How to Evaluate Mobile 360° Video Streaming Systems?

Shivang Aggarwal[1], Sibendu Paul[2], Pranab Dash[2], Nuka Saranya Illa[1],
Y. Charlie Hu[2], Dimitrios Koutsonikolas[1], Zhisheng Yan[3*]
[1]University at Buffalo, The State University of New York, [2]Purdue University, [3]Georgia State University

ABSTRACT

360° video brings an immersive experience to users by projecting panoramic content onto the display. Viewing 360° videos on untethered mobile platforms further enhances the immersive user experience. As a result, a large number of systems for optimized high-resolution 360° video streaming to mobile devices have been proposed over the past few years.

In this paper, we review the diverse set of research methodologies in the system design and evaluation of recently proposed mobile 360° video streaming systems and discuss a number of pitfalls that prevent a fair and meaningful comparison among different systems. Our discussion suggests that there is an urgent need to redefine the design objectives and to develop an effective methodology for a meaningful evaluation and comparison of different systems. We finish with a set of concrete guidelines on the design and evaluation methodology of future mobile 360° video systems.

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Human-centered computing → Mobile computing**; **Ubiquitous and mobile computing design and evaluation methods**.

ACM Reference Format:
Shivang Aggarwal, Sibendu Paul, Pranab Dash, Nuka Saranya Illa, Y. Charlie Hu, Dimitrios Koutsonikolas, Zhisheng Yan. 2020. How to Evaluate Mobile 360° Video Streaming Systems? In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications (HotMobile '20), March 3–4, 2020, Austin, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3376897.3377865

1 CHALLENGES IN EVALUATING MOBILE 360° VIDEO STREAMING SYSTEMS

360° videos enable users to interact with the immersive virtual world by moving their heads using a head-mounted display (HMD) and visualize any particular spatial regions they desire, referred to as viewport, in the entire 360° scene. To support true immersive experience, 360° videos should feature high frame rate, high resolution, low motion-to-frame latency, and be viewed from a mobile device. This set of requirements has motivated the design of a number of 360° video streaming systems that aim to deliver high-quality videos wirelessly to mobile devices.

In this paper, we argue that the set of unique video parameters and content features of 360° videos also pose a set of distinct new challenges to the evaluation of 360° video streaming systems.

First, the multiple viewports of 360° videos and the requirement of close-eye display showing every single detail entail an extremely high resolution, frame rate, and bitrate for 360° videos. Only by meeting these requirements the sense of immersion can be created in the HMD. However, given the emerging nature of the research topic, such high-resolution, high-bitrate videos may not be widely available to the research community. Furthermore, when streaming systems exploit optimization techniques such as viewport adaptation to deliver only part of the 360°videos, the content to be streamed and displayed on an HMD is not only dependent on the offline camera capture as in regular videos but also determined by how the user interacts with the 360° video and selects the viewport. This further complicates consistent evaluation across multiple studies. In summary, *the wide range of 360° video features makes it challenging to select consistent source videos for streaming system evaluation.*

Second, the higher quality of experience (QoE) of 360° videos results in a wireless network bandwidth requirement that is larger than ever before. Studies have shown that a 360° video with an equivalent HD TV viewing experience would require a 4K by 2K resolution for a viewport at 60 fps. The entire video would become 12K resolution with a 400 Mbps bitrate [15]. The diverse bitrates of different 360 videos also lead to a large range of bandwidth requirements. As a result, *it is challenging to consistently and fairly evaluate the bandwidth efficiency of 360° video streaming systems in real-world settings.*

Third, a user's QoE in a 360° video streaming system has a clear departure from traditional video systems. A 360° HMD video received with the same bitrate, frame rate, delay, or rebuffer rate can be perceived significantly different from a regular desktop video. Therefore, traditional metrics for evaluating regular video streaming systems are often no longer appropriate and *it becomes imperative to design proper new evaluation metrics for 360 video quality and QoE.*

In addition to the challenges in the evaluation methodology, we argue that the same set of unique video parameters and content features of 360° videos along with the diverse set of QoE requirements also make it challenging to define a consistent set of design objectives for mobile 360° video systems. Because the design objectives and evaluation methodologies are inter-related, the lack of common design objectives adds to the list of challenges for a consistent and fair evaluation of alternative system designs.

*Shivang Aggarwal and Sibendu Paul are co-primary authors.

HotMobile '20, March 3–4, 2020, Austin, TX, USA

ACM ISBN 978-1-4503-7116-2/20/03…$15.00
https://doi.org/10.1145/3376897.3377865

Table 1: Comparison of evaluation methodologies in recent works.

Protocol	Videos	Bitrates	Bandwidth	Evaluation Metrics
Flare [17]	YouTube 4K/8K	12.9-21.5 Mbps (4K)	90 Mbps (WiFi), 9.6±4.5 Mbps (LTE traces)	CRF level, stalls/min, User Perceived Ratio, BW savings
Rubiks [12]	YouTube 4K/8K	N/A	4K: 0.15-2.1 MBps, 8K: 3-27 MBps (LTE traces)	own QoE metric, SSIM, rebuffering time, BW savings
Freedom [18]	dataset 4K, YouTube 8K	N/A	3.7-11.5 Mbps (WiFi, LTE)	Latency, BW savings
Cutting the Cord [13]	live 2K/4K	N/A	1.6 Gbps (WiGig)	Latency, Quality
Pano [11]	dataset 3K	N/A	0.71/1.05 Mbps (LTE traces)	PSPNR, MOS, BW savings, buffering ratio
POI360 [22]	live 4K	3 Mbps	0-5 Mbps (comm. LTE)	PSNR, MOS, latency, freeze ratio
BAS-360 [20]	N/A	0.64-6.4 Mbps	1.45-6.15 Mbps (manually set)	supported bitrate, stalling time
360ProbDASH [21]	3K	1.4-21.6 Mbps	1-3 Mbps (manually set)	V-PSNR, supported bitrate, stall ratio, viewport deviation
[16]	YouTube 8K	2.5-9,5 Mbps	21.8±12.3 Mbps (LTE traces), 5/35 Mbps fixed	QP level, BW efficiency, perceived BW, freeze time

2 PITFALLS

In this section, we discuss the pitfalls in the evaluation and design of recently proposed mobile 360° video streaming systems.

2.1 Disparate Evaluation Methodologies

Table 1 compares the evaluation methodologies used by a number of recent works published by the systems/networking [11, 12, 17, 18, 20, 22] and multimedia community [16, 21] across four aspects: type of videos used in the evaluation, bitrates of encoded videos, bandwidth, and evaluation metrics. It is evident that there are no clear guidelines to drive the evaluation of mobile 360^o video streaming systems; often, each system is evaluated using an arbitrary set of evaluation parameters and different evaluation metrics.

2.1.1 Unsuitable or Arbitrary Evaluation Metrics. A very large number of metrics has been used in the evaluation of 360° video streaming systems, classified in three categories: video quality metrics, QoE metrics, bandwidth related metrics. We discuss bandwidth savings in §2.1.3 and focus here on video quality and QoE metrics.
Video quality metrics. Traditional metrics, such as PSNR, PSPNR, SSIM, and QP/CRF level, measure the distortion between the received planar frame and the source planar frame. The problem is that 360° videos are not viewed in the format of planar frame. They need to be projected to the sphere so that users only view a viewport of the 360° video. A high distortion of the entire planar frame does not reflect the user experience, as the user's viewport may be in high quality with negligible distortion. To address the spherical format of 360° videos, some new metrics, which are variants of regular video metrics, have been proposed. Metrics such as V-PSNR measure the similarity between the viewport and the source content. However, pixel-level distortion has been long shown to be only remotely related to user experience.
QoE metrics. Metrics such as delay, stalls/min, freeze ratio, and rebuffering time, which capture the system performance, are often used as QoE metrics. However, replacing user experience by these metrics can be misleading as has long been shown in traditional video streaming systems [19]. MOS is a subjective metric that has been used for decades in evaluating regular video streaming systems. Although MOS indeed manifests user experience to some extent, the evaluation methodology must be carefully designed to infer the user experience. The reason is that 360° video streaming is highly related to the user interaction. Measuring MOS without specifying the user interaction or the actual user viewport can lead to unfair conclusions. For example, MOS can be high when a user fixes her viewport on the static content but the same system may demonstrate a low MOS when the user frequently moves her head. Some works have tried to define their own QoE metric (e.g., [12]), but in doing so, there is always a risk that they end up designing a metric tailored to their own design objectives.

Another issue that becomes evident from Table 1 is that there is no guideline on which metric(s) to use from each category; often times, each work uses a different subset of metrics. Among the video quality metrics, QP level, CRF level, PSNR, PSPNR, SSIM are each used by a single work in Table 1. Surprisingly, some works do not use any video quality metric in their evaluation, e.g., [18, 20]. Among the QoE related metrics, stalling/rebuffering/freeze time is the most popular one (7 instances), followed by latency (only 3 instances) and MOS (only 2 instances).

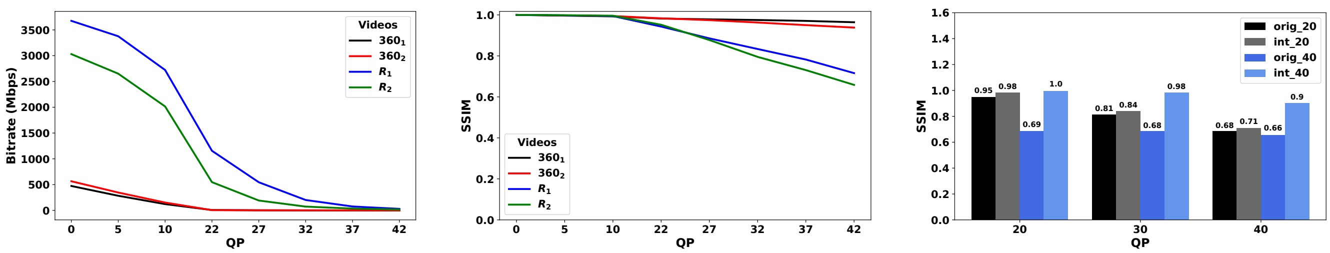
2.1.2 Arbitrary Videos for Testing. As evidenced from Table 1, recent works use very different videos for the evaluation of the proposed systems with respect to three aspects: (i) *Different sources.* The majority of works have been evaluated using videos downloaded from YouTube (e.g., [10, 12, 16–18]). Few works (e.g., [11, 18]) have used videos from publicly available repositories [6, 7] or live videos [22]. (ii) *Different resolutions.* Most works use 4K and 8K videos but 2K and 3K have also been used, even by very recent works, e.g., [11]. (iii) *Different content.* The lack of a common video repository results in videos of different content used in different works. Additionally, some works provide no details about the videos used in their evaluation, e.g., [20].

There is a close interplay among these three aspects and the other evaluation aspects shown in Table 1. Clearly, using different resolutions does not allow for a fair comparison between two protocols. However, even in the case of the same resolution, different content can lead to drastically different encoded bitrates or different QoE for the same encoding parameters (e.g., same QP/CRF level). This in turn can result in misleading conclusions about the benefits of a new protocol in terms of QoE or bandwidth savings.

As an example, we consider four videos encoded at different QP levels (QP 0 corresponds to raw/uncompressed video) and compare their bitrates and SSIM in Figures 2a and 2b, respectively. Two of these videos (R_1 and R_2, in Figures 1a, 1b) are regular 4K videos downloaded from Derf's collection under Xiph [2], a publicly available dataset that has been used in the evaluation of regular video streaming systems. The other two (360_1 and 360_2, in Figures 1c, 1d) are 360° 4K videos downloaded from SJTU [14] – this is the only publicly available dataset of 360°videos we are able to find. Note that there are no specific reasons for choosing these videos other than their availability; other videos could also have been used. We make two observations:
(1) Different content leads to different bitrates. It is expected that 360° videos have higher bitrates than regular videos of the same resolution (4x-6x according to [17]). Nonetheless, Figure 2a shows the opposite. The bitrate of the two 360° videos is about 500-600

(a) R_1 frame. (b) R_2 frame. (c) 360_1 frame. (d) 360_1 frame.

Figure 1: Four videos used in our study.



(a) Bitrate comparison of four 4K videos encoded at different QP levels. (b) SSIM comparison of four 4K videos encoded at different QP levels. (c) Decoding and re-encoding YouTube videos.

Figure 2: Pitfalls related to the use of arbitrary videos for evaluation.

Mbps in their raw format (and hence they could be streamed uncompressed over a high quality 802.11ac WLAN) and less than 10 Mbps when encoded with QP 22-42 (typical values used in recent works). In contrast, the bitrates of the two regular videos are higher than 3 Gbps in their raw formats (and hence, they cannot be streamed even over 60 GHz WLANs) and they drop to a few 10s of Mbps only for QP levels higher than 40.

(2) Different content leads to different QoE. Figure 2b shows the SSIM of the four videos for different QP levels. Clearly, the same QP level can lead to very different SSIM values for different videos, depending on the content. Videos 360_1 and 360_2 maintain excellent SSIM values for the whole range of QP levels, since their content allows for high compression without loss of information. In contrast, R_1 and R_2 maintain high SSIM (above 0.9) only for QP levels below 25, which yield extremely high bitrates, as we saw in Figure 2a. In summary, *different videos can lead to different conclusions about the same video quality metric*. For example, SSIM is almost meaningless for videos 360_1 and 360_2, as it remains excellent for all QP levels.

YouTube videos. We now take a closer look at a common practice in the evaluation of recent systems [10, 12, 16–18] – the use of YouTube videos, which are downloaded, decoded, and re-encoded at different QP/CRF levels. The problem with this methodology is that encoding is lossy, and hence it is not possible after decoding to obtain the raw video that was uploaded on YouTube. Hence, QoE metrics such as SSIM or PSNR are calculated with respect to the decoded (lossy) video and not with respect to the original video, and they may appear higher than in reality.

To evaluate the impact of the decoding and re-encoding process, we encoded video R_1 at two different QP levels (20 and 40), decoded each version, and re-encoded at three different QP levels (20, 30, 40). Figure 2c shows the SSIM of the video obtained after the encoding, decoding, and re-encoding process when using as a reference the original raw video and the intermediate (decoded) video ($orig_{20}$, $orig_{40}$ and int_{20}, int_{40}, respectively). If a video is initially encoded at a low QP level (QP 20), one can restore it with low loss. As a result, the SSIM of the re-encoded video at any QP level is similar when using as a reference either the original video or the decoded video (compare $orig_{20}$ vs. int_{20} in Figure 2c). However, if the video is initially encoded at a high QP level, the quality of the restored video is much lower compared to the initial video. As a result, the SSIM of the re-encoded video appears to be very high when the intermediate video is used as a reference but it is in fact very low with respect to the original video (compare $orig_{40}$ vs. int_{40} in Figure 2c).

Assume now that Bob downloads video R_1 from YouTube, encoded at QP level 40 yielding a 200 Mbps bitrate (Figure 2a), to use it in the evaluation of his new system. Bob decodes this video, re-encodes it at QP 20, and streams it using his system. Since Bob has no way to obtain the original YouTube video, he uses the decoded video as a reference to calculate SSIM and obtains a value of 1 (Figure 2c). However, the actual SSIM with respect to the original video YouTube uploaded is only 0.69. Note that YouTube actually encodes 4K/8K videos at much lower bitrates (up to 48 Mbps for 8K). Thus, high motion videos, such as R_1, are most likely already heavily compressed when uploaded on YouTube. *Using such videos in the evaluation of new systems after decoding and re-encoding can lead to false claims about high supported video quality.*

2.1.3 Arbitrary Bandwidth. Different systems have been evaluated over very different bandwidth levels. With the exception of [13, 18] that stream videos over real WiFi/LTE networks, the selected bandwidth levels in most works are either based on publicly available network traces [11, 12, 16, 17]) or chosen arbitrarily [20, 21]. While the use of network traces provides much more realistic settings than manually fixing the bandwidth, it is important to ensure that the traces are based on recent measurements and reflect current network technologies. Unfortunately, this is not often the case. Most recent works [11, 16, 17] published in 2017-2019 use a 4G/LTE trace collected back in 12/2015-02/2016. Even worse, the authors in [12] use an HSDPA dataset [1] from 2011-2013.

The choice of bandwidth is often tied to the choice of videos used for testing and the target quality. For example, if the target quality is an SSIM of 0.95, Figure 2b shows that videos 360_1, 360_2 can satisfy this requirement with a large range of QPs, which in turn offers a lot of flexibility in the choice of bandwidth (Figure 2a). On the other hand, if R_1 and R_2 are used, the same requirements can only be satisfied with QP levels of at most 22, which in turn requires a bandwidth of at least 600 Mbps. As a result, we often see works using very different bandwidth levels in their evaluation even though they consider videos with the same resolution and similar QP/CRF levels. As an example, the authors of Flare [17] and Rubiks [12], two recently proposed state-of-the-art systems, use 4K 360° videos with similar QP levels in their evaluation but bandwidths of 9.6 Mbps and 1.2-16.8 Mbps, respectively. While one would expect more recent works to use higher bandwidth levels, this is, surprisingly, not always the case. The authors of Pano [11], published only 5 months ago, use only 3K videos over 0.71 Mbps and 1.05 Mbps, although the network trace they use provides logs of up to 95 Mbps!

Arbitrary selection of bandwidths can lead to misleading conclusions. Assume Bob and Alice propose two new systems, P_1 and P_2, respectively. P_1 incorporates a novel, very aggressive viewport prediction algorithm that results in higher bandwidth savings than P_2 (50% vs. 20%) but at the cost of higher prediction error. Bob can choose a video like R_1 at QP level 32 (which, based on Figure 2a, requires a bitrate of about 300 Mbps) over a bandwidth of 200 Mbps for the comparison of the two systems, and show that the video can be played with P_1 thanks to the 50% bandwidth savings achieved by its viewport prediction algorithm. In contrast, P_2 requires a bandwidth of at least 240 Mbps (only 20% savings) and hence, the same video will result in many stalls if streamed over 200 Mbps with P_2. On the other hand, Alice can choose to evaluate the two systems using videos 360_1 and 360_2 at QP 22 over a bandwidth of 100 Mbps, which can be supported by both systems, and demonstrate better QoE with P_2 compared P_1 thanks to a smaller number of prediction errors.

2.2 Disparate Design Objectives

The design objectives of mobile 360° video streaming systems should be dictated by the QoE requirements of 360° videos.

User QoE. To support acceptable user experience, which is dictated by the human biology, 360° streaming systems have to meet three critical QoE requirements:

(1) **Latency** is a direct measure of the responsiveness of 360° video streaming applications. As in regular video streaming, *stall time* is the duration in which the video player freezes due to missing frames or empty buffer. Different from regular video streaming, 360° video streaming imposes an additional user-perceived latency, often described as *motion-to-photon*, i.e., the elapsed time between an input (e.g., changing the head position) and the moment when a corresponding frame is displayed on the screen. Although the motion-to-photon latency is often associated with VR gaming, where user

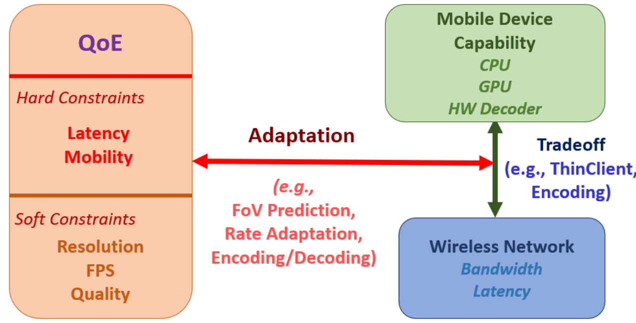


Figure 3: Design tradeoffs in supporting 360° videos.

inputs cannot be either predicted or prefetched, it is highly relevant and should be evaluated for any 360° video displayed via an HMD.[1] Most recent mobile 360° video streaming systems rely on viewport prediction and no viewport prediction algorithm can achieve 100% accuracy. For example, if there is a sudden movement, viewport prediction can be wrong and the prefetched content will not match the movement. In such cases, the system needs to fetch new content, which causes rebuffering and large motion-to-photon delay.

(2) **Frame rate** needs to be sufficiently high (30-60 FPS) in order to have smooth immersive experience. With a fixed **resolution** (4K/8K) of input video streams, the video **quality** perceived by consumer must lie inside an acceptable range (e.g., SSIM > 0.9) [8].

(3) **Mobility** of users should not be impeded by tethered VR headsets (i.e., connected to a server via a cable).

Design tradeoffs. Amongst the three QoE metrics, motion-to-photon latency (i.e., less than 20 ms [3, 4]) and supporting mobility of users (i.e., HMD or mobile devices must be untethered) are *hard constraints*. On the other hand, FPS, video quality and resolution should be above a certain predefined threshold, and hence act as *soft constraints*. These constraints in turn dictate different design goals and design tradeoffs, as shown in Figure 3. First, the mobile device and wireless network capability together limit the software constraints that can be achieved – the FPS or quality or resolution beyond a minimum requirement for improved immersion requires either mobile devices with higher computational capability or a faster wireless network. Second, for a given resolution/FPS/quality, there exists a design tradeoff between resource usage of mobile devices (e.g., CPU, GPU usage, and battery lifetime) and the available wireless network bandwidth. Under a fixed resource constraint on mobile devices, in order to satisfy the stringent QoE requirements, computation workload must be offloaded to a remote server (e.g., remote rendering) which demands higher wireless bandwidth. On the other hand, the requirement of smooth immersive experience under constrained network bandwidth proliferates mobile computation workload (e.g., heavy decoding) and shortens the battery life. Third, the system needs to adapt the computation/network load and hence the soft constraints in reaction to network dynamics (e.g., rate adaptation.)

Disparate design tradeoffs. In this context, we argue that state-of-the-art systems are missing critical QoE requirements from their design objectives, shown in Table 2. This in turn contributes to their

[1] YouTube video creators recommend the use of HMDs to view 360° videos [5].

Table 2: QoE objectives of existing systems.

Protocol	Hard Constraints		Soft Constraints		
	Latency (<20 ms)	Mobility (Untethered)	Resolution (4K/8K)	FPS (30-60)	Quality (SSIM > 0.9)
Freedom [18]	X	✓	✓	✓	X
Rubiks [12]	X	✓	✓	X	X
Flare [17]	X	✓	✓	X	X
Cutting the Cord [13]	✓	X	✓	✓	✓
Pano [11]	X	✓	X	X	✓

inconsistent evaluation methodologies. Along with the minimum stall time guarantees, 360° video streaming must also satisfy the stringent requirement of motion-to-photon latency (i.e., < 20 ms). Most recent works on 360° video streaming on commodity mobile devices such as Rubiks [12], Flare [17], and Pano [11] do not include this stringent *motion-to-photon* latency requirement in their problem formulation. On the other hand, *Freedom* [18] endeavors to achieve this latency constraint for the current LTE network but fails by a large margin of 80 ms. Meanwhile, *Cutting the Cord* [13] achieves the motion-to-photon latency of 20 ms for high-resolution 360° video streaming and VR with the help of high bandwidth provided by 60 GHz WiGig (i.e., pushing towards maximum network resources possible). Although it eliminates the pre-existing tethered link between the rendering server and client, the display HMD is still connected with a client laptop. Thus, *Cutting the cord* fails to provide entirely seamless user experience and does not consider the capabilities of commodity mobile devices.

3 RECOMMENDATIONS

3.1 Evaluation Metrics

Developing a comprehensive and consistent set of evaluation metrics for 360° video quality and user experience has been an open research challenge in the multimedia community [9, 10, 23, 24]. An immediate action we recommend is to adopt HMD-tailored quality evaluation metrics – Sphere-PSNR (S-PSNR) and Viewport-PSNR (V-PSNR) [23]. S-PSNR projects the planar video format for streaming to the sphere format for viewing and then compares the MSE between the viewed sphere and the original sphere on the server. Instead of focusing on the whole sphere, V-PSNR only compares the difference between the user-perceived viewport and the corresponding region in the original sphere. Both metrics can be used for objective streaming algorithm designs.

However, similar to PSNR, S-PSNR and V-PSNR have been shown to be unable to quantify the QoE accurately [24]. Hence, we believe that it is imperative to design a QoE model that quantifies the user experience during viewport viewing for 360° videos. The viewport viewing pattern and the subjective user perception should be jointly considered. A possible approach is to conduct a MOS-based user study and correlate the user ratings with the viewport viewing pattern.

3.2 Common Dataset

The root cause of the pitfalls we discussed in §2.1.2, §2.1.3 is the lack of a common 360° video dataset agreed by the community. Although standard datasets for regular 2D video evaluation have been well established, no such dataset exits for 360° videos. Therefore, we recommend building a 360° video dataset for correct and consistent evaluation. First, the videos in the dataset must be raw 360° videos captured by a high-end 360° camera. This guarantees a high resolution and clarity of the source content. The transcoded videos from such a dataset would achieve a reasonable size for practical 4K or 8K video evaluation. Second, this dataset should have diverse video content types covering different content dynamics which may significantly affect the performance of streaming systems. In addition to different levels of object motion, brightness, scene complexity which have been shown to have impacts on regular video streaming, the motion of the 360° camera becomes a new property of a 360° video dataset because a lot of 360° videos are captured by a moving person/car holding a camera.

3.3 Design Objectives

Motivated by our discussions in §2.2, we outline two types of essential design objectives for mobile 360° video streaming systems: *feasibility* and *resource optimization*.

(1) Feasibility design objectives: For given types of wireless network and mobile device, is it feasible to support 360° videos of certain Resolution/FPS/Quality, i.e., satisfying the two hard constraints? Both Rubiks and Flare study the feasibility of supporting 4K/8K 360° videos on commodity mobile devices via WiFi and LTE, but they do not treat QoE, in particular latency, as a hard constraint.

(2) Resource Optimization Objectives: (i) For a given *Resolution/FPS/Quality*, how to optimize the bandwidth requirement for a given mobile device, or mobile device resources (e.g., power or battery life) for a given wireless network type? (ii) For a given Resolution/FPS/Quality, how to best tradeoff mobile resource usage and wireless network bandwidth?

4 SUMMARY

We discussed a number of pitfalls in the evaluation methodologies of recently proposed mobile 360° video streaming systems and proposed a set of initial recommendations to overcome such pitfalls in future research. Our goal is to stimulate further discussions in the community towards converging to a consistent evaluation methodology in this emerging research area.

ACKNOWLEDGMENTS

We thank our shepherd Wei Gao and the anonymous reviewers for their valuable comments. This work was supported in part by NSF and Intel under contract number CNS-1719369.

REFERENCES

[1] [n.d.]. DATASET: HSDPA-bandwidth logs for mobile HTTP streaming scenarios. hsdpa_dataset.
[2] [n.d.]. Xiph.org Video Test Media [derf's collection]. derf_link.
[3] 2014. What VR could, should and almost certainly will be in 2 years. Adarshblog_VR_future.
[4] 2016. Qualcomm Report: Keeping the virtual world stable in VR. keeping-virtual-world-stable-vr-2016.
[5] [Online]. Lesson: Introduction to 360-degree video and virtual reality. https://creatoracademy.youtube.com/page/lesson/spherical-video
[6] Z. Wang S. Yang C. Wu, Z. Tan. 2017. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In *Proc. of ACM MMSys*.
[7] X. Corbillon, F. De Simone, and G. Simon. 2017. 360-Degree Video Head Movement Dataset. In *Proc. of ACM MMSys*.
[8] E. Cuervo, A. Wolman, L.P. Cox, K. Lebeck, A. Razeen, S. Saroiu, and M. Musuvathi. 2015. Kahawai: High-quality mobile gaming using gpu offload. In *Proc. of ACM Mobisys*. 121–135.

[9] V.R. Gaddam, M.Riegler, R. Eg, C. Griwodz, and P. Halvorsen. 2017. Tiling in Interactive Panoramic Video: Approaches and Evaluation. *IEEE Transactions on Multimedia* 18 (2017), 1819 – 1831. Issue 19.

[10] M. Graf, C. Timmerer, and C. Mueller. 2017. Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP: Design, Implementation, and Evaluation. In *Proc. of ACM MMSys*.

[11] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang. 2019. Pano: Optimizing 360° Video Streaming with a Better Understanding of Quality Perception. In *Proc. of ACM SIGCOMM*.

[12] J. He, M.A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han. 2018. Rubiks: Practical 360-Degree Streaming for Smartphones. In *Proc. of ACM MobiSys*.

[13] L. Liu, R. Zhong, W. Zhang, Y. Liu, J. Zhang, L. Zhang, and M. Gruteser. 2018. Cutting the cord: Designing a high-quality untethered VR system with low latency remote rendering. In *Proc. of ACM Mobisys*. 68–80.

[14] X. Liu, Y. Huang, L. Song, R. Xie, and X. Yang. 2017. The SJTU UHD 360-Degree Immersive Video Sequence Dataset. In *Proc. of IEEE ICVRV*.

[15] S. Mangiante, G. Klas, Z. GuanHua A. Navon, J. Ran, and M.D. Silva. 2017. VR is on the edge: How to deliver 360 videos in mobile networks. In *Proc. of Workshop on VR/AR Network*. ACM, 30–35.

[16] S. Petrangeli, V. Swaminathan, M. Hosseini, and F.D. Turck. 2017. An HTTP/2-Based Adaptive Streaming Framework for 360° Virtual Reality Videos. In *Proc. of ACM Multimedia*.

[17] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan. 2018. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proc. of ACM MobiCom*.

[18] S. Shi, V. Gupta, and R. Jana. 2019. Freedom: Fast Recovery Enhanced VR Delivery Over Mobile Networks. In *Proc. of ACM MobiSys*.

[19] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang. 2009. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In *Proc. of ACM Multimedia*.

[20] M. Xiao, C. Zhou, V. Swaminathan, Y. Liu, and S. Chen. 2018. BAS-360° Exploring Spatial and Temporal Adaptability in 360-degree Videos over HTTP/2. In *Proc. of IEEE INFOCOM*.

[21] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo. 2017. 360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming. In *Proc. of ACM Multimedia*.

[22] X. Xie and X. Zhang. 2017. Panoramic Mobile Video Telephony over LTE Cellular Networks. In *Proc. of ACM CoNEXT*.

[23] M. Yu, H. Lakshman, and B. Girod. 2015. A framework to evaluate omnidirectional video coding schemes. In *Proc. of IEEE ISMAR*.

[24] B. Zhang, J. Zhao, S. Yang, Y. Zhang, J. Wang, and Z. Fei. 2017. Subjective and objective quality assessment of panoramic videos in virtual reality environments. In *ICMEW*. IEEE, 163–168.