be attributed to the fact that each tile is requested *sequentially*, and thus more tiles results in longer delay. Fig. 11(b) illustrates the transmission time over the cellular networks. We find that smaller tile size leads to shorter transfer time. This is because our 360° video streaming system only requests the useful tiles. In summary, our cellular base station can support at least 3 clients watching 360° videos.

VII. Conclusion

In this paper, we design measurement experiments to quantify the performance of emerging VR streaming over cellular networks. We build a streaming testbed and conduct extensive experiments using real user traces [28]. In particular, we investigate the pros and cons on tiled 360° video streaming, including the degraded coding efficiency due to motion constraints and the higher flexibility offered by requesting viewer FoV only. Our experiments reveal that: (i) FoV-based video streaming can save up to 80% in bandwidth consumption and (ii) more tiles suffer from lower coding efficiency and late segments due to higher bandwidth consumption. Our measurement results provide better understanding on tiled-based 360° video streaming over cellular networks. We acknowledge that the current work can be extended in several directions. For example, based on a fixation prediction algorithm [32], we are developing a bitrate allocation algorithm to optimize the mobile AR (Augmented Reality)/VR systems with HMDs.

Acknowledgements

This work was partially supported by the Ministry of Science and Technology of Taiwan (# 105-2221-E-007-088), and by the Lite-On Technology Corporation.

References

[1] "Facebook Oculus Rift," 2017, https://www.oculus.com.

[2] "HTC Vive," 2017, https://www.htcvive.com.

[3] "Samsung Gear VR," 2017, http://www.samsung.com/global/galaxy/gear-vr.

[4] "Richo Theta S," 2017, https://theta360.com.

[5] "Luna 360 VR," 2017, http://luna.camera/.

[6] "Samsung Gear 360," 2017, http://www.samsung.com/global/galaxy/gear-360/.

[7] "Facebook," 2017, https://www.facebook.com/.

[8] "YouTube," 2017, https://www.youtube.com/.

[9] "Global Internet Phenomena," 2017, https://www.sandvine.com/trends/global-internet-phenomena/.

[10] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," in *Proc. of International ACM Conference on Multimedia Systems (MMSys'11)*, San Jose, CA, 2011, pp. 133–144.

[11] R. Aparicio-Pardo, K. Pires, A. Blanc, and G. Simon, "Transcoding live adaptive video streams at a massive scale in the cloud," in *Proc. of ACM International Conference on Multimedia Systems (MMSys'15)*, Portland, OR, 2015, pp. 49–60.

[12] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," in *Proc. of Virtual Reality Conference (VR'10)*, Waltham, MA, 2010, pp. 211–218.

[13] K. Ngo, R. Guntur, and W. Ooi, "Adaptive encoding of zoomable video streams based on user access pattern," in *Proc. of ACM Conference on Multimedia Systems (MMSys'11)*, San Jose, CA, 2011, pp. 211–222.

[14] H. Wang, M. Chan, and W. Ooi, "Wireless multicast for zoomable video streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 1, pp. 5:1–5:23, 2015.

[15] J. Feuvre and C. Concolato, "Tiled-based adaptive streaming using MPEG-DASH," in *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*, Klagenfurt, Austria, 2016, pp. 41:1–41:3.

[16] L. D'Acunto, J. Berg, E. Thomas, and O. Niamut, "Using MPEG DASH SRD for zoomable and navigable video," in *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*, Klagenfurt, Austria, 2016, pp. 34:1–34:4.

[17] V. 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.

[18] O. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Lim, "MPEG DASH SRD: Spatial relationship description," in *Proc. of ACM International Conference on Multimedia Systems (MMSys'16)*, Klagenfurt, Austria, 2016, pp. 5:1–5:8.

[19] C. Zhou, Z. Li, and Y. Liu, "A measurement study of Oculus 360 degree video streaming," in *Proc. of ACM Conference on Multimedia Systems (MMSys'17)*, Taipei, Taiwan, 2017, pp. 27–37.

[20] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over http," in *Proc. of ACM Conference on Multimedia Systems (MMSys'17)*, Taipei, Taiwan, 2017, pp. 261–271.

[21] T. El-Ganainy, "Spatiotemporal rate adaptive tiled scheme for 360 sports events," *arXiv preprint arXiv:1705.04911*, 2017.

[22] A. Ghosh, V. Aggarwal, and F. Qian, "A rate adaptation algorithm for tile-based 360-degree video streaming," *arXiv preprint arXiv:1704.08215*, 2017.

[23] M. Chen, W. Saad, and C. Yin, "Virtual Reality over wireless networks: Quality-of-Service model and learning-based resource management," *arXiv preprint arXiv:1703.04209*, 2017.

[24] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. of the Workshop on All Things Cellular: Operations, Applications and Challenges (ATC'16)*, New York, NY, 2016, pp. 1–6.

[25] C. Feldmann, C. Bulla, and B. Cellarius, "Efficient Stream-Reassembling for Video Conferencing Applications using Tiles in HEVC," in *Proc. of International Conferences on Advances in Multimedia (MMEDIA'13)*, Venice, Italy, 2013, pp. 130–135.

[26] "MP4Box," 2017, https://gpac.wp.imt.fr/mp4box/.

[27] "MP4Client," 2017, https://gpac.wp.imt.fr/player/.

[28] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu, "360° video viewing dataset in head-mounted Virtual Reality," in *Proc. of ACM International Conference on Multimedia Systems (MMSys'17)*, Taipei, Taiwan, 2017, pp. 211–216.

[29] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. Hämäläinen, "Kvazaar: Open-source HEVC/H.265 encoder," in *Proc. of ACM on Multimedia Conference (MM'16)*, Amsterdam, The Netherlands, 2016, pp. 1179–1182.

[30] "Wireshark," 2017, https://www.wireshark.org/.

[31] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR'15)*, Fukuoka, Japan, 2015, pp. 31–36.

[32] C. Fan, J. Lee, W. Lo, C. Huang, K. Chen, and C. Hsu, "Fixation prediction for 360° video streaming in head-mounted virtual reality," in *Proc. of ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'17)*, Taipei, Taiwan, 2017, pp. 67–72.