A Survey on 360° Video Streaming: Acquisition, Transmission, and Display

CHING-LING FAN, WEN-CHIH LO, YU-TUNG PAI, and CHENG-HSIN HSU,
National Tsing Hua University

Head-mounted displays and 360° videos have become increasingly more popular, delivering a more immersive viewing experience to end users. Streaming 360° videos over the best-effort Internet, however, faces tremendous challenges, because of the high resolution and the short response time requirements. This survey presents the current literature related to 360° video streaming. We start with 360° video streaming systems built for real experiments to investigate the practicality and efficiency of 360° video streaming. We then present the video and viewer datasets, which may be used to drive large-scale simulations and experiments. Different optimization tools in various stages of the 360° video streaming pipeline are discussed in detail. We also present various applications enabled by 360° video streaming. In the appendices, we review the off-the-shelf hardware available at the time of writing and the open research problems.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Information systems** → **Multimedia streaming**; • **Human-centered computing** → **User studies**; **Virtual reality**; • **Computing methodologies** → **Image and video acquisition**; **Virtual reality**;

Additional Key Words and Phrases: Virtual reality, 360° videos, video streaming

ACM Reference format:
Ching-Ling Fan, Wen-Chih Lo, Yu-Tung Pai, and Cheng-Hsin Hsu. 2019. A Survey on 360° Video Streaming: Acquisition, Transmission, and Display. *ACM Comput. Surv.* 52, 4, Article 71 (August 2019), 36 pages.
https://doi.org/10.1145/3329119

1 INTRODUCTION

Increasing access bandwidth gradually enables Internet users to share images, audios, and video [129] through various applications. Furthermore, the technology advances in mobile networks also stimulate users to use these multimedia applications anywhere and anytime. In fact, Cisco forecasts that 82% of all network traffic will be video related by 2022 [77], showing that Internet users have been investing more bandwidth in transmitting videos over the years. Additional bandwidth is used toward High-Definition (HD) and Ultra-High-Definition (UHD) videos, two- and multichannel stereoscopic videos, and, more recently, 360° videos for better viewing experience. Among them, 360° videos omnidirectionally acquire scenes and thus allow viewers to dynamically

This work was partially supported by the Ministry of Science and Technology of Taiwan (#107-2221-E-007-091-MY3 and #107-2918-I-007-014) and partially supported by a NOVATEK Fellowship.
Authors' addresses: C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu, Department of Computer Science, National Tsing Hua University, No. 101, Sec. 2, Kuang-Fu Rd., Hsin-Chu City, 30013 Taiwan; emails: {ch.ling.fan, wchih.lo, cealia312}@gmail.com, chsu@cs.nthu.edu.tw.

0360-0300/2019/08-ART71 $15.00
https://doi.org/10.1145/3329119

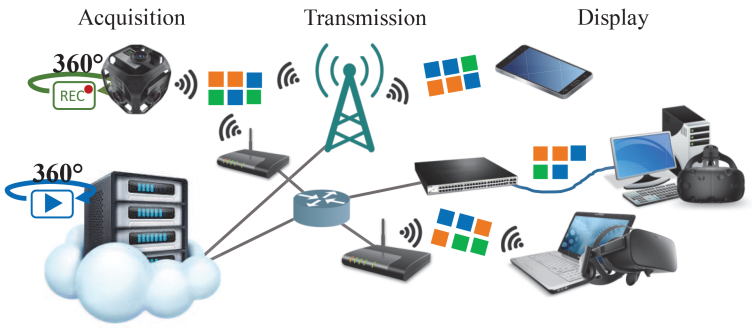

Fig. 1. Overview of 360° video streaming systems.

change their viewports at playout time for immersive viewing experience. Therefore, increasingly more content providers, including YouTube and Facebook, offer on-demand 360° video content and actively develop 360° video–related technologies. For example, Virtual Reality (VR) and Augmented Reality (AR) benefit from the rising popularity of 360° videos and are expected to become popular at a staggering speed [113] in the near future.

We illustrate a high-level overview of 360° video streaming systems in Figure 1. The 360° videos are either acquired by live 360° cameras or from saved video files on cloud servers. These videos are transmitted through the Internet via various access networks, including Ethernet, WiFi, or cellular networks. Various devices support displaying 360° videos, such as desktops, mobile devices, and Head-Mounted Displays (HMDs) (also known as VR headsets [96]). Watching 360° videos with HMDs offers a more immersive experience. These HMDs, including HTC Vive [29], Sony PlayStation VR [104], Oculus Rift [133], and Google Cardboard [59], allow viewers to rotate their heads or roll their eyes to see different viewports of each 360° video. These 360° videos enable novel applications. For example, Facebook Spaces [78] allow users to socialize with their friends from different places in a shared virtual room. In real estate, potential buyers can virtually walk through new houses. For retailers, customers may buy clothes online with virtual fitting rooms to save commute time and expense. In education, HMDs may capture more students' attentions at lower costs. For example, astronomy students may learn how the universe works without being sent to outer space. Similarly, medical students may be trained for surgery with HMDs; indeed, the first remote human operation was performed in 2016 [56]. These emerging applications will change the landscapes of various fields, such as the military, education, real estate, retail, entertainment, healthcare, and communications.

Although 360° videos are useful in many novel applications, acquiring, transmitting, and displaying 360° videos is no easy task. For example, acquiring 360° videos often dictates real-time stitching videos from multiple cameras. The stitching algorithms require careful calibration and time-consuming feature matching, and thus 360° video acquisition is vulnerable to stitching artifacts and excessive delay, which result in degraded qualities of 360° videos. In addition, 360° videos typically contain many more pixels compared to conventional videos and are encoded into larger files (or say at higher bitrates). Transmitting 360° videos over bandwidth-constrained networks, therefore, is quite challenging and may result in an inferior user experience due to a long initial buffering time, low and fluctuating video fidelity, and frequent playout interrupts. Furthermore, 360° videos are projected from spherical surfaces to a 2D flat surface before being encoded with standard video codecs. Before 360° videos are displayed, the 2D flat surface is projected back to the spherical surface to be viewed in HMDs. The projections may lead to additional shape and texture distortion, whereas the HMD viewers may suffer from sickness due to long response times and insufficient resolutions. The preceding sample challenges in 360° video streaming are by no means

Table 1. Terms and Synonyms Related to 360° Video Streaming

Term	Definition	Synonym
360° Video	A video that captures lights from all directions to a camera	Omnidirectional video, panoramic video, zoomable video, spherical video
Virtual Reality (VR)	A human-made virtual environment with artificial objects, which can be explored by users	
Augmented Reality (AR)	Similar to VR, but AR combines some virtual objects with real environments, where users are currently	Mixed Reality (MR)
Viewport	A portion of videos that are visible to a 360° video viewer	Field of View (FoV), Region of Interest (RoI), fixation
Projection	A mathematic transform to convert videos from one space (e.g., spherical space) to another space (e.g., 2D planar plane)	Mapping
Head-Mounted Display (HMD)	A display mounted in front of a viewer's eyes for an immersive video viewing experience	VR headset

exhausted, and researchers around the globe have been working on the challenges from various aspects. For example, 360° videos are often encoded as rectangular *tiles*, which are independently decodable. Tiled streaming allows skipping tiles that are unlikely to be watched to save bandwidth consumption.

In this article, we collect, classify, and present the latest research on 360° video streaming. We survey off-the-shelf hardware at the time of writing. We also discuss crucial open challenges of 360° videos from different angles, which may shed some light on future research directions. As an emerging field, the terms used in 360° video studies are not consistent. To avoid confusion, Table 1 shows the definition of some terms used in this article and their synonyms (if any). When presenting research works in the rest of this article, we may adjust the terminology used by those works in the literature to increase reliability.

1.1 Scope and Related Work

This article focuses on the emerging research on 360° video streaming, which to the best of our knowledge has not been thoroughly surveyed in the literature. The closest work is Lee et al. [98], which surveys Free Viewport Video (FVV). Compared to our work, the authors consider general FVV systems. For example, they discuss both *inward* FVV, where a foreground object is shot by a camera array pointing at it, and *outward* FVV, where a background scene is shot by camera(s) pointing at all directions. In addition, they consider three 3D scene representations: image-, depth-, and model based, where image-based representations contain RGB videos from dense camera setups, depth-based representations add depth video for scene synthesis within limited ranges, and model-based representations build 3D models for scene synthesis from arbitrary camera locations and orientations. Modern 360° video streaming systems, surveyed in our article, are mostly image-based outward FVV systems with HMDs. Readers interested in more general FVV systems should refer to Lee et al. [98].

Domanski et al. [41] present the development of FVV supports in the MPEG-I project, where *I* stands for immersive media. Their work considers single-view 360° videos, which allow viewers to watch in all directions at a fixed camera location. Single-view 360° videos offer three Degrees-of-Freedom (DoFs)—yaw, pitch, and role, which can be changed by HMD orientations derived from

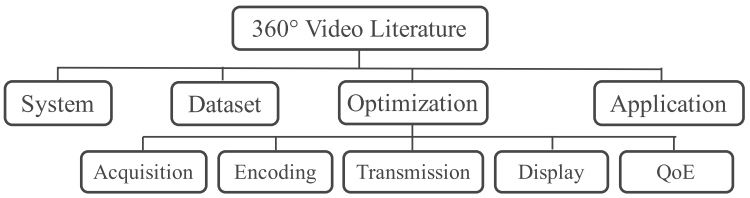

Fig. 2. Classifications of 360° video literature.

sensor readings. Single-view 360° videos, however, do not provide depth perception or binocular disparity, which lead to degraded immersive experience. Multi-view 360° videos contain 360° videos captured by multiple displaced (and fixed) cameras so that viewers' two eyes can see different 360° videos for more accurate depth perception and binocular disparity. The standardization efforts of MPEG-I on the aforementioned technologies are also summarized in their work.

The 360° video streaming can be leveraged in immersive communications. Apostolopoulos et al. [12] and Abbasi and Baroudi [1] present immersive communication systems that support the visual, auditory, haptic, smell, taste, and other senses. Both immersive audio and video communications are discussed in these two works, whereas 360° video streaming can be leveraged in immersive video communications. Furthermore, 360° video streaming may also be used in VR applications. In VR, the whole 3D scenes are synthesized by computer graphics; in contrast, in AR, the synthesized virtual objects are mixed up with the nature scenes. AR applications are generally more complex than VR ones, and thus AR applications attract increasingly more attention from the research community. Readers interested in (i) general AR technologies are referred to Krevelen and Poelman [93], (ii) a mobile AR survey are referred to Chatzopoulos et al. [24], and (iii) AR games are referred to Thomas [181] for the latest developments on VR-like applications.

1.2 Classifications and Article Organization

Figure 2 summarizes the classifications of 360° video works in the literature. The first level consists of four classes:

- *System* works present the designs or performance measurements of complete 360° video systems with acquisition, transmission, and display components. These works do not concentrate on the optimization of individual components; instead, they consider overall system designs.
- *Dataset* works provide inputs on viewer demographic information, video characteristics, and viewing behavior, and they enable fair comparisons among different solutions. The viewer demographic information may contain gender, age, and job titles, among others. The video characteristics capture the projection model, encoding bitrate, resolutions, and so on. The viewing behavior contains viewing trajectory, experience, and so on.
- *Optimization* works enhance individual components in the complete systems and can be further divided into five subclasses: (i) *acquisition* focuses on the optimization of 360° video acquisition, such as the placement and calibration of multiple cameras, and video stitching with the captured videos; (ii) *encoding* presents the optimization of encoding process, including projection models and video codecs; (iii) *transmission* optimizes on data transmission, such as tiling strategies, bitrate allocations, and buffer management; (iv) *display* focuses on different ways to present 360° videos, including conventional monitors and HMDs; and (v) *quality assessments* studies various factors that may affect user experience when viewing 360° videos. We summarize the optimization techniques proposed in these works.

- *Application* works talk about novel applications of 360° videos. For example, VR and AR are among the major applications of 360° videos, and they have gathered much attention in academic, entertainment, educational, and medical fields.

In Sections 2 through 5, we survey the literature in the four first-level classes. These are followed by the conclusion in Section 6. Due to space limitations, we present off-the-shelf hardware in Appendix A, the open challenges in Appendix B, and the summary tables in Appendix C.

2 SYSTEMS

We present the 360° video streaming systems in the literature, followed by the general pull- and push-based 360° video streaming framework.

2.1 Existing Systems

Multiple studies develop 360° video systems for various purposes, such as demonstrating the practicality or evaluating the optimization ideas. Table 4 in Appendix C summarizes and compares the prototype systems proposed in the literature. Some 360° video systems render videos stored in local storage spaces. For example, Petry and Huber [151] present a 360° display system that allows viewers to interact with the video through an HMD and a mounted gesture recognizer. In particular, the viewers are able to play, pause, forward, and rewind the video by performing different mid-air hand gestures. Alface et al. [9] propose a system that is able to handle 16K videos by only processing the required pixels in the viewer's viewport. In particular, they compose a moving 4K canvas that always keeps the viewport at the center of the canvas. In this way, the processed video is always in 4K resolution, which reduces the demands for processing power. Anderson et al. [11] develop a more comprehensive 360° video system that contains capturing, stitching, and rendering components. They first mount cameras on a rig so that all light rays from the viewer's eyes are recorded as 360° stereo videos. All 16 videos are stitched using their proposed optical flow and composition algorithms. Ferworn et al. [50] develop a special-purposed 360° video system for dogs to perform urban search and rescue. In their proposed system, a dog wears multiple cameras, which keep capturing video frames. These captured video frames are uploaded to a Personal Computer (PC) after the mission is over. Next, the PC analyzes videos, performs tracking, and stabilizes frames for better-quality 360° videos.

There are more systems coming with Internet streaming supports. Gaemperle et al. [58] adopt a multi-camera system with image blending algorithms to capture 360° videos. The video is then distributed through the server to the client, and the HMD viewport is reconstructed by the client. Several studies try building low-cost streaming systems for 360° videos. Canessa and Tenze [21] develop a 360° video Real-time Transport Protocol (RTP) streaming system on Raspberry Pi with a fish-eye camera module and several open source packages, such as FFmpeg, OpenCV, and MPlayer. However, the resolution of their system is only about 360p. Choi and Jun [27] further consider multiple camera-equipped Raspberry Pis. They send the images over a network to a PC for stitching. Several optimization tools are employed to improve the system performance, including simplified algorithm and multi-threading. They try different blending methods proposed in the literature and quantify their performance. Jiang et al. [82] build a 360° video streaming system for power consumption measurement. They compare 360° video streaming systems to conventional 2D ones. The measurement results show that viewport generations consume the most power due to the high computation overhead for viewports. The network transmission consumes the second most, followed by the screen displays and video codecs. Based on their observations, they propose several power-saving approaches, such as viewport-based streaming and edge-based rendering.

Several other systems further consider tiling for improving the performance of encoding or streaming 360° videos. Ochi et al. [131, 132] build a 360° video streaming system that streams high-bitrate tiles to the viewer's viewports and low-bitrate tiles to other parts, for reduced bandwidth consumption. However, the latency of their proposed streaming system still has room for improvement. Qian et al. [154] present a streaming system over cellular networks, which only streams the tiles in the viewport based on head movement predictions. Xie and Zhang [202] present an interactive 360° streaming system over cellular networks. They develop a conservative compression strategy to keep the quality in the viewport more stable, even when the variation of viewports is high. In addition, their proposed system monitors the buffer occupancy of the uplink for congestion detection. Once the congestion is detected, the encoding bitrate is adjusted to maintain the video quality. Schafer et al. [164] develop a 360° video capturing and streaming system. Their system splits videos into several subvideos and performs stitching in the compressed domain so that the client only needs to decode one stream.

Different from the preceding systems, several studies split the videos into equal-size tiles and store the tile information, such as resolution and position, in metadata files [51, 102]. Feuvre and Concolato [51] apply tiling to general 4K videos with unequal tile quality levels to achieve viewport-aware adaptive streaming. Ozcinar et al. [139] also develop an algorithm to select the representation of each tile to achieve viewport-aware adaptive streaming in their proposed system. Lo et al. [106] build a streaming system and compare the performance of transmitting all tiles versus visible tiles only. They further study the impact of tiling, such as coding efficiency and tiling overhead. Graf et al. [60] build a tiled-streaming system considering different streaming strategies. Kim et al. [88] develop a streaming system for 360° video in virtual spaces. In particular, they adopt virtual cameras in Unity and Unreal engines. Scenes from 12 virtual cameras are captured and stitched to generate the 360° videos. The videos are then tiled and segmented for adaptive streaming through a Content Delivery Network (CDN). Nasrabadi et al. [119] exploit Scalable High-efficiency Video Coding (SHVC) to further adaptively streaming 360° videos with multiple layers. The base layer of all tiles are prefetched to avoid video stalls, whereas the enhancement-layer tiles in viewports are transmitted with the residue bandwidth. Combined with state-of-the-art network technologies, Petrangeli et al. [148] leverage HTTP/2.0 and OpenFlow to reduce latency and avoid network congestion, respectively.

Similar to 360° videos, live broadcast events, such as soccer or basketball games, are often streamed as panorama videos. These videos contain wider horizontal viewing angles (which may be less than 360°). Considering that most live events have a single Region-of-Interest (RoI), tiles are also used in panorama videos to support zooming [57, 80, 90, 91, 127]. Inoue et al. [80] develop a tiled-streaming system based on Multi-View Coding (MVC) to support tiles. A rate-quality mapping table is used for determining the transmitted viewports to maximize the visual quality under restricted bandwidth. Kimata et al. [90] propose an interactive panorama video streaming system allowing viewers to control their viewports for high-quality videos and sounds. Redundant viewports are also adaptively streamed to guarantee smooth and fast switches of viewports. Some extensions of this system are proposed in Kimata et al. [91]. They extend the system to support mobile and multiple devices. In particular, the viewers are allowed to interact, such as zoom in/out, with their handheld mobile devices while watching high-resolution panorama videos on larger screens. Niamut et al. [127] propose an end-to-end panorama streaming system with acquisition, transmission, and display components. In their system, the scenes are captured with multiple representations (i.e., resolutions and frame rates). Some analyses, such as saliency detection and person tracking, are performed for automatic camera selection for serving a larger number of viewers. The videos are encoded with multiple representations and are streamed to the clients. Their system supports several gesture interactions to control the viewports and the

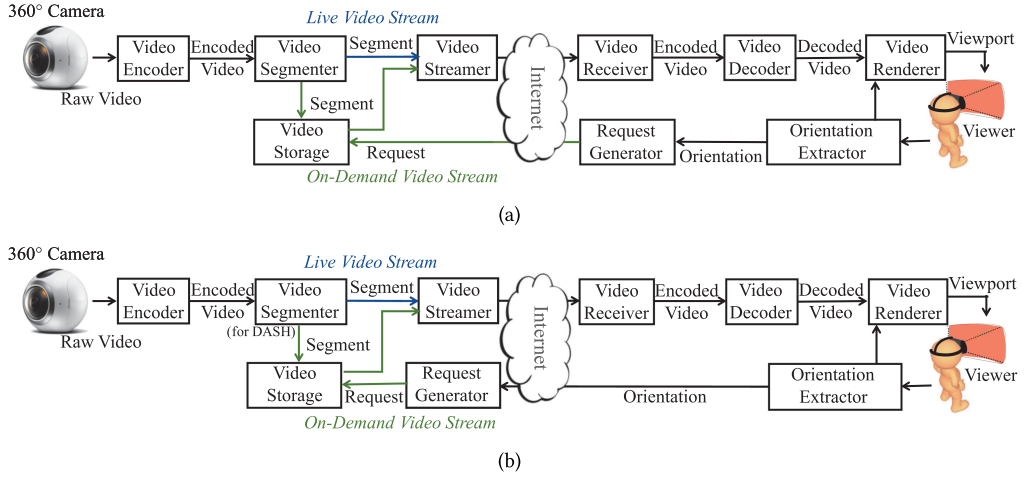

Fig. 3. General 360° video streaming systems: pull based (like HTTP/1.1) (a) and push based (like RTP) (b).

playbacks. For generating virtual views for individual clients, Gaddam et al. [57] propose to leverage tiling for quality allocation and GPUs for rendering acceleration.

The 360° video streaming systems mentioned previously contain different subsets of computation and networking components: from playing local 360° video files to streaming 360° videos over the Internet. Insights gathered in these studies are beneficial to engineers who plan to build similar systems, and to researchers who plan to evaluate their proposed solutions in real testbeds.

2.2 General 360 Video Streaming Framework

Figure 3 shows two 360° video streaming frameworks: (i) pull based (client driven) (Figure 3(a)), which puts the intelligent components at the client to determine the requested video segments, and (ii) push based (server driven) (Figure 3(b)), which, in contrast, puts them at the server to decide which segments to push to the client. We describe the functions of individual components in Figure 3 as follows:

- *Video encoder* encodes the captured 360° videos. It may further support tiling for partially streaming and rendering to save bandwidth consumption.
- *Video segmenter* splits the encoded video into segments, where each segment contains a few consecutive video frames and lasts for a few seconds.
- *Video storage* stores the encoded video on the server, which is used for on-demand video streaming. It may store several versions of the encoded video at different quality levels to support adaptive video streaming.
- *Video streamer* is responsible for sending the encoded video to the client.
- *Orientation extractor* computes viewer orientations using inputs from HMD sensors or other devices.
- *Request generator* generates requests of videos or video segments. It is placed at the client in pull-based systems and at the server in push-based systems. It is usually the core component that makes decisions for optimizing the streaming system. For example, a bitrate allocation algorithm can be implemented in this component for generating video segment requests based on network conditions.
- *Video receiver* receives the video streamed from the server.
- *Video decoder* is responsible for decoding the encoded videos.

- *Video renderer* computes the viewports according to the viewer's orientation. It may also convert among 360° video projection models [45], such as equirectangular, equalarea, and cube.

The interactions among the pull-based components in Figure 3(a) are as follows. At the server side, the 360° camera sends a 360° video to the video encoder. It compresses this video with or without tiling support and sends the encoded video to the video segmenter. The video segmenter splits the received video into temporal segments and either (i) directly sends the segments to the video streamer for live video streaming or (ii) stores them in the video storage for on-demand video streaming. At the client side, the orientation extractor keeps recording the viewer's orientation, which is computed from the sensor readings of the HMD.[1] The request generator considers the current network condition and may take the viewer's orientation as input to generate the requests. For example, the adaptive streaming system can request for video segments or frames at different quality levels, and the tiled-video streaming system may further skip some tiles based on the viewer orientations. The video streamer at the server side then streams the encoded videos from the video segmenter (live video) or from the video storage (on-demand video) to the client. The video receiver passes the received video to the video decoder. The video decoder then decodes the encoded video. The video renderer renders the decoded video according to the viewer's orientation to the viewer. The push-based components in Figure 3(b) are similar, except the request generator is at the server side.

The pull- and push-based systems have diverse pros and cons. The pull-based streaming systems often adopt HTTP/1.1 protocol. This protocol supports dynamic adaptive streaming using short segments, which typically are a few seconds long and independently decodable. Such an approach is better known as Dynamic Adaptive Streaming over HTTP (DASH). The pull-based streaming systems are not affected by the Network Address Translation (NAT) traversal problems. In addition, it is convenient to reuse the WWW infrastructure, including servers, caches, and CDNs. In DASH streaming, the media contents are encoded into various versions and stored on the server beforehand to adapt to varying and dynamic networks. This consumes a large amount of storage space. In contrast, push-based streaming systems, which often adopt RTP as the transmission protocol, stream the media content to the clients without waiting for the requests. This leads to lower latency compared to pull-based streaming. However, it requires the streaming server to keep track of the states of potentially a large number of clients. Without a reliable protocol like TCP, the push-based streaming systems may result in inferior video quality due to network impairments, such as insufficient bandwidth, packet loss, and NAT traversal issues. Based on the pros and cons described earlier, we believe that the pull-based streaming systems are more applicable for *presentational* video streaming, such as YouTube and Netflix, where a few seconds of one-way delay is acceptable [2]. In contrast, the push-based systems are more suitable for *conversational* video streaming, such as video conferencing, where the ultra-low latency is required for the high interactivity. Requiring both high video quality and ultra-low latency, streaming 360° videos over these two types of systems needs additional optimization tools, such as viewport prediction or resource allocation, which will be discussed in Section 4.

We have introduced the general framework of pull- and push-based 360° video streaming systems. Researchers and practitioners may start from the general frameworks and add specialized components to meet their needs.

[1] Other input devices are possible when displays other than HMDs are used.

3 DATASETS AND OPEN SOFTWARE

In this section, we present several datasets of 360° videos. We also survey several open source projects, which may be used to collect datasets and build 360° video streaming systems.

3.1 Datasets

Datasets allow researchers and developers to generate reproducible simulation and experiment results for fair comparisons among different solutions. Compared to conventional video datasets, 360° video datasets must contain both video content and user behaviors. Otherwise, the 360° video datasets cannot be used for evaluations, because the performance of 360° video streaming depends, e.g., on the user viewports. Several 360° image datasets have been published in the literature. Rai et al. [157] collect a dataset with 60 360° images, each watched by at least 40 subjects. Subjects are told to navigate freely through the 360° images from any orientation using eye-tracking–enabled HMDs for 25 seconds, and both head and eye movements are recorded. They analyze the collected data and find that the viewers constantly move their gazes to explore the images.

360° video datasets are also available, such as that of Lo et al. [105], which download and analyze 10 representative 1-minute 360° videos from YouTube. The authors groups these 360° videos in two dimensions based on (i) natural or computer-generated videos and (ii) fast and slow motions. Fifty subjects are recruited to watch each video using an HMD. The dataset contains the raw and processed data collected from video frames and HMD sensor readings. Corbillon et al. [35] collect head motions of 59 users viewing five 70-second 360° videos using an HMD. They also implement an open source C++ program using OSVR API [158] to collect the viewer orientations. Analysis on head movements is presented by the authors, which includes the maximum and average angular speeds under different video segment lengths. Under different usage scenarios, Wu et al. [194] present their datasets for (i) Video on Demand (VoD) and (ii) live events. The subjects who watch the VoD videos are free to look around and navigate through the virtual environments. In contrast, the live events often have salient object(s) at the center of the recorded events. Thus, viewers usually only look at the center. The subjects who watch live events are given a quiz after finishing each video to check their memory on the content. Fremerey et al. [54] develop a Python tool for collecting orientations when a viewer is watching 360° videos using an HMD. They use the developed software to collect a 360° video viewing dataset, which consists of 48 viewers and 20 videos that are 30 seconds in duration. They also collect the Simulator Sickness Questionnaire (SSQ) scores from each viewing session. Some analysis on sickness perceived by different genders are discussed. With eye-tracker equipped HMD, David et al. [39] collect both the head movements and the eye movements on viewing 360° videos. Their eye tracker tracks the viewer's gazes at 250Hz with 0.2° precision. There are 57 viewers and 19 videos in their dataset. With the eye movements, the dataset is useful for developing saliency model at gaze level, which may be beneficial to foveated rendering and streaming. These datasets [35, 39, 54, 105, 157, 194] are useful for researchers and practitioners for (i) optimizing 360° video streaming systems, such as through viewport-based Rate-Distortion Optimization (RDO) and visual attention modeling, and (ii) developing novel applications, such as user identification and crowd-driven camera movements. Later, Table 5 in Appendix C summarizes the preceding datasets.

3.2 Open Software

Recently, more and more open software projects have been developed for or have been extended to support 360° videos. We classify them into four main categories: (i) codec, (ii) packager, (iii) player, and (iv) platform. We survey well-known software projects as follows:

- *Codec.* More and more codec projects are extended to support 360° tiled videos following the video coding standards, such as High-Efficiency Video Coding (HEVC) and SHVC (more details can be found in Section 4.2):
  - *Kvazaar [187]* is developed in C with tiling support, which runs in real time for 4K videos on workstations.
  - *HM [42]* and *SHM [66]* are reference software projects for encoding and decoding in HEVC [177] and SHVC [19], respectively. Both of them support tiling.
- *Packager.* The packager can be used for manipulating container files, such as AVI and MP4:
  - *MP4Box [142]* allows users to package the tile-encoded HEVC bitstream into MP4, where tiles of each frame are represented in different tracks. Individual tiles can be removed without influencing the decoding process. In addition, MP4Box also supports DASH and splits each tiled video into temporal segments. More details on DASH streaming of 360° videos are given in Section 4.3.
- *Player.* Existing 360° video players may support conventional 2D displays and HMDs. The main new features compared to traditional video players are the supports of viewports and tiled videos. The following is list a few representative players:
  - *MP4Client [143]* is a multimedia player included in GPAC [144]. MP4Client supports requesting and displaying segments and tiled segments from an HTTP server or local storage. It has been extended to only requesting and displaying the tiles that are most likely to be viewed by the viewers to save bandwidth consumption [106].
  - *ExoPlayer [79]* is a media player for Android developed by Google, which supports DASH, smooth streaming, and adaptive playback. It is well documented and extensible, and thus allows researchers to extend it for 360° video streaming [149].
  - *WebVR [193]* is an open specification for Web browsers. It allows users to watch 3D 360° videos by visiting WebVR Web site using smartphones. Moreover, users can wear Google Cardboard [59] to experience 3D VR environments.
- *Platform.* Several 360° video platforms are developed and extended in various aspects:
  - *360Lib [81]* provides tools to convert 360° videos among various projection models [45], such as equirectangular, cubemap, and octahedron. It depends on HM reference software for projection conversion in the compression domain. Furthermore, the computations of several objective quality measures for 360° videos are also supported. Similarly, Facebook [25] and Corbillon [32] also provide tools for projection conversion.
  - *OpenTrack [61]* is an application implemented in C++, which tracks head movement. In particular, OpenTrack logs the HMD positions and orientations in *x, y, z, yaw, pitch*, and *roll* over time, which are useful for VR applications.
  - *Open Source Virtual Reality (OSVR) [158]* is a VR platform supporting several game engines, systems, and devices, including Unity, Android, Oculus, and HTC Vive. It is extensible for diverse purposes, such as dataset collection and adaptive rendering [35].

4 OPTIMIZATION

In this section, we present various optimization approaches in literature, from cameras to encoding, transmission, and display. We also discuss QoE optimization at the end of this section.

4.1 Acquisition

Capturing and pre-processing (e.g., stitching and projection converting) images and videos from heterogeneous cameras for 360° videos are the starting points of offering an immersive experience, which requires unique optimization approaches. Schreer et al. [166] discuss *format agnostic* production, which jointly leverages videos from multiple camera sensors at different temporal

and spatial resolutions for 360° videos that support diverse applications from mobile devices to wide-angle displays. Different from conventional production systems, format-agnostic production systems have no fixed frame size and support virtual cameras controlled by directors or viewers. The techniques and experiments presented in their work are valuable to researchers working on ultra-high-resolution 360° videos.

Hardware cameras have been built for 360° videos, which could be based on either *single camera* or *camera arrays*. Single camera approaches are less expensive and are immune to the artifacts due to stitching. For example, Krishan and Nayer [94] present a fish-eye camera for 360° images, which attaches a curved mirror to a fish-eye lens. They propose an algorithm to stitch the two images from fish-eye lens and mirror into a seamless 360° image. Couture et al. [186] present a stereoscopic 360° video capturing system using a pair of rotating commercial-grade video cameras. The authors employ full video frames for stereo motion alignment in the temporal domain. They do not stitch adjacent video frames; instead, they blend the video frames. This is because blending seams are continuous and thus are harder to spot. Aggarwal et al. [6] propose using a filter mirror for stereoscopic 360° videos, where the light rays into left and right eyes are captured by a single camera. The filter mirror's surface is parameterized, and the parameters are mathematically optimized for maximizing the quality of captured images. Compared to (i) multi-camera and (ii) moving camera systems, the filter mirror approach reduces the device size, simplifies the synchronization/calibration, and works on commodity cameras. Belbachir et al. [16] attach a pair of linear light sensors to a rotation platform and generate stereoscopic 360° videos in real time. This is achieved in three steps. First, two linear dynamic vision sensors, which are designed for capturing asynchronous images, are mounted on a high-speed rotating disk. Second, an algorithm is proposed to reconstruct intensity images using the sensor data. Last, stereoscopic 360° videos, anaglyph images, and depth images are created from the high-dynamic-range cameras at high frame rates.

Camera arrays often have higher total resolutions. For example, Foote and Kimber [52] build a camera array by attaching off-the-shelf cameras along the wall of a cylinder. This automatic camera system, called *FlyCam*, generates seamless videos by fusing data from several nearby cameras. They present an approach for stitching and anti-distortion to generate 360° videos. They also present motion analysis and camera control algorithms. Their lightweight methods for 360° videos could be used in several applications, such as teleconferences, lectures, and meetings. Afshari et al. [4] construct a camera array by adding cameras on a sphere, which is inspired by flying insects. With FPGA, they design a system for real-time videos captured from up to 30 cameras. The cameras are put on a spherical pointing at different directions. In addition, they present a 360° video reconstruction algorithm, its configurations, and a real FPGA implementation. Cogal et al. [30] present a similar camera array with 44 high-resolution cameras, achieving a total resolution of 220 Mega-Pixels (MP). The cameras are optimally arranged on a sphere for 360° × 100° Field of View (FoV). The detailed hardware design of a 360° video capturing and recording system is presented in their work. It is reported to achieve 21.6MP at 30 frames-per-second (fps) and 82.3MP at 9.5fps in real time.

Images and videos from cameras need to be pre-processed before being useful. Stitching, which merges multiple images/videos into a higher-resolution one, is probably the most commonly seen pre-process. Countless works propose ways to improve the quality of image stitching. For example, Zomet et al. [217] develop optimization algorithms to maximize the stitching quality. The stitching quality is defined as a function of (i) similarity between the output and input images and (ii) invisibility of the artifacts along the seams of stitched images. Several cost functions are introduced and tested, whereas the seam visibility is quantified in the gradient domain. Their proposed solution minimizes the adopted cost function and can be used to generate 360° images and object

blending, among other applications. Xiong and Pulli [203] also solve the stitching problem for images. They concentrate on minimizing artifacts appearance on the seam of two stitched images, due to the color and luminance difference between them. This is achieved by color matching across input images with techniques such as color correction and image blending. The resulting 360° images show high color consistency and smooth color transition. The proposed solution is implemented and evaluated on smartphones for 360° images with visually appealing results. Xiao and Wang [198], in contrast, propose an algorithm to generate panorama images directly from fish-eye images. In particular, they formulate the projection conversion equations and map the points on fish-eye images to the panorama using the backward mapping approach.

Video stitching is more challenging and receives increasingly more attention. For example, Lin et al. [103] present a framework to stabilize and stitch videos captured by freely moving cameras. Each stitched video is generated by first identifying the camera paths and constructing the 3D scene. Next, a new camera path is built by smoothing all of the input camera paths. This new camera path is then employed to warp the input videos into a stitched one. Their framework has various applications, such as stitching 360° videos, social media content creation, and multi-robot vision. Jiang and Gu [83] design a spatial-temporal content preserving stitching approach for videos. Their proposed algorithm adopts warping to stitch imaging and stabilize videos, but with fixed camera positions. The algorithm consists of two steps: (i) aligning frames from multiple videos and (ii) finding spatial-temporal seams. The first step aligns frames from different videos in a temporally consistent manner. The second step is transformed into a 3D graph cut problem, where the weights are functions of objects and motion to maximize the stitching quality.

Perazzi et al. [147] capitalize local warping to remove parallax from multiple unstructured cameras for 360° videos. Their proposed algorithm is unique for three reasons. First, they propose a patch-based error measure as a function of image gradients, which is used to maintain content similarity between input videos and the resulting 360° videos. Second, they design a method to analyze the relative camera positions, scene content, and order of pairwise warping, so as to improve the warping quality. Last, they introduce a weighted warping procedure for the final 360° videos, which mitigates the temporal artifacts. Because stitching videos is computationally intensive, they proposed to employ GPU to accelerate video stitching. Calagari et al. [20] build a similar system for sports 360° videos, but with videos from regular cameras that have been installed in the sports fields/courts. They first generate a static 360° image, which serves as the background image when some areas are not covered by any camera. For example, spectators are not often covered by cameras nor are they important to the sports games. Then, the authors derive player motions from the main (center) camera and apply various techniques such as warping to remove the parallax and align videos from all cameras. Last, the resulting video is blended with the background image. Silva et al. [169] connect multiple (four or six) GoPros to a computer via HDMI cables and capture cards. The captured video frames are loaded to GPU card memory. The last four (or six) video frames in the queue are then stitched with vertex and fragment shaders. The resulting video frames are encoded by the GPU and streamed by the CPU. Lee et al. [99] propose approaches to solve issues of creating 360° videos from a structured camera array. These issues include the misalignment between two adjacent cameras and the relatively low resolution of the final 360° video. First, they leverage a moving checkerboard to perform calibration for estimating various settings of individual cameras. The depth disparities are then computed through feature extraction to recover 3D points for minimizing the parallax artifacts. Second, they propose sampling more important regions at higher frequencies and less important regions at lower frequencies. Their results show that their proposed approach achieves higher rendering quality and preserves more content details than the equirectangular projection.

Table 6 in Appendix C summarizes the key literature of 360° content acquisition. In terms of capturing, the studies can roughly be classified based on the camera composition: (i) a fixed camera with a mirror, (ii) two rotating cameras, and (iii) a fixed camera array. The images/videos captured by most of these settings need to be stitched to form a final 360° image/video. Among them, the camera composition with a camera array is more popular. This is because the calibration based on the known relative camera positions is easier and more precise, which in turn leads to better stitching quality. The studies on pre-processing can therefore be classified into two groups based on the awareness of the camera positions. Generally, the stitching process with camera positions achieves better performance than the stitching process solely based on content features.

4.2 Encoding

Typical 360° videos are spherical videos projected to rectangle videos in high resolutions. The viewports accessed by viewers are essentially subsets of video frames. Encoding 360° videos, therefore, is very challenging and requires advanced supports from video codecs. Heymann et al. [67] extend the MPEG-4 codec to divide the 360° video into independently encoded subvideos for the support of decoding and rendering parts of videos. Rerabek et al. [160] propose to encode 360° images into 360° JPEG files that can be decoded using the legacy JPEG decoder for backward compatibility. More specifically, their encoder first estimates the viewer viewports using saliency maps and encodes the viewports of 360° images using the regular JPEG encoder. The encoder then compresses the whole 360° image also using JPEG and embeds the resulting bitstream as the metadata. By doing so, 360°-capable decoders and projectors can render the 360° images, whereas legacy JPEG decoders render the viewport images.

HEVC [135] supports Motion-Constrained Tile Set (MCTS), where tiles are disjoint rectangular regions that can be independently decoded. Tiles allow (i) parallel decoding for a decoder speedup to cope with high resolutions and (ii) random decoding of dynamic viewports. Tiles, however, impose constraints on the encoding process, which needs to be considered carefully. In particular, MCTS motion dictates constraint among tiles, which reduce the coding efficiency because motion vectors do not go across the tile boundaries. Therefore, there exists a critical tradeoff between the coding efficiency and the tiled streaming flexibility, which can be controlled by the number of tiles. More details on the HEVC standard are given in Sullivan et al. [177], whereas the details about the MCTS supports in HEVC can be found in Misra et al. [114]. It is reported that (i) HEVC results in a 50% rate cut at similar visual quality [177] than AVC and that (ii) tiles achieve up to a 5.5% luminance bitrate reduction [114] than regular slices. Due to their superior coding efficiency, HEVC and its tiling support are widely used in 360° video systems. HEVC standard does not specify the precise optimization algorithms used at the encoder side. Among existing open source HEVC codecs, Kvazaar [187] is developed in C language and provides an option to be optimized in Assembly. Kvazaar implements various coding tools defined in HEVC, which enable parallelization on multi-core CPUs and hardware acceleration. It supports three parallel processing approaches, including tiled encoding, and thus can be leveraged by 360° video testbeds.

Several optimization techniques on video codecs for 360° tiled videos have also been studied, which can be classified into two groups: (i) parameter selection and (ii) stream rewriting. In parameter selection, Sanchez et al. [162] consider the problem of optimizing the tile dimension to minimize the bitrate of the viewports. They propose a model using spatio-temporal activity metrics to achieve optimal tiling of the 360° videos for streaming. Their evaluations show that the proposed method results in higher coding efficiency compared to static tile size. Khiem et al. [86] adapt the encoding parameters of different regions in zoomable videos, based on the historical viewer access patterns. They propose two ways to dynamically crop viewports. The first way is to merge the tiles that fall into a single viewport, which is suitable for tiled encoding. The

second way is to limit the motion search range of the referenced macroblocks, which is suitable for conventional encoding.

In stream rewriting, Sanchez et al. [55] propose a compression domain algorithm to rewrite multiple HEVC tiles into a single bitstream of the current viewport, which can be decoded by a single hardware decoder. They also propose a solution to reduce the bandwidth consumption when users switch their viewports. This is a critical challenge for 360° video systems, as high bandwidth consumption may lead to playout interruptions. Their core idea is to insert redundant reference pictures to compensate the temporal tiles that may not be streamed based on the viewers' viewports. Skupin et al. [173] propose the technique of dynamic tiling, which aims to adjust the resolution on the fly according to the viewer's viewports. They encode each video into high and low resolutions. They then vary the ratio of high- and low-resolution tiles over time so that the viewer's viewports are sent in high resolution while other portions are sent in low-resolution tiles. Sanchez et al. [206] apply a similar approach on SHVC, which is the scalable extension of HEVC, to support multiple resolutions. The rewritten bitstream can be decoded by a single hardware decoder. Their proposed solution reduces the bitrate when switching among viewers' viewports. They utilize the concept of open Group-of-Picture (GoP) for better compression efficiency, because closed GoP may suffer from frame loss during decoding. Their solution supports seamless playback with and without the enhancement layers.

Son et al. [174, 175] implement MCTS in HM and SHM, which are the reference software of HEVC and SHVC, respectively. To extract and transmit tiles independently, they propose solutions to correct the motion vectors when only decoding partial tiles. In HEVC, the encoder performs intra-prediction when the selected tiles temporally refer to other tiles. In SHVC, both the base layer and the enhancement layer of previous frames are searched if a tile refers to tiles at the same location; otherwise, only the base layer is used for reference. Extra metadata are generated by the encoder so that the decoder can independently extract and decode the selected tiles. Zare et al. [211] propose an OMAF-compliant mixed-resolution packing arrangement for 360° videos, which guarantees that the viewport comes from 6K video while other regions are covered by lower-resolution content. In particular, they spatially split the video into pole and other regions. The pole regions are encoded in 4×1 tiles at 3K or 1.5K resolutions, whereas other regions are encoded in 8×1 tiles at 6K and 3K resolutions. Their proposal saves up to 32% bandwidth consumption.

MVC standards are also adopted for encoding 360° videos [80, 90, 91]. For example, Inoue et al. [80] propose to utilize MVC to jointly encode individual tiles (as different views). They build a rate-quality mapping table to select the best viewport that maximizes the visual quality under the limited bandwidth. Their proposed algorithm iteratively increases the bitrate of the worst-quality tile until the available bandwidth is used up. Kimata et al. [90, 91] use MVC in a different way. They reserve one of the views in MVC as a low-resolution thumbnail for the full frame. The other views are high-resolution viewports that may be viewed by the viewer. These views are merged into a single stream, which is then sent to the client. They also adaptively stream some redundant viewports to smooth out the viewport switches. Kimata et al. [91] further improve their proposed system to support mobile devices so that the viewers are able to interact using their mobile devices. For example, they can zoom in/out and pan the viewport using the sensors on mobile devices.

Table 7 in Appendix C summarizes the common codecs used for 360° video compression. Among these codecs, both HEVC and SHVC support MCTS, which allows tiles to be independently decodable. This, in turn, allows streaming sessions to better adapt to viewport switches without expensive transcoding. With both MCTS and multi-layer support, SHVC achieves higher flexibility and smooth quality degradation becomes possible. MVC enables combining a low-resolution full thumbnail with one or multiple high-resolution viewport regions into a single stream for efficient transmission.

Several studies aim to optimize the coding efficiency for 360° videos and images. Sauer et al. [163] consider the convex polytope projection and propose a solution to compensate the geometric distortion for better motion compensation performance. Their work is motivated by the observation that straight lines are bent at the border of two adjacent faces when 360° videos are projected to a polytope. Such shape distortion results in suboptimal motion compensation when some motion happens across the borders. To cope with this issue, the authors propose to extend each face by projecting the adjacent faces to it using homographies. This is to ensure straight lines remain straight after projection, so as to increase the motion compensation performance and reduce the bitrate. Li et al. [101] extend each face of the cubic projection to maintain texture continuity across face boundaries, so as to enhance motion compensation. They propose a padding method that projects the reference and current pixels on the same surface, before encoders performing motion estimation. Compression algorithms of non-rectangular images and videos have also been investigated. Tosic and Frossard [182] propose a 360° image compression algorithm that takes the geometric proprieties into considerations. The crux of their work is redundant storage of basic geometric shapes on a sphere. This 360° image is projected on a sphere and then passed into an iterative algorithm for a suboptimal solution of the weighted sum of a series of atoms. This is followed by an adaptive quantizer before being sent to the decoder. The resulting codec achieves superior performance at lower bitrates, where image geometry dominates image texture in terms of entropy. Youvalari et al. [209] consider compressing 360° videos that are pseudo-cylindrically projected. However, using pseudo-cylindrical projection leads to some problems, such as coding inefficiency and coding artifacts. They propose intra- and inter-frame coding tools for higher coding efficiency and mitigate coding artifacts.

Ozcinar et al. [138] select the tile bitrates in 360° videos. They formulate it as an Integer Linear Programming (ILP) problem, which chooses a subset of bitrates for each tile, to maximize a weighted sum of video quality and resource (storage and network) cost. The weights are heuristically chosen by the service providers, and the resulting problem is solved using a general ILP solver. Yu et al. [118] also optimize the bitrate of the tiles while jointly considering the sampling densities of the different part of the 360° videos. The rationale is that some projection models, like the equirectangular model, oversample the sphere videos close to north and south poles. Because the resulting problem is a variant of knapsack problems, the authors propose a suboptimal algorithm to first determine sampling density, followed by deciding the bitrates. Xie et al. [199], in contrast, consider a more general bitrate allocation problem, where tiles have different viewing probabilities. They formulate the problem as a mathematical optimization problem, with a weighted sum of (i) overall viewport distortion and (ii) inter-tile distortion variance. The distortion and bitrates of 360° video titles coded with diverse codec settings, such as Quantization Parameters (QPs), are empirically derived. Moreover, their optimization problem is an ILP problem, which is solved with general solvers. Xiao et al. [195] carefully derive an optimal projection model consisting of multiple variable-size viewports, which are smaller than one side of the cubic projection. They solve an optimization problem to minimize the total storage cost of all chosen viewports while guaranteeing that the sphere is fully covered. Their solution strives to achieve a good balance among (i) oversampled pixels, (ii) coding inefficiency, and (iii) overlapped viewport regions. Experiments from a real dataset and actual implementation show that their proposed projection model saves 16% of the storage space without degrading visual quality.

4.3 Transmission

Although not exactly the same as 360° videos, *zoomable* video transmission has been studied in the literature. Zoomable videos allow viewers to freely pan, tilt, and zoom their viewports for interactions. For example, Wang et al. [191] multicast tiled videos to heterogeneous viewers and

solve the bitrate allocation problem for individual time slots. Doing so over wireless networks is challenging because of heterogeneous resolution, viewport, and bandwidth requirements from individual users. They also propose to encode tiles at mixed resolutions and study the acceptable tile bitrate difference of different users. Mavlankar and Girod [112] use a Peer-to-Peer (P2P) network to multicast tiled videos and linearly perform viewport predictions for shorter latency and better video quality. In particular, they estimate the velocity of users' viewports using basic moving average. They also propose a more comprehensive prediction method that tracks the local features in thumbnails and motion vectors in the encoded bitstreams. The P2P network enables peers to share these data with one another in real time for lower server load. Although zoomable video transmission [112, 191] addresses some common challenges with 360° video transmission (e.g., only a small viewport of the extremely high-resolution video is viewed at any moment), zoomable video transmission does not take the 3D spheres of 360° videos into consideration.

Modern multimedia transport standards can be used in push- and pull-based transmission systems as mentioned in Section 2.2. Table 8 in Appendix C summarizes popular transport protocols, including (i) MPEG Media Transport (MMT), (ii) RTP, (iii) DASH over HTTP/1.1, (iv) DASH over HTTP/2, and (v) DASH over Quick UDP Internet Connection (QUIC). Used by push-based systems, MMT enables broadcast and synchronization services, and it is suitable for media distribution and multi-party conferencing calls. RTP supports both unicast and multicast services. It can be integrated with Real-time Transport Control Protocol (RTCP) and Real-Time Streaming Protocol (RTSP) in push-based systems. DASH works on top of several protocol stacks. The most common one is HTTP/1.1, where the server sends each segment after receiving the request from the receiver. HTTP/2 provides additional features: (i) multiplexed streams, (ii) prioritized streams, (iii) stream termination, and (iv) server push. QUIC protocol [97] has been adopted as an IETF standard, which runs on UDP and was designed to replace HTTP/2, TLS, and TCP protocols. In particular, QUIC offers several main features: (i) secured communications, (ii) multiplexed streams, (iii) prioritized schedulers, and (iv) low latency.

To support 360° video streaming, the MPEG DASH standard has included an amendment on Spatial Representation Description (SRD), which enables clients to request viewports of whole videos with 2D coordinates. The SRD standard is presented in Niamut et al. [128], along with several use cases of tiled videos. SRD expands Media Presentation Description (MPD) to define the relative spatial positions of tiles. It provides attributes like x- and y-axis coordinates, as well as width and height. DASH clients can determine which tiles to request. Like MPD, SRD only provides the spatial organization of content, without dictating how DASH clients leverage such information. Several use cases of SRD have been proposed and discussed, such as zoomable, mobile, and TV-wall displays, where tiled streaming provides additional flexibility. Concolato et al. [31] further discuss HEVC and ISO Base Media File Format (ISOBMFF) standards for encoding and encapsulating tiled videos for transmission. Combining SRD, HEVC, and ISOBMFF, a client may merge several tiles into a video stream, which is decodable by a decoder. Their evaluations show that the proposed approach incurs a minor streaming overhead when delivering the tiled videos compared to the non-tiled ones. Therefore, standard DASH clients can decode a subset of tiles in different quality levels based on the available bandwidth in dynamic networks.

Several authors share their experience of realizing standard-based 360° video transmission. Hu et al. [72] propose a 360° video broadcast system using MMT for broadcasting over the Internet. The authors divide and encode the 360° videos into multiple tiles. The encoded tiles are encapsulated into multiple MMT assets, which can be individually received by receivers. The authors employ MMT signaling messages to describe the spatial relationship of MMT assets. This allows receivers to subscribe the tiles in their viewports at a high bitrate and other tiles at a low bitrate. Canessa and Tenze [21] adopt RTP in their developed 360° video streaming system on Raspberry

Pi, where a fish-eye camera is adopted to capture 360° videos. Their considered video resolution is 352 × 288, which is much lower than the common 4K resolution of 360° videos. D'Acunto et al. [37] provide guidelines on realizing navigable video transmission using SRD. They summarize the design choices that allow players to render SRD-enabled DASH content, including (i) selecting the best resolution layers for the current viewport and (ii) enabling seamless switches among tiled videos. In addition, they give examples of how a player may use SRD to support zoomable and navigable videos by extending dash.js [137], which is an MPEG DASH reference client. Feuvre and Concolato [51] employ several open source projects, such as Kvazaar [187], MP4Box [142], and MP4Client [143], to realize tile-based adaptive transmission using MPEG-DASH and SRD. Furthermore, they discuss different adaptation policies of 360° tiled videos, where the tiles are either compressed independently or with tile-constrained motion vectors. Graf et al. [60] present a tile-based 360° streaming system and implement tools to evaluate the pros and cons of using different encoding and streaming strategies. They explore various options enabling the bandwidth-efficient 360° video adaptation over HTTP. They find that 6 × 4 tiles provide the best tradeoff between tiling overhead and bandwidth consumption. Nguyen et al. [126] propose streaming 360° tiled videos over HTTP/2 protocol (instead of HTTP/1.1), so as to capitalize its three unique features: (i) server pushes for achieving lower overhead and shorter response time, (ii) stream priority for sending more important tiles faster, and (iii) stream termination for stopping overdue tile transfers. Petrangeli et al. [150] also adopt HTTP/2 protocol for multiple representation transmission to reduce the network overhead. They also propose an algorithm to predict the tiles that may be watched in the future and use HTTP/2 to reduce the latency. The same authors extend the work into a complete system [149]. For example, dead reckoning is adopted to predict future viewports, and thorough evaluation results are reported. Yahia et al. [207] perform viewport prediction twice before the playout time of each segment to deal with the inevitable viewport prediction errors. In particular, the predicted tiles within the shorter playout time is delivered with higher priority. Their results show that lower bandwidth consumption and more stable video quality are achieved by prioritized streams. Yen et al. [208] adopt DASH over QUIC to further reduce the latency of the important tiles. In particular, they leverage fixation prediction and existing Adaptive Bitrate (ABR) algorithms to determine the (tile) stream priorities. In addition, a real 360° video streaming system over QUIC is built and evaluated in their study.

Conducting measurement study and carrying out reverse engineering on the commercial 360° video services, such as YouTube and Facebook, is another research direction. For example, Afzal et al. [5] analyze the characteristics of online 360° videos. They collect thousands of 360° videos from YouTube and classify them into several genres. They further analyze the variability in videos resolution and bitrate, and the possible underlying causes of these variabilities compared to transmitting non-360° videos. Zhou et al. [215] perform detailed measurements on Oculus 360° videos from Facebook and describe the offset cubic projection implemented by Oculus. In addition, they calculate the angles of users' viewports by their mathematical formula to give high-quality videos in the current viewports. Several experiments with different conditions are done to test how many segments are unwatched and wasted. Da Costa Filho et al. [36] propose developing a perceived quality model consisting of video playout performance, such as startup delay, visual quality, quality switches count, and video stalls. They adopt regression tree to model these metrics using network metrics, including delay, packet loss, and TCP throughput. In addition, tiling schemes are also considered as the model input. They first generate 360° tiled videos, then conduct experiments to collect the dataset. Their proposed model is then trained and validated using their dataset. They make a few observations (e.g., the video playout performance mainly depends on the network delay). These studies [5, 36, 215] shed some light on how to optimize 360° video transmission, although they may not directly achieve that.

Most 360° video transmissions are optimized through sending tiles at different quality levels. More precisely, the tiles in the viewer's current viewport are sent at higher quality, and others are sent at lower quality, so as to cut the bandwidth usage without viewing quality degradation. For example, Zare et al. [212] propose encoding each 360° video into tiles in two representations: high- and low resolutions. They then transmit the viewport tiles at high resolution and other tiles at low resolution. They adopt motion-constrained HEVC tiles and propose three heuristic schemes for 360° video transmission to HMDs. Even though no intelligent adaptation is done with their tiling schemes, experiment results reveal that their solution leverages the common patterns of head movements and achieves better coding efficiency. Nguyen et al. [122] also stream 360° videos in two regions: the center region, which is rectangular and covers the most-probable viewports, and the residue region. They solve an optimization problem to decide the size and encoding bitrate of the center region, so as to maximize video quality under the bandwidth constraint. Ju et al. [156] encode each 360° video into two representations and transmit the low-resolution full 360° videos along with high-resolution viewports. They propose considering the heat map of a viewer's attention for live transmitting the high-viewing probability portion. Corbillon et al. [34] encode 360° videos into two representations and take diverse projection models into consideration. In addition to varying the bitrates in different representations, they also consider the viewports of 360° videos in HMDs. In other words, each 360° video is divided into segments, where each segment is compressed multiple times with all combinations of viewports and bitrates. Each user then requests the proper representation via the DASH protocol. The same authors extend their work to include some theoretical models for viewport-adaptive 360° video streaming [33]. These models are then simplified by the following assumptions: (i) uniform coding complexity, (ii) two representations, (iii) maximum quality gap between the representations, and (iv) rectangular viewports. They then propose a viewport-adaptive streaming algorithm to exercise the tradeoff between the viewport size and the tile bitrates.

Duanmu et al. [43] generalize the preceding two-representation approach and encode each 360° video into a base and multiple enhancement representations. They then create different buffers for these representations and present their prioritized buffer control mechanisms. They give the highest priority to the playback continuity by first guaranteeing the transmission of the base representation. The residual bandwidth is then used to download enhancement representations. He et al. [65] propose adopting SHVC to compress the 360° videos. The core idea is to encode the whole 360° video as the base layer at basic quality and the viewports as the enhancement layer at a higher quality. Such approach mitigates the negative impact of wasting bandwidth on high-quality tiles that are outside of the viewports. Nasrabadi et al. [119] also stream SHVC tiles of 360° videos. Their core idea is to prefetch and buffer the base-layer tiles of the whole 360° videos earlier to avoid playout interruption and long rebuffering time. For tiles in viewports, enhancement-layer tiles are transmitted with residue bandwidth. Adaptations based on viewer orientations and network conditions are both considered in their work. Ozcinar et al. [139] propose sending the viewport at the highest-possible bitrate and gradually reducing the bitrates that are proportional to the distance to the viewport. The experiments show that their solution provides better viewing quality than the baselines, in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) of viewports. Hosseini and Swaminathan [69] also consider multiple representations, but they propose a new projection model for better viewing quality. Moreover, they leverage peripheral vision and further reduce the quality of some tiles in viewers' peripheral vision region. With viewport tracking, the dynamically delivered tiles within a user's viewports are at a higher bitrate. A bandwidth saving of 72% is reported without clear quality drops.

Nguyen et al. [123, 125] study the problem of optimally selecting the versions (bitrates) of individual tiles in 360° video streaming systems. In Nguyen et al. [125], they propose to add bitrates

and quality levels of tiles in the meta-data of DASH streaming. Next, three estimators are proposed at the client side to statistically predict the viewports, bitrates, and quality levels. These predictors are then used as the inputs to the version selection algorithms for maximizing the video quality. Nguyen et al. [123] solve the same version selection problem while taking the fixation prediction errors and user head movement into consideration. Their proposed algorithms extend the predicted viewports to absorb the fixation prediction errors while assigning high bitrate to the tiles in the viewports. Xiao et al. [197] formulate the bitrate selection of tiles for 360° videos as an ILP problem, where the goal is to minimize the sum of the unused bandwidth and the bitrate of invisible tiles. They rearrange tiles into macro-streaming units according to the characteristics of human visual systems. The features of HTTP/2, such as stream terminations and stream priority, are adopted for adjusting tile qualities and reducing request overhead for a better user experience. Chakareski et al. [23] study the RDO problem of 360° tiled videos in three steps. First, they propose building dynamic saliency maps to capture the diverse probabilities of different tiles to be watched. Second, they build the rate and distortion models on various QP values. Last, they formulate and solve a convex programming problem for optimal QP values of individual tiles, which can be used to optimally encode 360° videos.

Xie et al. [200] conduct a user study and find that viewers tend to be attracted by similar RoIs. Thus, the streaming server clusters the viewers by their viewing behaviors. The viewing probabilities of tiles are then predicted according to the viewer clusters. They then solve the optimal bitrate allocation problem based on the viewing probabilities with the constraints of the available bandwidth. Almquist et al. [10] conduct a very detailed analysis on the viewer behaviors with different videos. In particular, they classify 360° videosinto four categories based on the camera movements and video content. In addition, the impacts of the viewing duration on viewer behaviors are investigated. They design a utility function to address the tradeoff between the prefetching window and the quality selection for different potential viewing directions. Some other studies design tiling scheme (unequal-size tiles) to deliver 360° videos. Xiao et al. [196] propose an optimal tiling scheme for 360° videos to minimize the network and storage cost on server. They split 360° videos into tiles and model the storage cost as an ILP problem based on the features, such as the tile size, number of relocated motion vectors, storage overhead, and the number of tiles. The network cost is derived by the viewing probabilities and the number of downloads of each segment. Compared to the optimal fixed-size tiling scheme, their proposed method slightly reduces the storage and network overhead. Ozcinar et al. [140] study the similar problem by leveraging the fact that different areas of each 360° video attract diverse amounts of attention. They first develop a new objective quality metric weighted by user attention. Based on this metric, they develop a scheme to generate tiles in different sizes to retain the benefits of both smaller tiles (more flexible streaming decisions) and larger tiles (higher coding efficiency). Last, a bitrate allocation approach is presented for the varying-size tiles for optimal streaming quality. Similarly, Nguyen et al. [124] also solve the problem of adaptively determining the tile sizes to exercise the tradeoff between (i) flexibility on tile streaming and (ii) coding efficiency. These studies strive to save bandwidth by reducing the quality of unwatched or less-noticed tiles, which is agnostic to the network and computation infrastructures.

Optimizing 360° video transmission over specific networks and computational infrastructures has also been studied. For example, Qian et al. [154] specifically take the properties of cellular networks into account when optimizing 360° video transmission. They measure the performance of two 360° video content providers and find inferior performance in cellular networks. They then propose to predict the future viewports using weighted linear regression to mitigate such issue. The potential performance gain is quantified using numerical analysis based on the traces collected from five subjects. Sun et al. [178] study the optimal rate allocation for 360° tiled video streaming

in 5G networks. They propose adopting both tile-based rate allocation and fixation prediction to improve the streaming quality (qualified by the bitrate) without increasing the consumed bandwidth. The experiment results show that they indeed cope with fixation prediction errors and achieve higher video quality. Ahmadi et al. [7] study the problem of optimally multicasting tiles of 360° videos in multicast-enabled networks, such as Evolved Multimedia Broadcast Multicast Services (E-MBMS). They consider a DASH-based tile streaming system in multicast-enabled networks, where receivers are grouped based on channel conditions, and receivers in a group receiving the same copy of multicast stream. They propose a statistical approach to assign tile weights based on analysis on 1,300 head movement traces from HMDs and a heuristic rate adaptation algorithm to determine the bitrates of individual tiles for each receiver group. Their rate adaptation algorithm supports custom utility functions and allocates higher bitrates to tiles in the viewports. Bao et al. [15] consider the problem of hybrid multicast and unicast of tiles of 360° videos. Their motivation is on the observation that HMD viewers tend to have correlated viewport orientations, and thus multicasting overlapped regions of 360° videos should reduce the total bandwidth consumption. To achieve that, their system contains three main components: (i) viewport predictor, (ii) required regions calculator, and (iii) multicast and unicast selector. The viewport predictor takes the historical viewing direction as the input of neural networks to predict the future viewing directions, where yaw, roll, and pitch are predicted. The more detailed descriptions of their prediction networks can be found in their earlier work [14]. The required regions are computed based on a maximum missed pixel ratio specified by service providers. The multicast and unicast selector assumes that the channel conditions to every receiver are readily available. It compares the required network resources of transmitting each tile in multicast and unicast, and picks the lower one.

Hayes et al. [62–64] address the bitrate adaption problem of 360° video streaming over Multipath TCP (MPTCP), QUIC, and Software-Defined Network (SDN) in a series of work. They first consider the basic setup of a single 360° video client in Hayes et al. [62] and propose to leverage SDN to monitor the network conditions for better decision making in a 360° video streaming session. Two heuristics are proposed in their work to reduce (i) the startup time by intelligently select the best startup strategy (e.g., MPTCP vs. QUIC) and (ii) the out-of-order buffer by dynamically switching the network paths. Hayes et al. [63] extend the preceding work to support multiple 360° video clients in a network managed by an SDN controller. They propose two additional heuristics. First, the candidate paths are clustered into groups so that individual 360° video clients are assigned with paths having similar delays to avoid the out-of-order buffer problem. Second, active paths are dynamically assigned priority levels based on the client and network states. Moreover, the complementary nature of MPTCP and QUIC is exercised. For example, when a client suffers from an extremely low playout buffer, it switches to QUIC for a shorter response time. The evaluations with 500+ clients are done in the NS-3 simulator, which show (i) up to 40% reduction on multipath delay difference and (ii) up to 30% increase on network efficiency. Hayes et al. [64] explore the possibility of designing a Reinforcement Learning (RL)-based ABR algorithm for 360° video streaming over MPTCP and QUIC protocols in an SDN network. Although offline training is resource hungry and time consuming, with the trained algorithm, they report improvements on user experience, such as shorter stall time. Although Hayes et al. [62–64] adopt 360° video streaming as their driving application, they focus more on the networking aspects of the system. In particular, their proposed solutions treat 360° videos as ultra-high-resolution videos without, for example, considering viewer fixation and tiled encoding. Huang et al. [74] tackle the resource allocation problem among multiple multihomed 360° video clients. By *multihomed*, we refer to the clients come with several access networks. In their work, a cellular (LTE) and multiple WiFi links are considered. They adopt the popular logarithm function of bitrate to approximate the video

quality of the HMD viewports. The formulated resource allocation problem is NP-hard, which is solved by a heuristic algorithm. In terms of the practical concerns, the authors also propose new buffer management strategies for 360° video streaming.

Mangiante et al. [110] propose capitalizing the rendering capability of the edge cloud to optimize 360° video transmission over cellular networks. They propose an edge computing model that will be able to perform viewport rendering of 360° videos to end users. With Network Function Virtualization (NFV) and Mobile Edge Computing (MEC), their proposed solution optimizes the bandwidth usage and reduces the computation workload of receivers. Moreover, the lower network latency is guaranteed for real-time viewport rendering. Kamarainen et al. [84] develop a VR streaming system with a thin client by offloading most rendering processes to the cloud. Several optimization approach are proposed: (i) custom frustum culling, which adaptively renders the objects close to the viewer's viewport, (ii) multi-resolution rendering, which renders the front region of the viewport at high resolutions, and (iii) dynamic object replacement, which locally renders the objects interacted with the viewer while the control messages are also sent to the cloud. Their evaluation results, including a small-scale user study, show improvement in bandwidth consumption, interaction latency, and client computing loads. Lo et al. [107] also leverage an edge server to assist in rendering 360° tiled videos. In particular, they perform two rendering alternatives on edges: (i) combining tiles into a new video stream (e.g., high-quality tiles only in the viewport) and (ii) transcoding the viewport into a regular video stream. They also develop an algorithm to selectively serve each client with the most suitable alternative for optimal overall quality. Chakareski [22] suggests creating caches in cellular networks and studies the tradeoffs among caching, rendering, and transmission to improve the viewing quality. He also proposes a solution to best utilize the edge server resources at base stations, where different resources types (e.g., storage, networking, and computations) are provisioned. With assistance from edge servers, his proposed solution maximizes the aggregate award when serving the viewers. Mahzari et al. [109] propose a viewport-aware caching strategy to improve the hit ratio and bandwidth saving of different 360° videos. They consider the watched viewports of each 360° video to determine the caching strategy. Their experiment shows that the proposed approach improves the cache hit ratio by at least 40% and 17% compared to existing caching strategies such as LRU and LFU. Prins et al. [153] consider an AudioVisual (A/V) delivery network that is built on multiple A/V relayers and A/V proxies. Two transmission modes—tiled HAS and pub-sub—are supported, where tiled HAS targets individual viewers and pub-sub enables multicast transmission to many subscribers. Zhang et al. [213] propose leveraging multicast in Information Centric Network (ICN) to support multi-party video conferencing. ICN adopts pull-based streaming to fetch the desired content. It is, however, challenging to build a conferencing system with pull-based streaming. To ensure real-time requirements, they adopt a centralized server, specialized edge routers, and a web VR player. The measured latency is small and consistent—less than 150 and 350ms for audio and video, respectively. Most of the infrastructure-specific optimizations of 360° video transmission still have room for further optimization. For example, offloading VR rendering to the edge cloud still faces many challenges, such as low latency and mobility supports.

4.4 Display

Online services for sharing 360° videos, such as YouTube and Facebook, attract more and more users because more users capture 360° videos using handheld or wearable 360° cameras. However, such casual 360° videos often suffer from shaky content, which make viewers feel dizzy and suffer from a degraded user experience. Thus, several studies aim to stabilize the videos before displaying them to eliminate the discomfort of viewers. Kopf [92] develops a hybrid 3D-2D algorithm to stabilize the 360° videos. It is done by the following steps: (i) tracking the motion of feature points,

(ii) estimating the relative rotation on key frames in the 3D space, (iii) performing de-rotation in key frames and interpolations in other frames, and (iv) stabilizing the trajectories of the tracked feature points in the 2D space. The proposed stabilization algorithm runs in real time and reduces the video bitrate by at least 10%. Kasahara et al. [85] record First-person Omni-Directional Videos (FODVs) using six wide-angle cameras mounted on the subject's head. They propose algorithms to first calculate and eliminate the rotation in FODV, then reconstruct the smooth rotation to follow the viewer's orientation. They further conduct a user study to evaluate their algorithm in different scenarios. It is observed that their solution alleviates cybersickness in all scenarios.

The 360° videos can be displayed on regular, not head-mounted, displays. The majority of such studies concentrate on displaying 360° using Web browsers, which are more accessible to many users. For example, Quax et al. [155] propose encoding and encapsulating 360° videos using the WebM format in real time at the back-end; WebM is a container format supported by Google Chrome. The front-end program, implemented in JAVA, dynamically inserts a *canvas* element to enable user interactions and viewport displays. The visualization can be realized using 2D planar projection or 3D spherical texture projection according to various factors, such as the computing power of the device and the characteristics of the video content. Neng and Chambel [120] present a 360° hypervideo player that provides the interface for visualizing and navigating the 360° videos. Their proposed interface allows users to view scenes in different angles by dragging the scenes left or right. In addition, some widgets are developed for better user interface. For example, a pie chart indicating the current viewing angle is displayed to remind users of their orientations. In addition, a mini-map in equirectangular projection is placed at the bottom to let the users know the content outside their viewports. Some hotspots are further displayed on the mini-map, and users click on the hotspots to change their viewports.

The 360° videos can also be displayed in HMDs with eye-tracking capability. Such HMDs have been proposed in the literature [141] and commercialized [53]. Stengel et al. [117] propose an inexpensive but complete eye-tracking HMD without viewport reduction caused by eye-tracking cameras. Their eye-tracking HMDs are equipped with the following components compared to regular HMDs: (i) eye-tracking cameras, which track the user's gazes; (ii) an infrared illumination unit, which highlights the contour of the pupil; and (iii) dichroic mirrors, which allow the eye-tracking cameras to track the gazes without blocking the viewport of users. Both HMD and user calibrations are required for precise eye- and even pupil-tracking estimations. They recruited 33 subjects for a user study and demonstrated positive results in terms of user experience and usefulness. The stability and accuracy still have room for improvement, because cables on cameras and HMDs constrain users' head movements. Such eye-tracking HMD is essential for various applications, such as gaze control of virtual objects and foveated rendering. Pohl et al. [152] propose a method to perform foveated rendering in eye-tracking HMDs. In particular, they compute the sampling map of foveated rendering according to the lens astigmatism of HMD and the viewing angle of the tracked gazes. Their proposed method speeds up the rendering operations. Kim et al. [87] and Patney et al. [145] also demonstrate the feasibility of leveraging an HMD with an eye tracker for foveated rendering of 360° videos. In a nutshell, the viewer fixation is tracked by the HMD while the tiles closer to the fixation center are (i) streamed at higher quality (in the case of recorded 360° videos [87]) or (ii) 3D rendered with more details (in the case of computer-generated content [145]). Patney et al. [146] aim to boost the gain of foveated rendering in HMDs. They start from a user study on VR perceptual quality and identify two key factors affecting foveated rendering: (i) temporal stability and (ii) contrast preservation. Several tools for mitigating temporo-spatial aliasing and normalizing image contrast are proposed and implemented in their renderer. Romero-Rondon et al. [161] build a foveated streaming system for 360° videos. The server prepares videos with foveal area at the high resolution and other areas at the low resolution. This is to save the

network bandwidth and the client rendering resource consumption. Similar to foveated rendering and foveated streaming, Lee et al. [100] apply the same concept to stitching multiple camera feeds into a 360° video in real time. More specifically, they track the viewer fixation and generate regions closer to the viewer fixation point at higher resolutions, so as to achieve real-time stitching. Their work also takes the delay from different components (i.e., HMDs, servers, and networks) into consideration when adapting the parameters of their foveated stitching approach. Lai and Hsu [95] develop a 360° viewing system that adopts light-field technologies to realize image refocusing following the user's fixation. In particular, a light-field 360° image is stitched from multiple light-field images captured by rotating the light-field camera. They propose two mechanisms to speed up the refocusing process: (i) pre-rendering approximated viewports using representative depth values and (ii) rendering the viewer's viewports only. By doing so, the refocusing process runs in real time when viewers watch the 360° images.

Table 9 in Appendix C gives the comparisons among the display-related optimization works in the literature. Increasingly more studies focus on HMDs, which offer a more immersive experience to the users. Some of these HMDs are equipped with eye trackers, which enable new applications, such as foveated rendering, gaze transfer, and refocusing. Among them, foveated rendering reduces the required computing resources while still offering immersive experience. Gaze transfer for avatars improves the interaction experience in social VR. Refocusing based on the viewer's gaze and depth map makes the immersive environment closer to the real world and has great potential in future 3-DoF+ and 6-DoF applications.

4.5 Quality Assessments

Video QoE refers to the human-perceived video quality, which can only be quantified using rigorously designed testbeds and procedures. The QoE metrics are either (i) *subjective* or (ii) *objective metrics*. The subjective metrics are from user inputs, mostly through questionnaires, whereas the objective metrics are from computer algorithms. The subjective metrics are better inline with actual human perception but require more efforts to design, conduct, and analyze. In contrast, the objective metrics are easier to derive but may deviate from the human perception. QoE evaluations and optimization have been crucial for multimedia applications, even before 360° videos were popularized. For example, Tan et al. [179] present the standardized evaluation procedures for quantifying the subjective and objective QoE levels achieved by the latest H.265/HEVC video coding standard. The authors adopt PSNR as the objective quality metric. In terms of subjective quality assessment, they follow the procedure suggested by ITU-Rec. P910 [168] and ITU-Rec. BT500 [167], which are for multimedia applications and television pictures, respectively. In particular, the subjects are asked to view and rate a random series of basic test cells, where each cell contains two video clips: the original video clip followed by the reconstructed one. Their evaluation procedure is not suitable for 360° videos, which are often transmitted in tiles for selectively transmitting tiles that are more likely to be watched to save bandwidth consumption. Wang et al. [192] consider the QoE of zoomable video tiles, which may have diverse resolutions. Their evaluations reveal that users may not notice some tiles that are transmitted at lower resolutions. In particular, they conduct user studies to measure two thresholds for just noticeable/unacceptable differences. These two thresholds are then used to drive the resolution selection among tiles under the restricted bandwidth. Their experiment results demonstrate a 14%–20% bandwidth saving using their proposed method. In addition, the results also reveal that the two thresholds are related to the characteristics of video content, such as motion levels. However, their work focuses on conventional flat displays rather than modern HMDs.

QoE evaluation testbeds and procedures have gradually received more attention. A testbed is built [184] for evaluations of QoE metrics of 360° videos. They demonstrate the applicability of

their testbed by using it to collect the Mean Opinion Score (MOS) of 360° images and videos encoded at different quality levels. Their testbed allows subjects to view images and videos using HMDs based on mobile devices, such as Google Cardboard. During each assessment session, their proposed testbed tracks the subject's scores, orientation, and consumed time. Regal et al. [159] move a step further by integrating QoE questionnaires with the VR world. The testbed is implemented using Unity, and the virtual questionnaire is displayed as a 2D canvas. The collected scores are stored in a CSV file during the session for analysis. Singla et al. [171] assess sickness caused by watching 360° videos in HMDs via subjective evaluations. They consider two commercial HMDs and two resolutions in their experiments. In other words, each video content is viewed four times with different combinations of HMDs and resolutions. Twenty-eight subjects are recruited to rate the 360° videos downloaded from YouTube, where six videos in total are used. Their results show that both the resolution and content have significant impact on subjects' experience, whereas the HMDs impose only a slight influence on it. Example observations include that HTC Vive provides a slight better overall quality compared to Oculus Rift. In addition, average female users suffer from more sickness, especially for disorientation. Singla et al. [170] quantify the implication of different bitrates and resolutions through subjective evaluations. The authors propose a modified subjective evaluation procedure, which (i) allows the subjects to view the test sequence twice for more reliable rating and (ii) asks the subjects to rate the test sequence through speech to prevent the interruption of wearing/taking off the HMD. Bessa et al. [18] study whether 3D (stereoscopic) views will improve the subjective QoE levels compared to 2D views. They recruit 63 participants to view a single video. Half of the participants watch the video in the 2D version, whereas the other half watch the video in the 3D version. Surprisingly, their results show that 3D 360° videos bring no benefit to the viewers compared to 2D 360° videos. This may be because the subjects have limited 360° video viewing experience, especially in 3D viewers. QoE evaluations on specific applications are also possible. For example, Hupont et al. [75] propose procedures to study the gaming QoE with HMDs. The authors conduct experiments to evaluate (i) the perceived presence scores and (ii) the usability scores on both conventional 2D displays and HMDs. Their results show that wearing HMDs provides better experience compared to 2D displays in various aspects, such as realism, possibility to act, and willingness to use, and it leads to higher complexities and steeper learning curves. Schatz et al. [165] consider VR-based training applications, such as assigning car parts to their corresponding positions. They study how the rendering styles, such as point cloud rendering or the physics-based Unity shader, and scene types may influence the subjective scores and task performance. Singla et al. [172] conduct a user study on 360° tiled videos. They build a streaming testbed and vary the factors, including bandwidth, delay, and resolution, In particular, they measure the perceived quality and sickness under different delay types. Their results show that 47ms is the sustainable network delay that does not influence the quality ratings. In addition, tile-based streaming is shown to be more reliable under bandwidth-limited scenarios. Although these studies [18, 75, 159, 165, 170–172, 184] shed some light on fine tuning the QoE of 360° videos, they do not actively optimize the QoE of 360° video systems.

Some work moves further and optimizes 360° video systems in terms of user QoE levels. Hsu et al. [71] carry out QoE evaluations on foveated rendering systems, where the objects in the foveal region are encoded at higher quality than the objects in the peripheral region. In their study, they vary the resolution of the foveal region and peripheral region on 2D displays and consider four types of subjective quality assessment methods. They find that most of the viewers do not perceive the distortion if the size of the foveal region is larger than 7.5°. Moreover, they evaluate the consistency and efficiency of the considered assessment methods. Based on the results from the considered four methods, they model the perceptual ratio on foveated rendering using regression analysis. Albert et al. [8] carry out detailed user studies to understand how the system latency

affects the VR user experience with desktops and HMDs. Several key factors are investigated, including: (i) the foveal area size, (ii) the severity of the degradation, and (iii) the degradation algorithm. Extensive experiments are conducted to understand whether human subjects can easily notice artifacts under diverse setups. The main takeaway message is that the acceptable eye-to-image latency is between 50 and 70ms. Vlahovic et al. [188] study the impact of locomotion, including first-person, teleportation, tunneling, and gesture based, in VR applications. Among these locomotions, the controller-based locomotion results in higher discomfort levels. The teleportation one, in contrast, receives the lowest discomfort level and the highest ranking in terms of QoE. Steed et al. [176] study how the user interface and conditions may affect the VR QoE of HMD users. In particular, the considered scenario is a virtual singer singing on the stage. The authors consider eight conditions in this scenario using Unity in their user studies. These conditions come from the combinations of three settings: (i) with or without a self-avatar, (ii) with or without the singer asking the user to tap along to the beats, and (iii) with or without the singer looking at the user. Their results reveal that the self-avatar makes a clear impact on the user experience. Moreover, the user experience is degraded due to synchronization issues between the subject and his or her self-avatar when tapping along with the beats. However, with the singer looking at the user, no negative effect on viewing experience is observed. The authors argue that this is because the user has no expectation of the singer to engage with him or her. Fernandes and Feiner [49] propose an evaluation design to understand the relations between viewport size, sickness, and perceived quality. In particular, they vary the viewport degree between 80° and 90°. Each subject is asked to wear an HMD and walk along a set of waypoints in virtual environments. They rate the discomfort levels every five waypoints. Their experiment results indicate that restricting viewport size helps subjects adapt to the virtual environments and reduce the discomfort level as long as the restricted viewport size is acceptable to the subjects. These works [8, 49, 71, 176, 188] concentrate on subjective evaluations.

Recruiting viewers for subjective evaluations is costly, error prone, and tedious. Therefore, several works [44, 183, 185] discuss how to estimate the subjective results using the objective results, so as to reduce the overhead of subjective evaluations. Upenik et al. [185] conduct subjective evaluations with 45 subjects, trying to analyze the correlation between the subjective MOS values and objective quality levels. Their considered objective quality metrics include PSNR, SSIM, M-SSIM, VIFP, S-PSNR, WS-PSNR [118], and CPP-PSNR [210]. Their analysis on the subjective scores and objective metrics show that the existing objective quality metrics designed for 360° videos (e.g., S-PSNR and WS-PSNR) do not have higher correlation to the subjective scores than the original metrics (e.g., PSNR). Therefore, the authors conclude that there are still open issues in this research direction. For example, a better objective quality metric specifically designed for 360° videos is needed. Tran et al. [183] conduct similar evaluations on 18 subjects. Their findings are more positive compared to Upenik et al. [185]. For example, all of their considered objective metrics have high correlation to the subjective results. The sources of distortion are due to (i) changing video content format and (ii) transmitting content over networks. Last, Egan et al. [44] predict the QoE scores based on the biosensors. It is shown that the electrodermal activity has significant contributions to the QoE scores. On the contrary, the heart rate has no effect on the subjective scores. Moreover, their results confirm that exploring VR using HMDs leads to more immersive experience than using 2D displays.

Some recent studies [89, 201, 208] focus on developing QoE models to better reflect the user experience of watching 360° videos. Kim et al. [89] analyze both the velocity and direction of a subject's head motion and those of visual objects. In addition, the content features, such as the background complexity and object motion, are analyzed to understand the sensitivity of subjects under different conditions. They recruit 80 subjects, and each subject rates the video with a

five-scale sickness level. The collected scores are fed into a Support Vector Regression (SVR) model for sickness level prediction. Yao et al. [218] recruit 60 subjects to watch four different genres of 360° videos using their developed player. The considered factors include projection scheme, encoding QP, video characteristics, and objective quality metrics. Based on their collected subjective scores on different factors, they build a QoE model using linear regression. In terms of 360° tiled videos, Xie et al. [201] study the negative impacts of diverse quality levels between the region inside and outside of the viewports. A user study is performed to collect MOS scores, which are then used to construct a perceptual model on quantization, resolution settings, and fade-in/out period. The closed-form model is a product of two exponential functions.

Table 10 in Appendix C summarizes the user studies conducted in the literature. Most of the works study factors impacting the subjective quality and the sickness, where the considered factors include the video content, the adopted devices, and the interplay among subject behaviors and VR environments. With large search space due to the dynamic viewports, there is still room for QoE assessment on 360° videos.

5 APPLICATIONS

Several novel applications are built on or related to 360° video streaming systems. We classify these representative works into four categories: (i) stereoscopic 360° video streaming, (ii) 360° videos in immersive environments, (iii) object detection and tracking in 360° videos, and (iv) salience and fixation detection in 360° videos. These works can be used as foundations for developing future innovative applications and services.

Stereoscopic 360° video streaming has been investigated in the literature. For example, Thatte et al. [180] propose a new data file composed of left/right texture and depth images to realize stereo 360° videos. This data file can be rendered by the following steps. First, all of the scene pixels are mapped to a point cloud. Next, the stereoscopic viewports are synthesized by projection based on the eye position and viewing direction. Im et al. [76] also study stereoscopic 360° video reconstruction. In particular, they leverage the bundle adjustment of the unit sphere for a more accurate camera pose estimation. A sweeping algorithm is then developed for estimating the depth maps by exploiting the color consistency of overlapping regions observed on the virtual sphere centered at the camera. Their proposed techniques produce anaglyph 360° videos. Streaming stereoscopic 360° videos inherently leads to much more network traffic, which is even more challenging than ordinary 360° video streaming. Thus, the compression algorithms for stereoscopic 360° videos should be developed carefully. Moreover, more comprehensive ABR algorithms to wisely distribute the network bandwidth among different components of stereoscopic 360° videos (e.g., texture vs. depth) must be proposed.

Immersive environments may also be enhanced by 360° videos, such as through *hypervideos*. Hypervideos are videos that contain links to other videos. For instance, Neng and Chambel [120] present an interactive 360° hypervideo player with navigation supports. Noronha et al. [130] present an interactive player for visualizing and navigating georeferenced 360° hypervideos, which are captured with GPS and digital compass readings. These videos can also be indexed with geographic information reported by users. Viewers are allowed to search, watch, pan, and link to hotspots in 360° georeferenced videos. These videos can be used for city touring or kart racing with real maps. Ohta et al. [136] develop an AR Web shopping system, which has several interaction modes, such as picking up products with dragging, comparing products with AR interface, and navigating the store with photorealistic 360° videos. Berning et al. [17] present an AR prototyping system, which includes capturing, editing, and playback of AR videos. Becoming a common part of our living environment, these applications demand a tremendous amount of resources and may dictate aggregated supports from fog devices, edge gateways, and cloud servers. Hou et al. [70]

present several similar application scenarios, which are also resource hungry. They further propose techniques for capitalizing edge and cloud resources to meet the VR/AR requirements, which are crucial in immersive environments.

Object detection and tracking are challenging in 360° videos because of the shape distortion and discontinuity caused by projection. There are studies in the literature tackling these challenges. For instance, Markovic et al. [111] propose considering sensor geometry and processing data on unit sphere space to mitigate the negative impacts of projection. They compute the optical flow and extract the feature points from the unit sphere. The causes of the flow vectors are then classified into ego-motion or moving objects. For the vectors caused by moving objects, the gravity of the vector center is then calculated and tracked by a robot based on Bayesian estimation. Delforouzi and Grzegorzek [40] study the object tracking problem in 360° videos under more complex situations, such as occlusion and out-of-plane rotation. They leverage learning-based methods to enhance feature matching and classify the challenging situations. Once a challenging situation is detected, the corresponding handler will be triggered for better tracking quality. This application can be integrated with the Internet-of-Things (IoT) for various smart city applications. For example, stray dogs can be detected and tracked through 360° surveillance cameras mounted across the city during rabies outbreaks. Reducing the footprint of these applications to fit IoT devices unleashes many usage scenarios and improves quality of urban life.

Saliency and fixation detection (prediction) in 360° videos are challenging yet open up a lot of research opportunities. Traditional saliency and fixation detection algorithms are designed for 2D planar videos and may not work well with 360° videos. However, additional sensor readings from HMDs may be leveraged by new saliency and fixation prediction algorithms. We highlight the relevant literature in Table 11 in Appendix C. De Abreu et al. [3] build end-to-end testbeds for viewing 360° images and recruit dozens of participants to collect their viewport trajectories. They observe the equator bias of 360° videos and propose to fuse saliency maps predicted from multiple projected 360° videos. Their experiment results demonstrate a 20% improvement compared to the current saliency detection models. Monroy et al. [115] first map 360° images to six faces of the cubic projection, which reduces the distortion close to poles compared to the equirectangular projection. Each face is detected by a conventional saliency detection network with spherical coordinates for locating the face on the sphere. Finally, six detected saliency faces are combined into a single saliency map for the 360° image. Zhang et al. [214] develop a spherical Convolutional Neural Network (CNN) with spherical Mean Squared Error (MSE) loss function, which takes the angle to the center of the sphere into consideration. In addition, the starting position of the viewer watching each 360° video is considered as a feature for their model. Cheng et al. [26] map the 360° videos into the cubic projection for saliency detection. Wider angles and temporal models are developed to improve the accuracy. Ban et al. [13] propose to predict the viewer's head movements in three steps. First, an initial prediction is performed based on the viewer's previous viewing position using linear regression. Second, K-Nearest Neighbors (KNNs) are employed to find the nearest K viewpoints among all other viewers. Last, the viewport regions of the K-nearest viewpoints are calculated to predict viewing probability of each tile. Using features from both contents and sensors, Fan et al. [47] develop neural networks to estimate the viewer's viewports in the future. They demonstrate that (i) HMD orientations from HMD sensors and (ii) saliency and motion intensity from 360° videos could be used as the inputs of such neural networks. They present two neural networks and report their performance results from trace-driven simulations. Nguyen et al. [121] employ a similar fixation prediction network that considers both the orientation and saliency. They develop their own image saliency network trained on their 360° video viewing dataset. Their considered features are the detected saliency maps and the orientation maps of previous video frames. Xu et al. [204] develop the head movement prediction network based on RL considering the

previous viewer orientations and video content. The network aims to predict the next head-moving direction among the eight directions (top, left, bottom, right, and the direction between any two of them). Xu et al. [205] address the fixation prediction problem with eye-tracking–capable HMDs using CNN and LSTM. Their proposed solution considers three main factors: temporal and spatial saliency maps and historical fixation trajectories. The authors add an eye tracker to a regular HMD (withoutan eye-tracking feature) and collect a large dataset with 208 videos and 31 subjects, which is then employed in their evaluations.

Similar to saliency objects, 3D sounds can also attract a viewer's attentions in 360° videos. Vosmeer and Schouten [189] point out that viewers can freely choose their viewports, which make the film director hard to guide the viewers. The 3D sounds may help directors attract viewers' attention to particular directions according to their scripts. Chou et al. [28] consider a new research problem of automatically determining the viewports referred by the narrative in 360° videos so that some visual guidance can be added to the 360° videos for HMD viewers. Machine-learning tools, such as CNNs and Recurrent Neural Networks (RNNs), are adopted to identify the viewports with the maximum attention from given 360° videos and narrative subtitles. They collect a dataset of narrated 360° videos and train their model without labeling. Their proposed solution runs in real time and may be integrated with a 360° video streaming system to guide HMD viewers toward proper orientations.

There have been quite a few works focusing on saliency and fixation prediction, which can be attributed to the limited resources and high-quality requirements of 360° video streaming systems. Nevertheless, these algorithms still produce prediction error, which may cause *black holes* or *playout stalls* in the HMDs during playout. Some recent studies [207, 208] propose prioritizing the tiles and sending the missing tiles (due to imperfect prediction) to the client by leveraging emerging protocol features, such as stream prioritizing and multiplexing of HTTP/2 and QUIC.

6 CONCLUSIONS

Along with the advances of network bandwidth and speed, users demand a more immersive 360° video viewing experience. In this article, we survey recent 360° video streaming research, which to the best of our knowledge has not yet been considered. Our survey covers a wide spectrum of academic work in the literature: from systems to datasets to applications. We also dive deep into the 360° video streaming pipeline and discuss novel optimization techniques in individual stages. Furthermore, we survey the commodity hardware and software that can be used by engineers, researchers, and hobbyists to build a working 360° video streaming platform for diverse purposes. This survey will encourage more exciting work on 360° video streaming and eventually make immersive 360° viewing experience part of our daily life.

REFERENCES

[1] Ameer Abbasi and Uthman Baroudi. 2012. Immersive environment: An emerging future of telecommunications. *IEEE MultiMedia* 19, 1 (Jan. 2012), 80–80.

[2] M. Abdallah, C. Griwodz, K. Chen, G. Simon, P. Wang, and C. Hsu. 2018. Delay-sensitive video computing in the cloud: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 14, 3S (2018), 54.

[3] A. De Abreu, C. Ozcinar, and A. Smolic. 2017. Look around you: Saliency maps for omnidirectional images in VR applications. In *Proc. of QoMEX'17*. 1–6.

[4] H. Afshari, V. Popovic, T. Tasci, A. Schmid, and Y. Leblebici. 2012. A spherical multi-camera system with real-time omnidirectional video acquisition capability. *IEEE Transactions on Consumer Electronics* 58, 4 (Nov. 2012), 1110–1118.

[5] S. Afzal, J. Chen, and K. Ramakrishnan. 2017. Characterization of 360-degree videos. In *Proc. of ACM VR/AR Network'17*. 1–6.

[6] R. Aggarwal, A. Vohra, and A. Namboodiri. 2016. Panoramic stereo videos with a single camera. In *Proc. of IEEE CVPR'16*. 3755–3763.

[7] H. Ahmadi, O. Eltobgy, and M. Hefeeda. 2017. Adaptive multicast streaming of virtual reality content to mobile users. In *Proc. of ACM Thematic Workshops'17*. 170–178.
[8] R. Albert, A. Patney, D. Luebke, and J. Kim. 2017. Latency requirements for foveated rendering in virtual reality. *ACM Transactions on Applied Perception* 14, 4 (2017), 25.
[9] P. Alface, M. Aerts, D. Tytgat, S. Lievens, C. Stevens, N. Verzijp, and J. Macq. 2017. 16K cinematic VR streaming. In *Proc. of ACM MM'17*. 1105–1112.
[10] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, and D. Eager. 2018. The prefetch aggressiveness tradeoff in 360° video streaming. In *Proc. of ACM MMSys'18*. 258–269.
[11] R. Anderson, D. Gallup, J. Barron, J. Kontkanen, N. Snavely, C. Hernandez, S. Agarwal, and S. Seitz. 2016. Jump: Virtual reality video. *ACM Transactions on Graphics* 35, 6 (Nov. 2016), 198.
[12] J. Apostolopoulos, P. Chou, B. Culbertson, T. Kalker, M. Trott, and S. Wee. 2012. The road to immersive communication. *Proceedings of the IEEE* 100, 4 (Feb. 2012), 974–990.
[13] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang. 2018. Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming. In *Proc. of IEEE ICME'18*. 1–6.
[14] Y. Bao, H. Wu, T. Zhang, A. Ramli, and X. Liu. 2016. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In *Proc. of IEEE Big Data'16*. 1161–1170.
[15] Y. Bao, T. Zhang, A. Pande, H. Wu, and X. Liu. 2017. Motion-prediction-based multicast for 360-degree video transmissions. In *Proc. of IEEE SECON'17*. 1–9.
[16] A. Belbachir, S. Schraml, M. Mayerhofer, and M. Hofstatter. 2014. A novel HDR depth camera for real-time 3D 360° panoramic vision. In *Proc. of IEEE CVPR'14*. 425–432.
[17] M. Berning, T. Yonezawa, T. Riedel, J. Nakazawa, M. Beigl, and H. Tokuda. 2013. pARnorama: 360 degree interactive video for augmented reality prototyping. In *Proc. of ACM UbiComp'13*. 1471–1474.
[18] M. Bessa, M. Melo, D. Narciso, L. Barbosa, and J. Vasconcelos-Raposo. 2016. Does 3D 360 video enhance user's VR experience: An evaluation study. In *Proc. of Interaction'16*. Article 16, 4 pages.
[19] J. Boyce, Y. Ye, J. Chen, and A. Ramasubramonian. 2016. Overview of SHVC: Scalable extensions of the high efficiency video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 1 (Jan. 2016), 20–34.
[20] K. Calagari, M. Elgharib, S. Shirmohammadi, and M. Hefeeda. 2017. Sports VR content generation from regular camera feeds. In *Proc. of ACM MM'17*. 699–707.
[21] E. Canessa and L. Tenze. 2014. FishEyA: Live broadcasting around 360 degrees. In *Proc. of ACM VRST'14*. 227–228.
[22] J. Chakareski. 2017. VR/AR immersive communication. In *Proc. of ACM VR/AR Network'17*. 36–41.
[23] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan. 2018. Viewport-driven rate-distortion optimized 360º video streaming. In *Proc. of IEEE ICC'18*. 1–7.
[24] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui. 2017. Mobile augmented reality survey: From where we are to where we go. *IEEE Access* 5 (April 2017), 6917–6950.
[25] S. Chen. 2017. Transform360 Is an Equirectangular to Cubemap Transform for 360 Video. Retrieved April 26, 2018 from https://github.com/facebook/transform360.
[26] H. Cheng, C. Chao, J. Dong, H. Wen, T. Liu, and M. Sun. 2018. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proc. of IEEE CVPR'18*. 1420–1429.
[27] K. Choi and K. Jun. 2016. Real-time panorama video system using networked multiple cameras. *Journal of Systems Architecture* 64 (March 2016), 110–121.
[28] S. Chou, Y. Chen, K. Zeng, H. G. Hu, J. Fu, and M. Sun. 2018. Self-view Grounding given a narrated 360 video. In *Proc. of AAAI'18*.
[29] HTC Co. 2017. VIVE: Discover Virtual Reality Beyond Imagination. Retrieved April 26, 2018 from https://www.vive.com/us/.
[30] O. Cogal, A. Akin, K. Seyid, V. Popovic, A. Schmid, B. Ott, P. Wellig, and Y. Leblebici. 2014. A new omni-directional multi-camera system for high resolution surveillance. *Mobile Multimedia/Image Processing, Security, and Applications* 9120, 9 (May 2014), 91200N-1–91200N-9.
[31] C. Concolato, J. Feuvre, F. Denoual, E. Nassor, N. Ouedraogo, and J. Taquet. 2018. Adaptive streaming of HEVC tiled videos using MPEG-DASH. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 8 (Aug. 2018), 1981–1992.
[32] X. Corbillon. 2017. 360Transformations. Retrieved April 26, 2018 from https://github.com/xmar/360Transformations.
[33] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski. 2017. Optimal set of 360-degree videos for viewport-adaptive streaming. In *Proc. of ACM MM'17*. 943–951.
[34] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. 2017. Viewport-adaptive navigable 360-degree video delivery. In *Proc. of IEEE ICC'17*. 1–7.
[35] X. Corbillon, F. Simone, and G. Simon. 2017. 360-degree video head movement dataset. In *Proc. of ACM MMSys'17*. 199–204.

[36] R. da Costa Filho, M. Luizelli, M. Torres Vega, J. van der Hooft, S. Petrangeli, T. Wauters, F. De Turck, and L. Gaspary. 2018. Predicting the performance of virtual reality video streaming in mobile networks. In *Proc. of ACM MMSys'18*. 270–283.
[37] L. D'Acunto, J. Berg, E. Thomas, and O. Niamut. 2016. Using MPEG DASH SRD for zoomable and navigable video. In *Proc. of ACM MMSys'16*. Article 34, 4 pages.
[38] S. Dambra, G. Samela, L. Sassatelli, R. Pighetti, R. Aparicio-Pardo, and A. Pinna-Dery. 2018. Film editing: New levers to improve VR streaming. In *Proc. of ACM MMSys'18*. 27–39.
[39] E. David, J. Gutierrez, A. Coutrot, M. Da Silva, and P. Callet. 2018. A dataset of head and eye movements for 360° videos. In *Proc. of ACM MMSys'18*. 432–437.
[40] A. Delforouzi and M. Grzegorzek. 2017. Robust and fast object tracking for challenging 360-degree videos. In *Proc. of IEEE ISM'17*. 274–277.
[41] M. Domanski, O. Stankiewicz, K. Wegner, and T. Grajek. 2017. Immersive visual media MPEG-I: 360 video, virtual navigation and beyond. In *Proc. of IWSSIP'17*. 1–9.
[42] Doxygen. 2017. HEVC Test Model (HM). Retrieved April 26, 2018 from https://hevc.hhi.fraunhofer.de/HM-doc/.
[43] F. Duanmu, E. Kurdoglu, S. Hosseini, Y. Liu, and Y. Wang. 2017. Prioritized buffer control in two-tier 360 video streaming. In *Proc. of ACM VR/AR Network'17*. 13–18.
[44] D. Egan, S. Brennan, J. Barrett, Y. Qiao, C. Timmerer, and N. Murray. 2016. An evaluation of heart rate and electrodermal activity as an objective QoE evaluation method for immersive virtual reality environments. In *Proc. of QoMEX'16*. 1–6.
[45] T. El-Ganainy and M. Hefeeda. 2016. Streaming virtual reality content. arXiv: 1612.08350.
[46] Samsung Electronics. 2017. The GearVR Framework (GearVRf). Retrieved April 26, 2018 from https://github.com/Samsung/GearVRf.
[47] C. Fan, J. Lee, W. Lo, C. Huang, K. Chen, and C. Hsu. 2017. Fixation prediction for 360° video streaming in head-mounted virtual reality. In *Proc. of ACM NOSSDAV'17*. 67–72.
[48] W. Fenlon. 2013. The Challenge of Latency in Virtual Reality. Retrieved April 26, 2018 from http://www.tested.com/tech/concepts/452656-challenge-latency-virtual-reality/.
[49] A. Fernandes and S. Feiner. 2016. Combating VR sickness through subtle dynamic field-of-view modification. In *Proc. of IEEE 3DUI'16*. 201–210.
[50] A. Ferworn, B. Waismark, and M. Scanlan. 2015. CAT 360: Canine augmented technology 360-degree video system. In *Proc. of IEEE SSRR'15*. 1–4.
[51] J. Le Feuvre and C. Concolato. 2016. Tiled-based adaptive streaming using MPEG-DASH. In *Proc. of ACM MMSys'16*. Article 41, 3 pages.
[52] J. Foote and D. Kimber. 2000. FlyCam: Practical panoramic video and automatic camera control. In *Proc. of IEEE ICME'00*. 1419–1422.
[53] FOVE Inc.2019. FOVE: Eye-Tracking Virtual Reality Headset. Retrieved July 30, 2019 from https://www.getfove.com/.
[54] S. Fremerey, A. Singla, K. Meseberg, and A. Raake. 2018. AVtrack360: An open dataset and software recording people's head rotations watching 360° videos on an HMD. In *Proc. of ACM MMSys'18*. 403–408.
[55] Y. Sanchez, R. Skupin, and T. Schierl. 2015. Compressed domain video processing for tile based panoramic streaming using HEVC. In *Proc. of IEEE ICIP'15*. 2244–2248.
[56] The Medical Futurist. 2017. 5 Ways Medical Virtual Reality Is Already Changing Healthcare. Retrieved April 26, 2018 from http://medicalfuturist.com/5-ways-medical-vr-is-changing-healthcare/.
[57] V. Gaddam, H. Ngo, R. Langseth, C. Griwodz, D. Johansen, and P. Halvorsen. 2015. Tiling of panorama video for interactive virtual cameras: Overheads and potential bandwidth requirement reduction. In *Proc. of PCS'15*. 204–209.
[58] L. Gaemperle, K. Seyid, V. Popovic, and Y. Leblebici. 2014. An immersive telepresence system using a real-time omnidirectional camera and a virtual reality head-mounted display. In *Proc. of IEEE ISM'14*. 175–178.
[59] Google Inc. 2017. Google Cardboard. Retrieved April 26, 2018 from https://vr.google.com/intl/en_us/cardboard/.
[60] M. Graf, C. Timmerer, and C. Mueller. 2017. Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP. In *Proc. of ACM MMSys'17*. 261–271.
[61] S. Halik. 2017. OpenTrack: Head Tracking Software for MS Windows, Linux, and Apple OSX. Retrieved April 26, 2018 from https://github.com/opentrack/opentrack.
[62] B. Hayes, Y. Chang, and G. Riley. 2017. Omnidirectional adaptive bitrate media delivery using MPTCP/QUIC over an SDN architecture. In *Proc. of IEEE GLOBCOMM'17*. 1–6.
[63] B. Hayes, Y. Chang, and G. Riley. 2018. Controlled unfair adaptive 360 VR video delivery over an MPTCP/QUIC architecture. In *Proc. of IEEE ICC'18*. 1–6.
[64] B. Hayes, Y. Chang, and G. Riley. 2018. Scaling 360-degree adaptive bitrate video delivery over an SDN architecture. In *Proc. of IEEE ICNC'18*. 604–608.

[65] G. He, J. Hu, H. Jiang, and Y. Li. 2018. Scalable video coding based on user's view for real-time virtual reality applications. *IEEE Communications Letters* 22, 1 (2018), 25–28.
[66] Heinrich-Hertz-Institution. 2017. HEVC Scalability Extension (SHVC). Retrieved April 26, 2018 from https://hevc.hhi.fraunhofer.de/shvc.
[67] S. Heymann, A. Smolic, K. Muller, Y. Guo, J. Rurainsky, P. Eisert, and T. Wiegand. 2005. Representation, coding and interactive rendering of high-resolution panoramic images and video using MPEG-4. In *Proc. of PPW'05*.
[68] D. Hoffman, A. Girshick, K. Akeley, and M. Banks. 2008. Vergence-accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision* 8, 3 (March 2008), 1–30.
[69] M. Hosseini and V. Swaminathan. 2016. Adaptive 360 VR video streaming: Divide and conquer. In *Proc. of IEEE ISM'16*. 107–110.
[70] X. Hou, Y. Lu, and S. Dey. 2017. Wireless VR/AR with edge/cloud computing. In *Proc. of ICCCN'17*. 1–8.
[71] C. Hsu, A. Chen, C., C. Huang, C. Lei, and K. Chen. 2017. Is foveated rendering perceivable in virtual reality: Exploring the efficiency and consistency of quality assessment methods. In *Proc. of ACM MM'17*. 55–63.
[72] Y. Hu, S. Xie, Y. Xu, and J. Sun. 2017. Dynamic VR live streaming over MMT. In *Proc. of BMSB'17*. 1–4.
[73] J. Huang, Z. Chen, D. Ceylan, and H. Jin. 2017. 6-DOF VR videos with a single 360-camera. In *Proc. of IEEE VR'17*. 37–44.
[74] W. Huang, L. Ding, G. Zhai, X. Min, J. Hwang, Y. Xu, and W. Zhang. 2019. Utility-oriented resource allocation for 360-degree video transmission over heterogeneous networks. *Digital Signal Processing* 84 (2019), 1–14.
[75] I. Hupont, J. Gracia, L. Sanagustin, and M. Gracia. 2015. How do new visual immersive systems influence gaming QoE: A use case of serious gaming with oculus rift. In *Proc. of QoMEX'15*. 1–6.
[76] S. Im, H. Ha, F. Rameau, H. Jeon, G. Choe, and I. S. Kweon. 2016. All-around depth from small motion with a spherical panoramic camera. In *Proc. of ACM ECCV'16*. 156–172.
[77] Cisco Inc. 2017. Cisco Visual Networking Index: Forecast and Methodology, 2017–2022. Retrieved April 26, 2018 from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.
[78] Facebook Inc. 2017. Facebook Spaces. Retrieved April 26, 2018 from https://www.facebook.com/spaces.
[79] Google Inc. 2017. ExoPlayer: An Extensible Media Player for Android. Retrieved April 26, 2018 from https://github.com/google/ExoPlayer.
[80] M. Inoue, H. Kimata, K. Fukazawa, and N. Matsuura. 2010. Interactive panoramic video streaming system over restricted bandwidth network. In *Proc. of ACM MM'10*. 1191–1194.
[81] ISO/IEC JTC1/SC29/WG11/N17197. 2017. *Algorithm Descriptions of Projection Format Conversion and Video Quality Metrics in 360Lib*. Standard. International Telecommunication Union.
[82] N. Jiang, V. Swaminathan, and S. Wei. 2017. Power evaluation of 360 VR video streaming on head mounted display devices. In *Proc. of ACM NOSSDAV'17*. 55–60.
[83] W. Jiang and J. Gu. 2015. Video stitching with spatial-temporal content-preserving warping. In *Proc. of IEEE CVPR'15*. 42–48.
[84] T. Kamarainen, M. Siekkinen, J. Eerikainen, and A. Yla-Jaaski. 2018. CloudVR: Cloud accelerated interactive mobile virtual reality. In *Proc. of ACM MM'18*. 1181–1189.
[85] S. Kasahara, S. Nagai, and J. Rekimoto. 2015. First person omnidirectional video: System design and implications for immersive experience. In *Proc. of ACM TVX'15*. 33–42.
[86] N. Khiem, G. Ravindra, and W. Ooi. 2012. Adaptive encoding of zoomable video streams based on user access pattern. *Signal Processing: Image Communication* 27, 4 (April 2012), 360–377.
[87] H. Kim, J. Yang, M. Choi, J. Lee, S. Yoon, Y. Kim, and W Park. 2018. Eye tracking based foveated rendering for 360 VR tiled video. In *Proc. of ACM IMMSys'18*. 484–486.
[88] H. Kim, J. Yang, M. Choi, J. Lee, S. Yoon, Y. Kim, and W. Park. 2018. Immersive 360° VR tiled streaming system for esports service. In *Proc. of ACM MMSys'18*. 541–544.
[89] J. Kim, W. Kim, S. Ahn, J. Kim, and S. Lee. 2018. Virtual reality sickness predictor: Analysis of visual-vestibular conflict and VR contents. In *Proc. of QoMEX'18*. 1–6.
[90] H. Kimata, M. Isogai, H. Noto, M. Inoue, K. Fukazawa, and N. Matsuura. 2011. Interactive panorama video distribution system. In *Proc. of ITUWT'11*. 45–50.
[91] H. Kimata, D. Ochi, A. Kameda, H. Noto, K. Fukazawa, and A. Kojima. 2012. Mobile and multi-device interactive panorama video distribution system. In *Proc. of IEEE GCCE'12*. 574–578.
[92] J. Kopf. 2016. 360° video stabilization. *ACM Transactions on Graphics* 35, 6 (Nov. 2016), Article 195, 9 pages.
[93] D. Krevelen and R. Poelman. 2010. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality* 9, 2 (June 2010), 1–20.
[94] G. Krishnan and S. Nayar. 2008. Cata-fisheye camera for panoramic imaging. In *Proc. of IEEE WACV'08*. 1–8.

[95] Y. Lai and C. Hsu. 2018. Refocusing supports of panorama light-field images in head-mounted virtual reality. In *Proc. of AltMM'18*. 15–20.
[96] P. Lamkin. 2017. Best VR Headsets 2017: HTC Vive, Oculus, PlayStation VR compared. Retrieved April 26, 2018 from https://www.wareable.com/vr/best-vr-headsets-2017.
[97] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, et al. 2017. The QUIC transport protocol: Design and Internet-scale deployment. In *Proc. of ACM SIGCOMM'17*. 183–196.
[98] C. Lee, A. Tabatabai, and K. Tashiro. 2015. Free viewpoint video (FVV) Survey and future research direction. *APSIPA Transactions on Signal and Information Processing* 4, 15 (Oct. 2015), 1–10.
[99] J. Lee, B. Kim, K. Kim, Y. Kim, and J. Noh. 2016. Rich360: Optimized spherical representation from structured panoramic camera arrays. *ACM Transactions on Graphics* 35, 4 (July 2016), Article 63, 11 pages.
[100] W. Lee, H. Chen, M. Chen, I. Shen, and B. Chen. 2017. High-resolution 360 video foveated stitching for real-time VR. *Computer Graphics Forum* 36, 115–123.
[101] L. Li, Z. Li, X. Ma, H. Yang, and H. Li. 2017. Co-projection-plane based 3-D padding for polyhedron projection for 360-degree video. In *Proc. of IEEE ICME'17*. 55–60.
[102] S. Lim, J. Seok, , and T. Kim. 2015. Tiled panoramic video transmission system based on MPEG-DASH. In *Proc. of ICTC'15*. 719–721.
[103] K. Lin, S. Liu, L. Cheong, and B. Zeng. 2016. Seamless video stitching from hand-held camera inputs. *Computer Graphics Forum* 35, 2 (May 2016), 479–487.
[104] Sony Interactive Entertainment LLC.2017. PlayStation: PlayStation Console, Games, Accessories. Home Page. Retrieved April 26, 2018 from https://www.playstation.com/en-us/.
[105] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu. 2017. 360° video viewing dataset in head-mounted virtual reality. In *Proc. of ACM MMSys'17*. 211–216.
[106] W. Lo, C. Fan, S. Yen, and C. Hsu. 2017. Performance measurements of 360° video streaming to head-mounted displays over live 4G cellular networks. In *Proc. of APNOMS'17*. 205–210.
[107] W. Lo, C. Huang, and C. Hsu. 2018. Edge-assisted rendering of 360° videos streamed to head-mounted virtual reality. In *Proc. of IEEE ISM'18*. 44–51.
[108] K. Loria. 2016. Virtual Reality Is About to Completely Transform Psychological Therapy. Retrieved April 26, 2018 from https://www.businessinsider.com/how-virtual-reality-is-used-for-ptsd-and-anxiety-therapy-2016-1.
[109] A. Mahzari, A. Nasrabadi, A. Samiei, and R. Prakash. 2018. FoV-aware edge caching for adaptive 360° video streaming. In *Proc. of ACM MM'18*. 914–922.
[110] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. Silva. 2017. VR is on the edge: How to deliver 360° videos in mobile networks. In *Proc. of ACM VR/AR Network'17*. 30–35.
[111] I. Markovic, F. Chaumette, and I. Petrovic. 2014. Moving object detection, tracking and following using an omnidirectional camera on a mobile robot. In *Proc. of IEEE ICRA'14*. 5630–5635.
[112] A. Mavlankar and B. Girod. 2010. Video streaming with interactive pan/tilt/zoom. In *High-Quality Visual Experience*, M. Mrak, M. Grgic, and M. Kunt (Eds.). Springer, 431–455.
[113] T. Merel. 2016. Augmented Virtual Reality Revenue Forecast Revised to Hit 120 Billion by 2020. Retrieved April 26, 2018 from https://seekingalpha.com/article/3808846-augmented-virtual-reality-revenue-forecast-revised-hit-120-billion-2020.
[114] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou. 2013. An overview of tiles in HEVC. *IEEE Journal of Selected Topics in Signal Processing* 7, 6 (Dec. 2013), 969–977.
[115] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic. 2018. Salnet360: Saliency maps for omni-directional images with CNN. *Signal Processing: Image Communication* 69 (2018), 26–34.
[116] MPlayer. 2017. MPlayer: The Movie Player. Home Page. Retrieved April 26, 2018 from http://www.mplayerhq.hu.
[117] M. Stengel, S. Grogorick, M. Eisemann, E. Eisemann, and M. Magnor. 2015. An affordable solution for binocular eye tracking and calibration in head-mounted displays. In *Proc. of ACM MM'15*. 15–24.
[118] M. Yu, H. Lakshman, and B. Girod. 2015. Content adaptive representations of omnidirectional videos for cinematic virtual reality. In *Proc. of ACM ImmersiveMe'15*. 1–6.
[119] A. Nasrabadi, A. Mahzari, J. Beshay, and R. Prakash. 2017. Adaptive 360-degree video streaming using scalable video coding. In *Proc. of ACM MM'17*. 1689–1697.
[120] L. Neng and T. Chambel. 2010. Get around 360° hypervideo. In *Proc. of MindTrek'10*. 119–122.
[121] A. Nguyen, Z. Yan, and K. Nahrstedt. 2018. Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction. In *Proc. of ACM MM'18*. 1190–1198.
[122] D. Nguyen, H. Tran, A. Pham, and T. Thang. 2017. A new adaptation approach for viewport-adaptive 360-degree video streaming. In *Proc. of IEEE ISM'17*. 38–44.
[123] D. Nguyen, H. Tran, A. Pham, and T. Thang. 2019. An optimal tile-based approach for viewport-adaptive 360-degree video streaming. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1, 29–42.

[124] D. Nguyen, H. Tran, and T. Thang. 2019. Adaptive tiling selection for viewport adaptive streaming of 360-degree video. *IEICE Transactions on Information and Systems* 102, 1 (2019), 48–51.
[125] D. Nguyen, H. Tran, and T. Thang. 2019. A client-based adaptation framework for 360-degree video streaming. *Journal of Visual Communication and Image Representation* 59 (2019), 231–243.
[126] M. Nguyen, D. Nguyen, C. Pham, N. Ngoc, D. Nguyen, and T. Thang. 2017. An adaptive streaming method of 360 videos over HTTP/2 protocol. In *Proc. of NICS'17*. 302–307.
[127] O. Niamut, A. Kochale, J. Hidalgo, R. Kaiser, J. Spille, J. Macq, G. Kienast, O. Schreer, and B. Shirley. 2013. Towards a format-agnostic approach for production, delivery and rendering of immersive media. In *Proc. of ACM MMSys'13*. 249–260.
[128] O. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Lim. 2016. MPEG DASH SRD: Spatial relationship description. In *Proc. of ACM MMSys'16*. Article 5, 8 pages.
[129] J. Nielsen. 2016. Nielsen's Law of Internet Bandwidth. Retrieved April 26, 2018 from https://www.nngroup.com/articles/law-of-bandwidth.
[130] G. Noronha, C. Alvares, and T. Chambel. 2012. Sharing and navigating 360° videos and maps in sight surfers. In *Proc. of MindTrek'12*. 255–262.
[131] D. Ochi, Y. Kunita, K. Fujii, A. Kojima, S. Iwaki, and J. Hirose. 2014. HMD viewing spherical video streaming system. In *Proc. of the ACM MM'14*. 763–764.
[132] D. Ochi, Y. Kunita, A. Kameda, A. Kojima, and S. Iwaki. 2015. Live streaming system for omnidirectional video. In *Proc. of IEEE VR'15*. 349–350.
[133] Oculus VR LLC. 2017. Facebook Oculus Rift. Home Page. Retrieved April 26, 2018 from https://www.oculus.com/.
[134] Canadian Association of Optometrists. 2017. Are Virtual Reality Headsets Dangerous for Our Eyes? Retrieved April 26, 2018 from https://opto.ca/health-library/are-virtual-reality-headsets-dangerous-for-our-eyes.
[135] J. Ohm and G. Sullivan. 2013. High efficiency video coding: The next frontier in video compression [standards in a nutshell]. *IEEE Signal Processing Magazine* 30, 1 (Jan. 2013), 152–158.
[136] M. Ohta, S. Nagano, K. Nagata, and K. Yamashita. 2015. Mixed-reality Web shopping system using panoramic view inside real store. In *Proc. of SA'15*. Article 15, 3 pages.
[137] J. Oliva. 2017. A Reference Client Implementation for the Playback of MPEG DASH via Javascript and Compliant Browsers. Retrieved April 26, 2018 from https://github.com/Dash-Industry-Forum/dash.js/.
[138] C. Ozcinar, A. Abreu, S. Knorr, and A. Smolic. 2017. Estimation of optimal encoding ladders for tiled 360° VR video in adaptive streaming systems. In *Proc. of IEEE ISM'17*. 45–52.
[139] C. Ozcinar, A. Abreu, and A. Smolic. 2017. Viewport-aware adaptive 360° video streaming using tiles for virtual reality. In *Proc. of IEEE ICIP'17*. 2174–2178.
[140] C. Ozcinar, J. Cabrera, and A. Smolic. 2019. Visual attention-aware omnidirectional video streaming using optimal tiles for virtual reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (March 2019), 217–230.
[141] Y. Pai, B. Tag, B. Outram, N. Vontin, K. Sugiura, and K. Kunze. 2016. GazeSim: Simulating foveated rendering using depth in eye gaze for VR. In *Proc. of ACM SIGGRAPH'16 Posters*. 75.
[142] Telecom ParisTech. 2017. MP4Box. Retrieved April 26, 2018 from https://gpac.wp.imt.fr/mp4box/.
[143] Telecom ParisTech. 2017. MP4Client. Retrieved April 26, 2018 from https://github.com/gpac/gpac/wiki/MP4Client.
[144] Telecom ParisTech. 2017. GPAC-Multimedia Open Source Project. Retrieved April 26, 2018 from https://gpac.wp.imt.fr/.
[145] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke. 2016. Perceptually-based foveated virtual reality. In *Proc. of ACM SIGGRAPH'16*. 17.
[146] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics* 35, 6 (2016), 179.
[147] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. Gross. 2015. Panoramic video from unstructured camera arrays. *Computer Graphics Forum* 34, 2 (June 2015), 57–68.
[148] S. Petrangeli, J. Hooft, T. Wauters, R. Huysegems, P. Alface, T. Bostoen, and F. Turck. 2016. Live streaming of 4K ultra-high definition video over the Internet. In *Proc. of ACM MMSys'16*. Article 27, 4 pages.
[149] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. Turck. 2017. An HTTP/2-based adaptive streaming framework for 360° virtual reality videos. In *Proc. of ACM MM'17*. 306–314.
[150] S. Petrangeli, F. Turck, V. Swaminathan, and M. Hosseini. 2017. Improving virtual reality streaming using HTTP/2. In *Proc. of ACM MMSys'17*. 225–228.
[151] B. Petry and J. Huber. 2015. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proc. of ACM AH'15*. 217–218.
[152] D. Pohl, X. Zhang, A. Bulling, and O. Grau. 2016. Concept for using eye tracking in a head-mounted display to adapt rendering to the user's current visual field. In *Proc. of ACM VRST'16*. 323–324.

[153] M. Prins, O. Niamut, R. Brandenburg, J. Macq, P. Alface, and N. Verzijp. 2013. A hybrid architecture for delivery of panoramic video. In *Proc. of EuroITV'13*. 99–106.
[154] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. 2016. Optimizing 360 video delivery over cellular networks. In *Proc. of ATC'16*. 1–6.
[155] P. Quax, J. Liesenborgs, P. Issaris, W. Lamotte, and J. Claes. 2013. A practical and scalable method for streaming omni-directional video to Web users. In *Proc. of ACM ImmersiveMe'13*. 57–60.
[156] R. Ju, J. He, F. Sun, J Li, F. Li, J. Zhu, L. Han. 2017. Ultra wide view based panoramic VR streaming. In *Proc. of ACM VR/AR Network'17*. 19–23.
[157] Y. Rai, J. Gutierrez, and P. Callet. 2017. A dataset of head and eye movements for 360 degree images. In *Proc. of ACM MMSys'17*. 205–210.
[158] Open Source Virtual Reality. 2018. OSVR. Home Page. Retrieved April 26, 2018 from http://osvr.github.io/.
[159] G. Regal, R. Schatz, S. Johann, and S. Suette. 2018. VRate: A Unity3D asset for integrating subjective assessment questionnaires in virtual environments. In *Proc. of QoMEX'18*. 1–6.
[160] M. Rerabek, E. Upenik, and T. Ebrahimi. 2016. JPEG backward compatible coding of omnidirectional images. *Applications of Digital Image Processing XXXIX* 9971, 10 (Sept. 2016), 1–12.
[161] M. Romero-Rondon, L. Sassatelli, F. Precioso, and R. Aparicio-Pardo. 2018. Foveated streaming of virtual reality videos. In *Proc. of ACM MMSys'18*. 494–497.
[162] Y. Sanchez, R. Skupin, C. Hellge, and T. Schierl. 2017. Spatio-temporal activity based tiling for panorama streaming. In *Proc. of ACM NOSSDAV'17*. 61–66.
[163] J. Sauer, J. Schneider, and M. Wien. 2017. Improved motion compensation for 360° video projected to polytopes. In *Proc. of IEEE ICME'17*. 61–66.
[164] R. Schafer, P. Kauff, R. Skupin, Y. Sanchez, and C. Weissig. 2017. Interactive streaming of panoramas and VR worlds. *SMPTE Motion Imaging Journal* 126, 1 (Jan. 2017), 35–42.
[165] R. Schatz, G. Regal, S. Schwarz, S. Suettc, and M. Kempf. 2018. Assessing the QoE impact of 3D rendering style in the context of VR-based training. In *Proc. of QoMEX'18*. 1–6.
[166] O. Schreer, I. Feldmann, C. Weissig, P. Kauff, and R. Schafer. 2013. Ultrahigh-resolution panoramic imaging for format-agnostic video production. *Proceedings of the IEEE* 101, 1 (Jan. 2013), 99–114.
[167] ITU Radiocommunications Sector. 2012. Methodology for the subjective assessment of the quality of television picture. *ITU-R Recommendation* BT.500, 13 (Jan. 2012).
[168] ITU Telecommunication Standardization Sector. 2008. Subjective video quality assessment methods for multimedia applications. *ITU-T Recommendation* P.910 (April 2008).
[169] R. Silva, B. Feijo, P. Gomes, T. Frensh, and D. Monteiro. 2016. Real time 360° video stitching and streaming. In *Proc. of ACM SIGGRAPH'16 Posters*. Article 70.
[170] A. Singla, S. Fremerey, W. Robitza, P. Lebreton, and A. Raake. 2017. Comparison of subjective quality evaluation for HEVC encoded omnidirectional videos at different bit-rates for UHD and FHD resolution. In *Proc. of ACM Thematic Workshops'17*. 511–519.
[171] A. Singla, S. Fremerey, W. Robitza, and A. Raake. 2017. Measuring and comparing QoE and simulator sickness of omnidirectional videos in different head mounted displays. In *Proc. of QoMEX'17*. 1–6.
[172] A. Singla, S. Goring, A. Raake, B. Meixner, R. Koenen, and T. Buchholz. 2019. Subjective quality evaluation of tile-based streaming for omnidirectional videos. In *Proc. of ACM MMSys'19*.
[173] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl. 2016. Tile based HEVC video for head mounted displays. In *Proc. of IEEE ISM'16*. 399–400.
[174] J. Son, D. Jang, and E. Ryu. 2018. Implementing 360 video tiled streaming system. In *Proc. of ACM MMSys'18*. 521–524.
[175] J. Son, D. Jang, and E. Ryu. 2018. Implementing motion-constrained tile and viewport extraction for VR streaming. In *Proc. of ACM NOSSDAV'18*. 61–66.
[176] A. Steed, S. Friston, M. Lopez, J. Drummond, Y. Pan, and D. Swapp. 2016. An 'in the wild' experiment on presence and embodiment using consumer virtual reality equipment. *IEEE Transactions on Visualization and Computer Graphics* 22, 4 (Jan. 2016), 1406–1414.
[177] G. Sullivan, J. Ohm, W. Han, and T. Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (Dec. 2012), 1649–1668.
[178] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai. 2018. Multi-path multi-tier 360-degree video streaming in 5G networks. In *Proc. of ACM MMSys'18*. 162–173.
[179] T. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J. Ohm, and G. Sullivan. 2016. Video quality evaluation methodology and verification testing of HEVC compression performance. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 1 (Jan. 2016), 76–90.
[180] J. Thatte, J. Boin, H. Lakshman, and B. Girod. 2016. Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax. In *Proc. of IEEE ICME'16*. 1–6.

[181] B. Thomas. 2012. A survey of visual, mixed, and augmented reality gaming. *Computers in Entertainment* 10, 1 (Oct. 2012), 3.
[182] I. Tosic and P. Frossard. 2009. Low bit-rate compression of omnidirectional images. In *Proc. of PCS'09*. 1–4.
[183] H. Tran, N. Ngoc, C. Bui, M. Pham, and T. Thang. 2017. An evaluation of quality metrics for 360 videos. In *Proc. of ICUFN'17*. 7–11.
[184] E. Upenik, M. Rerabek, and T. Ebrahimi. 2016. Testbed for subjective evaluation of omnidirectional visual content. In *Proc. of PCS'16*. 1–5.
[185] E. Upenik, M. Rerabek, and T. Ebrahimi. 2017. On the performance of objective metrics for omnidirectional visual content. In *Proc. of QoMEX'17*. 1–6.
[186] V. Couture, M. Langer, and S. Roy. 2011. Panoramic stereo video textures. In *Proc. of ICCV'11*. 1251–1258.
[187] M. Viitanen, A. Koivula, A. Lemmetti, A. Yla-Outinen, J. Vanne, and T. Hamalainen. 2016. Kvazaar: Open-source HEVC/H.265 encoder. In *Proc. of ACM MM'16*. 1179–1182.
[188] S. Vlahovic, M. Suznjevic, and L. Skorin-Kapov. 2018. Subjective assessment of different locomotion techniques in virtual reality environments. In *Proc. of QoMEX'18*. 1–6.
[189] Mirjam Vosmeer and Ben Schouten. 2017. Project Orpheus a research study into 360° cinematic VR. In *Proc. of ACM TVX'17*. 85–90.
[190] T. Waltemate, F. Hulsmann, T. Pfeiffer, S. Kopp, and M. Botsch. 2015. Realizing a low-latency virtual reality environment for motor learning. In *Proc. of ACM VRST'15*. 139–147.
[191] H. Wang, M. Chan, and W. Ooi. 2015. Wireless multicast for zoomable video streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 12, 1 (Aug. 2015), Article 5, 23 pages.
[192] H. Wang, V. Nguyen, W. Ooi, and M. Chan. 2014. Mixing tile resolutions in tiled video: A perceptual quality assessment. In *Proc. of ACM NOSSDAV'14*. 25–30.
[193] WebVR. 2017. WebVR: Bringing Virtual Reality to the Web. Home Page. Retrieved April 26, 2018 from https://webvr.info/.
[194] C. Wu, Z. Tan, Z. Wang, and S. Yang. 2017. A dataset for exploring user behaviors in VR spherical video streaming. In *Proc. of ACM MMSys'17*. 193–198.
[195] M. Xiao, S. Wang, C. Zhou, L. Liu, Z. Li, Y. Liu, and S. Chen. 2018. MiniView layout for bandwidth-efficient 360-degree video. In *Proc. of ACM MM'18*. 914–922.
[196] M. Xiao, C. Zhou, Y. Liu, and S. Chen. 2017. OpTile: Toward optimal tiling in 360-degree video streaming. In *Proc. of ACM MM'17*. 708–716.
[197] 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'18*. 953–961.
[198] S. Xiao and F. Wang. 2011. Generation of panoramic view from 360 degree fisheye images based on angular fisheye projection. In *Proc. of DCABES'11*. 187–191.
[199] 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 MM'17*. 315–323.
[200] L. Xie, X. Zhang, and Z. Guo. 2018. CLS: A cross-user learning based system for improving QoE in 360-degree video adaptive streaming. In *Proc. of ACM MM'18*. 564–572.
[201] S. Xie, Y. Xu, Q. Qian, Q. Shen, Z. Ma, and W. Zhang. 2018. Modeling the perceptual impact of viewport adaptation for immersive video. In *Proc. of IEEE SCAS'18*. 1–5.
[202] X. Xie and X. Zhang. 2017. POI360: Panoramic mobile video telephony over LTE cellular networks. In *Proc. of CoNEXT'17*. 336–349.
[203] Y. Xiong and K. Pulli. 2010. Color matching for high-quality panoramic images on mobile phones. *IEEE Transactions on Consumer Electronics* 56, 4 (Nov. 2010), 2592–2600.
[204] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang. 2018. Predicting head movement in panoramic video: A deep reinforcement learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 8, 1–15.
[205] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao. 2018. Gaze prediction in dynamic 360 immersive videos. In *Proc. of IEEE CVPR'18*. 5333–5342.
[206] Y. Sanchez, R. Skupin, and T. Schierl. 2015. Compressed domain video processing for tile based panoramic streaming using SHVC. In *Proc. of ACM ImmersiveMe'15*. 13–18.
[207] M. Yahia, Y. Le Louedec, G. Simon, and L. Nuaymi. 2018. HTTP/2-based streaming solutions for tiled omnidirectional videos. In *Proc. of IEEE ISM'18*. 89–96.
[208] S. Yen, C. Fan, and C. Hsu. 2019. Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol. In *Proc. of PV'19*.
[209] R. Youvalari, A. Aminlou, M. Hannuksela, and M. Gabbouj. 2016. Efficient coding of 360-degree pseudo-cylindrical panoramic video for virtual reality applications. In *Proc. of IEEE ISM'16*. 525–528.

[210] V. Zakharchenko, K. Choi, and J. Park. 2016. Quality metric for spherical panoramic video. In *Proc. of SPIE OP'16*. Article 99700C, 9 pages.
[211] A. Zare, A. Aminlou, and M. Hannuksela. 2018. 6K effective resolution with 4K HEVC decoding capability for OMAF-compliant 360 video streaming. In *Proc. of PV'18*. 72–77.
[212] A. Zare, A. Aminlou, M. Hannuksela, and M. Gabbouj. 2016. HEVC-compliant tile-based streaming of panoramic video for virtual reality applications. In *Proc. of ACM MM'16*. 601–605.
[213] L. Zhang, S. Amin, and C. Westphal. 2017. VR video conferencing over named data networks. In *Proc. of ACM VR/AR Network'17*. 7–12.
[214] Z. Zhang, Y. Xu, J. Yu, and S. Gao. 2018. Saliency detection in 360 videos. In *Proc. of ACM ECCV'18*.
[215] C. Zhou, Z. Li, and Y. Liu. 2017. A measurement study of oculus 360 degree video streaming. In *Proc. of ACM MM-Sys'17*. 27–37.
[216] M. Zink, R. Sitaraman, and K. Nahrstedt. 2019. Scalable 360° video stream delivery: Challenges, solutions, and opportunities. *Proceedings of the IEEE* 107, 4 (April 2019), 639–650.
[217] A. Zomet, A. Levin, S. Peleg, and Y. Weiss. 2006. Seamless image stitching by minimizing false edges. *IEEE Transactions on Image Processing* 15, 4 (April 2006), 969–977.
[218] S. Yao, C. Fan, and C. Hsu. 2019. Towards quality-of-experience models for watching 360° videos in head-mounted virtual reality. In *Proc. of QoMEX'19*. 1–3.

Received May 2018; revised April 2019; accepted April 2019