A Tutorial on Immersive Video Delivery: From Omnidirectional Video to Holography

Jeroen van der Hooft, *Member, IEEE,* Hadi Amirpour, *Member, IEEE,* Maria Torres Vega, *Senior Member, IEEE,* Yago Sanchez, *Member, IEEE,* Raimund Schatz, *Member, IEEE,* Thomas Schierl, *Senior Member, IEEE,* and Christian Timmerer, *Senior Member, IEEE*

***Abstract*—Video services are evolving from traditional two-dimensional video to virtual reality and holograms, which offer six degrees of freedom to users, enabling them to freely move around in a scene and change focus as desired. However, this increase in freedom translates into stringent requirements in terms of ultra-high bandwidth (in the order of Gigabits per second) and minimal latency (in the order of milliseconds). To realize such immersive services, the network transport, as well as the video representation and encoding, have to be fundamentally enhanced. The purpose of this tutorial article is to provide an elaborate introduction to the creation, streaming, and evaluation of immersive video. Moreover, it aims to provide lessons learned and to point at promising research paths to enable truly interactive immersive video applications toward holography.**

***Index Terms*—Immersive video delivery, 3DoF, 6DoF, omnidirectional video, volumetric video, point clouds, meshes, light fields, holography, end-to-end systems**

I. Introduction

The COVID-19 crisis has unveiled the importance of remote communication, with people wanting to meet, collaborate, teach, and consume video content online. Many video-conferencing tools and streaming services are available to accommodate this need, accounting for about 66% of Internet traffic in 2022 [1]. However, many applications require more than today's traditional two-dimensional (2D) content. For instance, a doctor looking to operate on a patient remotely requires a reliable three-dimensional (3D) model of the patient's body. A psychologist looking to treat a phobia through exposure therapy can benefit from immersive video, simulating real-life scenarios within a safe environment [2]. Numerous other examples can be found in healthcare, education, and entertainment [3, 4].

Over the last years, many applications have evolved toward more immersive modalities such as omnidirectional (or 360-degree) video [5]. Using a head-mounted display (HMD), this type of video allows the user to rotate their head in three directions (pitch, yaw, and roll), providing the user with three degrees of freedom (3DoF). However, to enable full immersiveness through an HMD, the user needs to be able to freely explore the virtual scene with six degrees of freedom (6DoF), *i.e.*, the body can move in three dimensions (along the x-, y-, and z-axis) as well. While 6DoF video could unlock the next generation of immersive experiences, it has yet to be adopted by the industry.

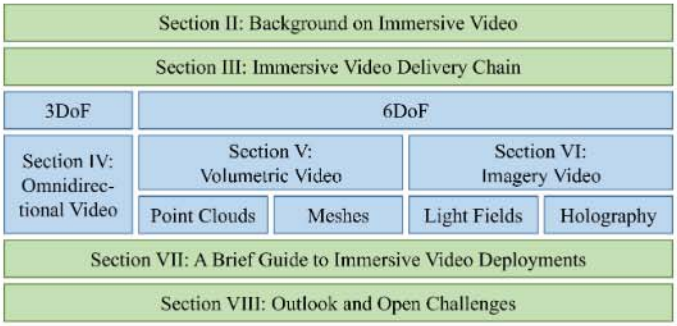

Fig. 1. Overview of the structure of this paper.

The purpose of this tutorial paper is to provide a thorough description of current solutions for immersive video delivery. In particular, the contributions of this article are as follows:

(i) We introduce a generic immersive video delivery chain, covering the required components for three relevant, immersive video formats: omnidirectional video, volumetric video, and 6DoF imagery video;
(ii) We review the latest developments related to algorithmic and software-based solutions for content capturing, compression, transmission, and quality perception in the context of video on demand (VoD), live video, and real-time communication;
(iii) We highlight challenges and potential research directions for future immersive video delivery architectures.

The remainder of this paper is organized as illustrated in Figure 1. Section II presents the background to this tutorial, focusing on different representations for immersive video and their applications in different domains. Section III introduces an end-to-end immersive video delivery chain with a high-level description of its components that this tutorial paper focuses on. Sections IV, V , and VI provide a thorough overview of current state-of-the-art advances in different immersive video formats, focusing on omnidirectional video, volumetric video, and image-based video, respectively. Section VII provides a brief guide to the deployment of 6DoF immersive video streaming. Finally, Section VIII outlines potential research

J. van der Hooft is with Ghent University, Belgium, email: jeroen.vanderhooft@ugent.be

H. Amirpour and C. Timmerer are with Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität Klagenfurt, Austria, e-mail: firstname.lastname@aau.at

M. Torres Vega is with KU Leuven and Ghent University, Belgium, email: maria.torresvega@kuleuven.be

Y. Sanchez and T. Schierl are with Fraunhofer/HHI, Berlin, Germany, email: thomas.schierl@hhi.fraunhofer.de

R. Schatz is with the AIT Austrian Institute of Technology, Vienna, Austria, email: raimund.schatz@ait.ac.at

Manuscript received October 22, 2021; revised Month Day, 2021.

Fig. 2. Immersive video options ordered by bandwidth requirements (from low to high) [5].

paths to make truly interactive immersive experiences a reality.

II. Background on Immersive Video

According to the Cambridge Dictionary, "immersion" is defined as "the fact of becoming completely involved in something" [6]. In visual media, the term refers to deeper viewer involvement in the content – be it in a story, a game, or a remote conversation. Most often, it is associated with virtual reality (VR), where the user can interact with an artificial 3D visual or other sensory environments through the use of computer modeling and simulation. In the former case, interaction is usually enabled through the use of an HMD, a display device that can be worn on the head and covers one or both of the user's eyes. By tracking the user's position and rotation through a built-in inertial measurement unit (IMU), the displayed video can be adapted to match the user's movement within a virtual environment.

Such interaction requires carefully designed video formats that allow the HMD to render the current field of view, *i.e.*, the extent of the observable world the user sees at any given moment. In this regard, a distinction has to be made between captured video and computer-generated imagery (CGI). In the former case, the virtual environment is created through video directly captured by advanced cameras or camera setups. Thus, events in the virtual environment reflect physical actions that occurred in the real world. In the latter case, the video is generated through computer graphics, an approach dopted in applications such as VR games and simulators. An example is the ITI VR crane & equipment simulator [7], which allows a crane operator in training to interact with physical joysticks. The addition of depth perception can make understanding the motion of the crane's chains easier, and being able to glance up/down or left/right (*e.g.*, to check the mirrors) can make the experience comparable to actually operating a crane [8]. While such CGI-based applications are very relevant, this article focuses on captured VR video.

The remainder of this section introduces the three main immersive video formats currently in use, ordered by bandwidth requirements from low to high (as shown in Figure 2). Section II-A provides details regarding omnidirectional video, Section II-B deals with volumetric media, and Section II-C delves into light-field video. In addition to describing their working principles, these sections also analyze their real-life applications.

Fig. 3. Virtual reality exposure therapy applied to arachnophobia [2].

A. Omnidirectional Video

The concept of panoramic or omnidirectional photography has been around since the mid-1800s when the first patent for a 150-degree camera was granted [9]. By the end of the 19th century, the Al-Vista, the first mass-produced omnidirectional panoramic camera, was released in the United States [10]. In the 100 years that followed, panoramic cameras were further advanced. By the late 1990s, hardware and software were sufficiently developed to allow the stitching of multiple images together. The first omnidirectional videos were created (*e.g.*, at Doo Interactive Offices [11]) in the early 2000s, and the first commercial omnidirectional video cameras (*e.g.*, the Ricoh Theta m15 [12]) were released in the 2010s. With the advent of these new devices, immersive video applications were introduced in several domains.

1) Healthcare: Omnidirectional video has played an essential role in mental health, with VR-based therapy actively being used to counter anxieties. By immersing patients in a virtual environment, they can be confronted with their fears at their own pace without being exposed to any physical dangers or discomfort. A recent study by Monaghesh *et al.* showed that integrating VR in therapy can improve symptoms in patients suffering from paranoia [13]. Another study by Minns *et al.* discussed VR exposure therapy (VRET) applied to self-contained interventions that require no therapist guidance [2]. Their results suggested that adopting automated VRET in the context of arachnophobia (*i.e.*, a fear of spiders) provides a promising self-help treatment under real-world conditions (see Figure 3). Omnidirectional video is also being used to prepare trainees for surgery, as is discussed below.

2) Education: Over the last decade, omnidirectional video has been used extensively for educational purposes. Here, the user is immersed in an authentic environment that stimulates learning. For example, the ability to look at a surgical field remotely with stereoscopic vision allows a trainee to learn what typically happens inside an operating room and get familiar with the required procedures followed by medical personnel. This can lead to a better perception of the workflow, team dynamics, and integration of additive technologies without the trainee needing to be present on-site [14]. Omnidirectional video can also be used to immerse children who cannot attend school (*e.g.*, due to long-term illness) in an authentic school environment or to make students familiar with specific

environments without the need for physical travel (*e.g.*, in the context of geography) or exposure to more challenging situations (*e.g.*, in the context of public speaking).

3) Entertainment: Several commercial products, such as YouTube [15] and Facebook [16], adopted omnidirectional content delivery in 2015 [17]. Where the focus was initially on video on demand, these services quickly moved to live video as well. Different capturing and broadcasting engines, such as XSplit [18], allow users to process omnidirectional camera inputs and direct the resulting video to these services. At the time of writing, both YouTube and Facebook offer omnidirectional video resolutions up to 4K [19, 20]. This allows users to stream and render high-quality video content, enabling them to remotely attend sports events, join an enthusiastic crowd at a concert, or enjoy a roller-coaster ride from within their homes.

B. Volumetric Video

Recently, more advanced applications have emerged. In 2021, ABBA announced the "Hologram Concert" tour, where an innovative show format will feature the four band members as holographic avatars [21]. Omnidirectional video is no longer sufficient in this case. While 3DoF solutions allow the user to turn their head, the subject's position is still fixed within the scene. Three additional degrees of freedom are required to allow total freedom of movement, resulting in immersive media with 6DoF.

To provide a 6DoF experience, all considered objects within the scene require a three-dimensional representation. Typically, two types of technologies to capture such content are considered: volumetric video-based and image-based solutions. In volumetric video-based solutions, objects are represented through a collection of points in space. Because the position of each point is known, the object can be rendered from any position and viewing angle [22]. Sampling many points of an object, each containing information on the geometry (x, y, z) and texture (*e.g.*, YUV or RGB values), results in a so-called point cloud.

The concept of point clouds can be extended to meshes, *i.e.*, collections of triangles that, together, form a three-dimensional representation of an object. The coordinates of the triangles' vertices can be used to render the object based on the user's position and viewing angle, taking into account the texture components of each triangle. Meshes are less suited for tiling and culling than point clouds because triangle and *uv* parameterization continuity cannot be respected. However, meshes better exploit graphics pipelines such as mipmaps and anisotropic filtering, resulting in higher visual quality at larger bitrates [23] (see Section V).

The first object to be captured in three dimensions is the Stanford bunny (see Figure 4), which was captured in 1994 as a model consisting of nearly 70 000 triangles [24]. Since then, the field of volumetric video has seen exceptional growth in development. Soon, it was used to make scans of people, objects, and even buildings and has proved essential in the success of early motion-tracking devices such as Microsoft's Kinect [25]. In recent years, volumetric video has also received

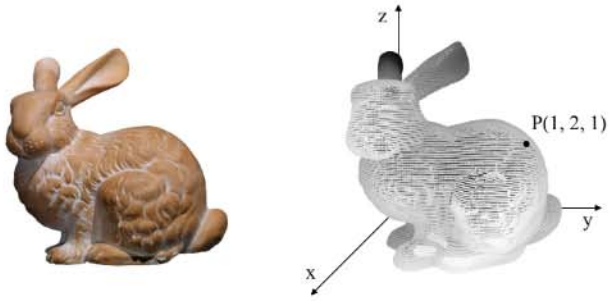

Fig. 4. Front image of the Stanford bunny [24] (left) and a point cloud representation (right). The object's geometry is determined by the (x, y, z) coordinates of each point.

increased attention as a means for the generation of more complex scenes. By capturing objects (*e.g.*, humans) with a grid-like camera setup, a 3D model can be generated by merging points captured by different cameras and views. Bringing multiple objects together into a single scene, an advanced immersive video experience can unfold (see Figure 5, where a scene is shown that is captured at 30 frames per second (FPS)).

Today, point clouds and meshes are used in a plethora of applications, including gaming, spatial planning, self-driving cars, and Industry 4.0. Below, only those applications related to captured volumetric video are presented in more detail.

1) Healthcare: Volumetric video can be used in remote medical consultations and procedures. For instance, providing a general practitioner with a 3D view of a patient's upper arm could enable a more substantiated diagnosis. Complex and urgent medical operations can be performed remotely by providing a surgeon with a high-quality visual feed, potentially extended with an auditory and tactile feed [26]. This use case requires ultra-high bandwidth (order of Gb/s) and ultra-low latency (order of ms). To meet these extreme requirements, an advanced network infrastructure, a limited physical distance to restrict propagation delay, and hardware capable of real-time en-/decoding are required to deliver the content. Due to these limitations, remote surgery has not become common practice yet.

2) Education: Similar to omnidirectional video, volumetric video can be used for educational purposes. Provided with the ability to move around with 6DoF, a user can familiarize her-/himself with new, unknown environments such as an operating room or the cockpit of an airplane. By extending such visual interaction with other sensory modalities such as touch (*e.g.*, through haptic gloves that can transfer a user's hand movement to the virtual environment), the user can also interact with 3D objects as if they were on-site. This opens up new paths for remote interaction-based learning and training.

3) Entertainment: The ability to create complex 3D scenes from captured video objects allows the user to move around in the immersive environment with 6DoF. This environment can consist of multiple human objects (*e.g.*, in the case of a virtual orchestra) or objects in an enclosed area (*e.g.*, a museum hall where several artifacts are displayed). Having the ability to freely explore the content (*e.g.*, walk in between the members of the orchestra or the audience or circle objects

Fig. 5. A scene consisting of four point cloud objects from the 8i dataset [27]. Every object is captured at 30 FPS, resulting in a volumetric video.

to inspect them from all sides) can increase one's sense of immersion, ultimately resulting in a more engaging and enjoyable experience of entertainment content (see Section V).

4) Remote Conferencing: Today, many video-conferencing tools are available to accommodate the need for remote communications. Truly immersive communication and social interaction, however, require users to interact with each other in the virtual environment as they would in the real world. To this end, a 3D model of all peers is required, captured in real-time, and sent over the network. This introduces several new challenges related to data processing, compression, and rendering (see Section V).

C. Light Fields and Holography

Similar to volumetric video, image-based approaches for immersive video have also seen a significant increase in research interest. Rather than using a 3D representation, image-based solutions render the view from a set of pre-acquired images, each captured at a different angle and tilt. As indicated in Section VI, current efforts focus on improving camera setups, light field compression, saliency detection, and rendering. In contrast to point clouds and meshes, however, live end-to-end content delivery is rarely considered by research. This is primarily due to the complex capturing task, which requires advanced setups with (ideally) multiple tens or even hundreds of cameras running simultaneously to create a unified view of the scene. Thus, the majority of works in this field so far has focused either on static content, where a single camera rig can be used to capture an object or scene over time or on video on demand, which allows ample time to preprocess all content before it is made available on the server side.

Ongoing research efforts at Google have recently advanced state-of-the-art light field video technologies. Broxton *et al.* [28] proposed a new camera rig and focused on light field compression and in-browser rendering. As a result of this research, 96 cameras can capture content simultaneously, resulting in videos that enable 6DoF, although with limited movement (a volume diameter of 70 cm) and limited visibility (a 180-degree field of view (FoV)). The system relies on cloud computing to process the videos, using hundreds to thousands of machines in parallel. To process 150 video frames, corresponding to five seconds of video at 30 FPS, a total of 4271 central processing unit (CPU) hours is required, which is equivalent to 28.5 CPU hours per frame [28]. The resulting light-field video can be consumed in a regular browser with a resolution of 1800×1350.

The above numbers illustrate why the end-to-end delivery of image-based solutions is currently not practical. In terms of computational and storage capacity, significant technical advancements are needed to enable the convenient streaming of light fields and holograms. All in all, the four limiting factors of volumetric media-based streaming also apply here: *(i)* tight synchronization between cameras in the same grid is required, *(ii)* high compression is needed to deal with the large amounts of generated data, *(iii)* bandwidth and latency requirements restrict contemporary content delivery services, and *(iv)* rendering often requires a significant amount of client resources, which are not available on low-end devices.

Light field image and holography imaging technologies are currently under development, with many ongoing research efforts to improve their technical capabilities and address the challenges associated with their end-to-end delivery. While their current applications are mainly limited to specific fields such as entertainment, medicine, manufacturing, and art, it is expected that in the near future, these technologies will find broader applications in various domains. With advancements in camera technology, compression techniques, and rendering methods, we expect to see an increased use of light field and holographic imaging in fields such as education, architecture, engineering, and construction. Additionally, once the technical challenges are overcome, we can expect to see more practical and convenient ways of delivering light field and holographic content, making these technologies more accessible to a wider audience.

III. Immersive Video Delivery Chain

All of the applications described in the previous section would benefit from a mature end-to-end system to capture, compress, deliver, and render content. However, 6DoF video has yet to be adopted by the industry. In fact, three significant barriers stand between current technology and remote immersive life-like experiences, namely *(i)* content realism, *(ii)* the motion-to-photon latency, and *(iii)* accurate human-centric quality control [29].

First, current capturing, encoding, and rendering techniques require intensive computation and expect bandwidths in the order of gigabits per second (Gb/s) [30] while still needing to achieve an entirely realistic representation. Second, the motion-to-photon latency, *i.e.*, the total delay between a change in the user's actions (*e.g.*, looking in a different direction) and the reflection of this change in the displayed content, should not exceed 10-20 milliseconds (ms) [31]. Third, to ensure that the end user feels present in the immersive environment, it is vital to keep their quality of experience (QoE) at the highest possible level [32, 33]. Lack of synchronization or quality degradations should be minimized to avoid feelings of cybersickness or loss of immersion.

To tackle these challenges, there is a need for optimizations in all areas of the delivery chain, presented in Figure 6. Below, an overview of the required system components is given. First,

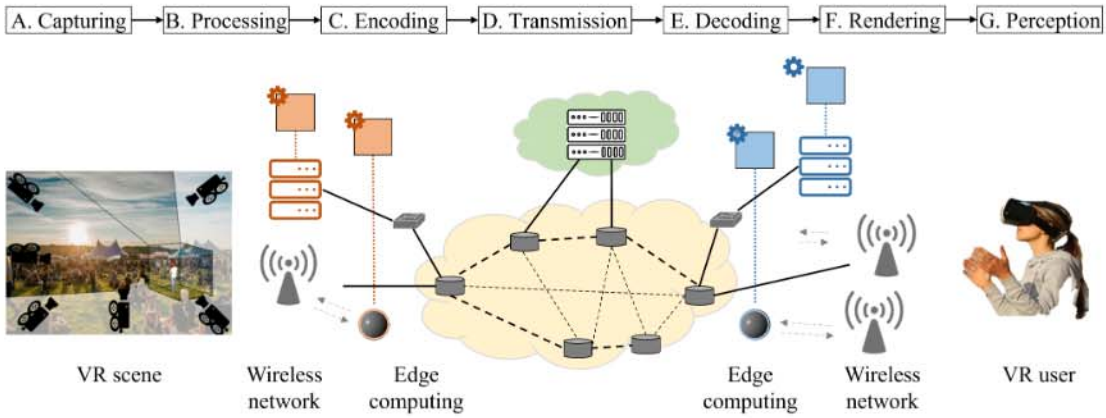

Fig. 6. End-to-end immersive video delivery chain.

Sections III-A, III-B, and III-C deal with the processes taken care of on the server side. Second, Section III-D provides a thorough overview of relevant network protocols and optimizations. Finally, Sections III-E, III-F, and III-G present the tasks required on the client side.

A. Capturing

When 3DoF are considered, omnidirectional video cameras are the de-facto standard. As a first step toward immersive video delivery, one or multiple cameras must capture the scenery and objects of interest. Existing camera systems can roughly be classified into three categories: mirror-based systems, systems with depth-aware stitching, and systems with depth-enabled light field rendering. Each of these systems has its advantages and drawbacks, which are discussed in Section IV.

When the user is offered 6DoF content, all considered objects within a scene require a three-dimensional representation. As introduced in the background section, objects can be captured either with image-based or volumetric video-based solutions. Image-based solutions require a representation at different angles and tilt, capturing a plethora of images either consecutively (*e.g.*, using a single camera array to capture a static scene) or simultaneously (*e.g.*, using multiple cameras to capture a dynamic scene). Image-based solutions thus rely on dense representations, where the video is made available on the server side as a sequence of images. Sparse representations, in contrast, require a collection of points in space containing information on geometry and texture. As the position of each point is known, the object can be rendered from any position and viewing angle [34].

Compared to image-based solutions, more complex processing is required (see the following subsection), and more computational resources are needed to display the content based on the user's position when using sparse representations. However, these solutions reduce storage and bandwidth requirements since there is no need to store and transfer an image for every angle and tilt. Sparse representations are discussed in Section V, making a distinction between point clouds and meshes; dense representations are discussed in Section VI, where both light fields and holography are addressed.

B. Processing

Once the content is captured by one or multiple cameras, it must be processed before compression can occur. In omnidirectional videos, for example, the spherical content is mapped on a 2D rectangular format using sphere-to-plane projection mapping (see Section IV). When multiple cameras are used (*e.g.*, in a camera rig to capture volumetric video), resulting feeds have to be merged in order to present the consumer with a single, unified representation of the content (see Sections V and VI).

C. Encoding

Regardless of the considered representation, *(i)* compression, *(ii)* random access, and *(iii)* packaging for view-aware streaming are needed to guarantee delivery over current and future network infrastructures.

1) Compression: Even when sparse representations are used, 6DoF solutions require significant amounts of data. As an example, a relevant point cloud dataset [27] contains objects (moving people) that require an average of 4.8 Gb/s per object at a frame rate of 30 FPS [36]. This is because for each of the captured points (approximately one million per frame in this dataset), geometry (x, y, z) and texture (*e.g.*, RGB values) are registered (see Section V). Advanced compression techniques are required to reduce the content's bitrate with throughput limitations in the order of 100 Mb/s to 20 Gb/s today.

2) Random Access: Compression techniques should not only focus on reducing the amount of data required to represent an object but also on so-called random access. In traditional video formats, it must be possible to decode a video sequence from particular points in time, *e.g.*, when picking up a live stream. This is realized by the provision of random-access points (RAPs), which break prediction to the past frames by using I-frames (key frames), which are typically inserted every two to ten seconds [37]. In the case of immersive video, random access is also needed under spatial segmentation. The client should be able to decode only those parts of the content that are relevant to the end user. This way,

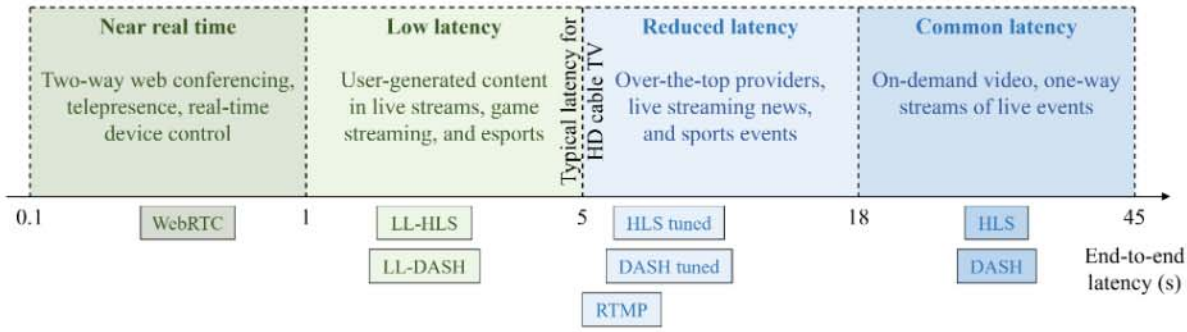

Fig. 7. Covered video streaming protocols as a function of the expected end-to-end latency [35].

TABLE I
ORDER OF MAGNITUDE OF SEVERAL PERFORMANCE METRICS RELATED TO DIFFERENT GENERATIONS OF MOBILE NETWORKS [38].

	3G	4G	5G	6G
Roll-out	2000	2010	2020	2030
Peak data rate	21 Mb/s	100 Mb/s	20 Gb/s	1 Tb/s
Latency	100 ms	10 ms	1 ms	100 µs
Rate per area	1 kb/s/m^2	100 kb/s/m^2	10 Mb/s/m^2	1 Gb/s/m^2
Devices/km	10^4	10^5	10^6	10^7

the bandwidth requirements are significantly reduced since only the region of interest (ROI), *e.g.*, a view, is transmitted to the end user. A trade-off, however, is established between the compression efficiency and random access that should be addressed.

3) Packaging: The content can be prepared for streaming once compressed. To this end, the video is generally packaged into streamable units containing several frames (temporal aspect) or regional parts (spatial aspect) of the video. While separate files can be used for this purpose (*e.g.*, each containing two seconds of video), specific byte ranges can also be retrieved based on the streaming session's metadata (more on this in the subsection below).

Several compression techniques have been proposed in the last few years, targeting one or more of these three aspects for omnidirectional video, point clouds, meshes, light fields, or holography. The main challenge is to find the right balance between the execution time of the encoding and the compression ratio: the former should be as low as possible, particularly when live video is considered, in order to limit the end-to-end delay, while the latter should be as high as possible to limit the bandwidth required for content delivery. In Sections IV, V, and VI, an overview of state-of-the-art approaches will be provided.

D. Transmission

Once the content has been captured and encoded, it can be transmitted to the end user. Naturally, the adopted network carrier plays an essential role in the transmission of video content. Several network technologies and standards have been developed in the last decades, resulting in the mobile networks we know and use today. To illustrate ongoing progress, Table I provides an overview of several metrics related to different generations of mobile networks. Even though 3G's and 4G's peak data rates are 21 Mb/s and 100 Mb/s, lower values are often recorded in practice [39, 40].

Today, 5G allows for a theoretical maximum of 20 Gb/s, with a recent measurement study reporting values up to 3 Gb/s [41]. While these values are generally sufficient to enable high-quality omnidirectional video (see Section IV), they do not necessarily suffice to enable highly interactive immersive video experiences with 6DoF movement. In this regard, the promise of 6G to allow for a peak data rate of up to 1 Tb/s is of utmost importance for immersive video services.

In light of interactivity, network latency also plays a crucial role. Whereas 3G and 4G offered latencies between 100 ms and 10 ms, 5G enables latencies down to 1 ms. This is especially important in applications requiring tactile feedback (*e.g.*, remote surgery), which requires an end-to-end latency of 1 ms [42]. The promise of 6G to even further reduce latency is expected to provide significant opportunities for immersive video applications.

The latency reported above corresponds to the access network, which, as pointed out, might be crucial for some immersive applications, *e.g.*, involving tactile feedback. However, we are primarily interested in the end-to-end delay of the system, as explained in the following paragraph. Figure 7 presents an overview of relevant video streaming protocols, indicating the expected end-to-end delay for traditional 2D video. Taking this delay into account, a distinction can be made between three types of delivery: *(i)* VoD, where the content has already been captured and encoded, and the end-to-end delay is of no importance; *(ii)* live, where the end-to-end delay is ideally limited to the order of seconds; and *(iii)* real time, where the end-to-end delay should remain lower than a few hundreds of milliseconds. These three types are elaborated upon below.

1) Video on Demand: Today, most on-demand content providers use approaches based on the hypertext transfer protocol (HTTP) combined with the transmission control protocol (TCP) for video streaming. While early approaches used progressive downloads, the concept of HTTP adaptive streaming (HAS) is now the go-to approach for HTTP-based content delivery. Using HTTP allows reusing the existing optimized and scalable network infrastructure of the Internet, while firewall and network address translation (NAT) traversal is guaranteed.

In HAS, the video content is encoded using several quality representations, temporally segmented, and stored within a

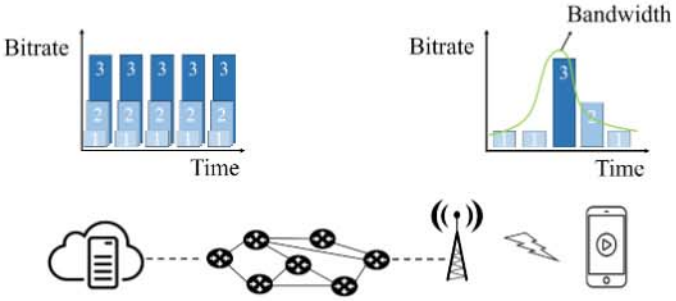

Fig. 8. The concept of HAS [43]. The video is encoded at different bitrates and temporally segmented so that the client can change the quality of the video based on, *e.g.*, available bandwidth.

content delivery network (CDN) (see Figure 8). During a streaming session, the client uses a rate adaptation heuristic to decide on the quality of each of the segments based on the network conditions, the buffer status, the device characteristics, and the user's preferences [44]. The segment duration is generally between one to ten seconds, depending on the provider and the considered use case.

Early HAS solutions include Microsoft smooth streaming (MSS) [45] and Apple's HTTP live streaming (HLS) [46], developed in 2009. While MSS is used by 29% of video developers today, HLS is used by 73% [47]. Although support for HLS was initially limited to iOS devices such as iPhones and iPads, native support has since been added to a wide range of platforms, including Android, Linux, and Microsoft devices. It has now grown to be the most prominent delivery format for HTTP-based delivery.

Next to these proprietary suites, the media pictures expert group (MPEG) has defined protocols and interfaces for HAS in the dynamic adaptive streaming over HTTP (DASH) standard, which was finalized in 2011 [48]. DASH defines, among others, the content of the media presentation description (MPD), which contains the required metadata for the client (*e.g.*, the base uniform resource locator (URL) and the available quality representations). This standard allows compliant players to request and play content from any HTTP server, increasing reusability, scalability, and reach.

Sections IV, V, and VI will discuss how recent studies and commercial products have adopted HAS to deliver immersive video on demand.

2) Live Video: In contrast to on-demand video streaming, live video streaming ideally limits the end-to-end delay to the order of (tens of) seconds. Once the content has been captured and processed, it needs to be made available as quickly as possible. To this end, an ingestion protocol such as the real-time messaging protocol (RTMP) is typically used. RTMP is a TCP-based protocol that maintains persistent connections to deliver video content [49]. It defines several virtual channels with a specific task, such as handling remote procedure calls (asynchronously) and sending video stream data, audio stream data, or control messages. Contrary to HTTP-based solutions, RTMP is a stateful protocol requiring a dedicated streaming server to deliver the content. This hampers scalability, which is one of RTMP's major drawbacks. Nevertheless, RTMP is used by broadcasters to overcome limited playback support

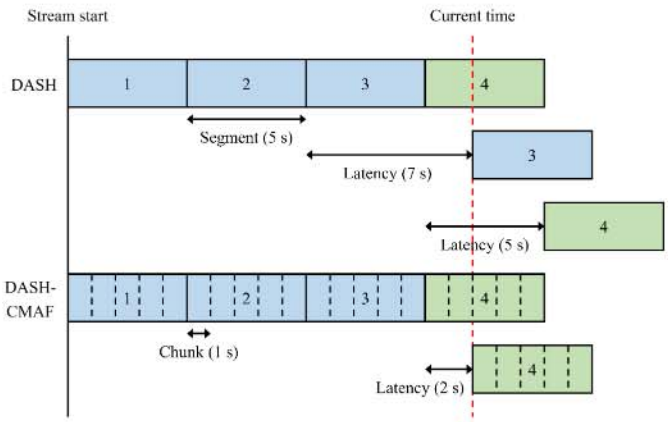

Fig. 9. The concept of LL-DASH [50]. Because of CMAF's higher granularity, the live delay can be significantly reduced compared to traditional DASH.

by initially encoding their live streams with RTMP and then transcoding the content for delivery to a range of players and devices. Service platforms such as YouTube and Facebook rely on the protocol to ingest the captured content to the cloud, where it can be transcoded and made available to HAS-based solutions, which provide higher scalability.

The content must be made available to the client in the next step. In an effort to overcome the higher latency observed in HTTP-based content delivery, the low-latency HLS (LL-HLS) protocol was introduced by Apple in 2019 [46]. LL-HLS reduces latency by dividing video segments into smaller parts – "partial segments" – listed separately in the client's playlist. Contrary to regular segments, a partial segment does not need to contain an I-frame since it is not expected to be decodable independently. When a video streaming session starts, the client requests the most recent partial segment and part of the streaming data containing the most recent I-frame to decode the video. Partial segments can have a duration as low as 200 ms so they can be packaged and published much earlier than their parent segment. This allows the client to remain closer to the live signal, resulting in a reduced end-to-end delay.

DASH has adopted low-latency streaming over HTTP using the CMAF with chunked transfer encoding (CTE) [51]. Similar to the approach used in LL-HLS[1], CMAF-CTE divides video segments into smaller parts – referred to as "chunks" – that can be requested independently of one another. An illustration of this concept is shown in Figure 9, where a segment duration of five seconds and a chunk duration of one second are considered. When playout is immediately started for regular DASH, the previously released segment is retrieved and played out, resulting in a delay of seven seconds. When the client waits for a new segment to be released, the playout has to be delayed for three seconds, after which a minimum delay of five seconds is perceived. When CMAF is considered, however, the available chunks can be retrieved immediately, resulting in a lower end-to-end delay.

Another approach to reducing latency involves adopting QUIC for video delivery [52]. QUIC is a protocol built

[1]Note that both LL-HLS and LL-DASH are compatible with the CMAF standard with respect to the segment format.

on top of user datagram protocol (UDP) that multiplexes several streams – each carrying a unique file or resource – to cover a single connection. This approach avoids head-of-line (HOL) blocking on the transport layer, resulting in faster recovery. Unlike TCP, QUIC's handshake during connection establishment combines the negotiation of both cryptographic and transport parameters. This means that a single negotiation for QUIC and transport layer security (TLS) is sufficient, reducing the connection establishment time by one round-trip time (RTT). Furthermore, since QUIC runs in user space rather than kernel space, the protocol allows customizing the transport layer to the application's needs [53]. Because of these advantages, QUIC is now supported by major browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. In the following sections, the application of QUIC to both omnidirectional video and volumetric video delivery will be discussed.

3) Real-Time Video: The latency due to TCP's congestion and flow control, along with its in-order delivery requirement, makes TCP unsuitable for real-time communication. While UDP-based approaches with built-in reliability on the application layer (*e.g.*, real-time transport protocol (RTP)) result in lower delays, they generally do not go below the one-second threshold. However, Web real-time communication (WebRTC) is a suite of real-time communication protocols that reduce the end-to-end delay to a few hundreds milliseconds [54]. Google released WebRTC as an open-source project in 2011 as a means of real-time communication between browsers, mobile platforms, and Internet of things (IoT) devices. It has been adopted by 25 to 28% of streaming services [35], particularly for remote video conferencing. WebRTC is, however, peer-to-peer in nature, requiring each sender to encode a separate stream for each of the receivers. This hampers scalability so that only smaller groups of clients can directly communicate with each other. Although some approaches have been proposed to improve scalability and increase video quality (*e.g.*, [55]), WebRTC cannot deliver high-quality video at scale. Still, it has shown promising results in augmented reality and volumetric video streaming, as discussed in Sections IV and V.

4) In-Network Optimizations: With the adoption of 5G, software-defined networking (SDN) also becomes feasible. SDN allows for programmatically and dynamically adjusting network configurations by separating the data layer from the control layer. Packets are sent on the former, while their routing is softwarematically defined by the latter [56]. This allows for intelligent decision-making in the network, including bandwidth shaping, packet prioritization, rerouting, and caching. In the context of 6DoF video streaming, SDN can be adopted to meet stringent requirements in terms of bandwidth and latency. Initial SDN-based solutions have recently been proposed in the context of immersive video streaming, as discussed in the sections below.

On top of SDN, network function virtualization (NFV) allows network functions to be deployed as virtualized software entities running on commodity hardware. Various services in 6DoF video streaming can be mapped to respective network functions and deployed as a service function chain (SFC) [5]. The SFC can be distributed to different locations in accordance with various requirements such as hardware capacity, bandwidth, distance, latency, reliability, and their respective trade-offs. Multiple network functions can run parallel, reducing the processing delay [57]. Intelligently placing SFC components in the network, the end-to-end latency can be reduced significantly, as illustrated further in this tutorial.

Given the high complexity of capturing, encoding, and rendering immersive video, required operations may be unfeasible on end devices due to complexity constraints and energy consumption. Cloud and edge processing is an enabler for such services, offloading computational tasks to the network. Recent solutions have successfully applied in-network solutions in the context of omnidirectional and volumetric video, as shown in Sections IV and V.

E. Decoding

The decoding of delivered video content is closely dependent on the selected encoder. In the case of real-time video, both encoding and decoding times should be limited to a minimum. Typically, low-complexity compression techniques are used in this case, resulting in fast execution with a limited compression ratio. In the case of VoD, only the decoding step poses a limiting factor since the encoding can be executed offline. In practice, the encoder for the standardized AVC and HEVC is indeed of higher complexity than the decoder. Since both components are dependent on one another, compression will be discussed in a single subsection in Sections IV, V, and VI.

F. Rendering

Rendering of immersive video strongly depends on the content representation and the type of scenario (*e.g.*, 3DoF versus 6DoF streaming). Most often, HMDs render the content, ensuring the user feels immersed in the VR scene. Well-known examples include the HTC VIVE [58] and the Oculus Quest [59], although many other commercial HMDs are available today. Even though physical hardware is not the main topic of this paper, (in-network optimizations for) immersive video rendering will be discussed for different content representations.

G. Perception

It is crucial to continuously monitor the user's perception of the video streaming service on the client side. In the case of VR applications, this perception is directly related to the user's feeling of immersiveness, which can be understood as the combination of the user's perception of quality (*i.e.*, QoE) and the user's well-being (*i.e.*, absence of cybersickness).

The user's QoE has traditionally been assessed utilizing standardized tests [60]. These tests are performed by letting subjects rate the quality of impaired video sequences, where the average score over all subjects is the mean opinion score (MOS). However, since the users can explore a large environment, they likely watch different portions of the content. This

TABLE II
COMPARISON OF DIFFERENT CAMERA SYSTEMS [31].

System	Advantages	Disadvantages
Mirror-based	• Parallax-free setup • Easy stitching • Almost no overlap • High resolution (10K, 60 FPS) • Full lens control • Real-time 2D/3D processing • Capable of live transmission	• Bulky system • Calibration needed • Sensitive to damages
Depth-aware stitching (segmented stereo, stereo by extreme overlap)	• Small form factor, light weight • Robust and compact • Easy handling • No calibration • Existing stitching software • Established post production	• Usually closed system, no lens control • Restricted real-time processing, limited use for live events • Reduced resolutions (due to overlap) • Extreme distortions are possible for stereo by extreme overlap
Depth-enabled light field rendering	• Parallax-free • Enables producing novel views • Still error-prone process • No real-time capabilities yet	• Complex computing • Supervised post production

makes it difficult to compare scores among users. Furthermore, while subjective evaluations provide the most accurate manner to assess QoE, they can only be performed after the experience. To provide feedback while in the immersive session, objective metrics can be used to estimate the user's subjective perception. Several attempts exist to expand two-dimensional video metrics toward three-dimensional content, all of which will be discussed in the following sections.

With the increasing availability of VR systems, more frequent reports of cybersickness have appeared [61]. Questionnaires are the most common form of detection, where the simulator sickness questionnaire (SSQ) is the most popular option [62]. To cope with the required real-timeliness, several methods can be used. Verbal single-question rating scale questionnaires allow symptom severity analysis over time [63]. Alternatively, objective-based options, such as postural instability, have been proposed. Postural instability states that the user starts losing stability due to cybersickness. In practice, this approach can disturb the participant if not well embedded in the experience, requiring the user to position themselves on a standardized stance every few minutes. Both approaches will be discussed in the remainder of this manuscript.

IV. 3DoF Omnidirectional Video

This section provides a thorough explanation of the working principles of the building blocks presented in the previous section for the case of omnidirectional video. First, Section IV-A presents the working principles for capturing omnidirectional video. Then, Section IV-B deals with the encoding intricacies, preparing the content to be streamed through the network as VoD (Section IV-C) or live video (Section IV-D). Section IV-E deals with in-network optimizations for video delivery, while Section IV-F introduces currently used techniques for rendering and perception analytics. Finally, Section IV-G presents an overview of relevant datasets, covered studies, and surveys related to omnidirectional video streaming.

A. Capturing

Capturing a full omnidirectional video is typically done by multiple cameras mounted on a sphere. Each camera's views are stitched together to combine the individual views into a single sphere. Existing camera systems can roughly be classified into mirror-based systems, systems with depth-aware stitching, and systems with depth-enabled light field rendering. A summary of the advantages and disadvantages of these systems is included in Table II, based on [31].

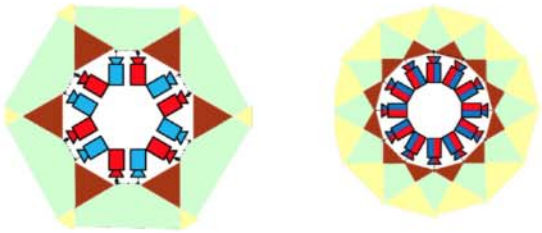

Fig. 10. Segmented stereo (left) and stereo by extreme overlap (right) [31].

Multi-camera arrangements for capturing such videos require the focal points of all camera views to coincide at a common point so that stitching can be performed ideally parallax-free, *i.e.*, so that there is no difference in the apparent position of an object viewed along different cameras. 3D content can be captured by stereo camera pairs with a relatively small overlap arranged in a star configuration. However, such systems typically suffer from parallax errors, which can be reduced by using mirror-based systems [64]. Another option is to use stereo camera pairs with extreme overlap. In Figure 10 on the right, the stereoscopic content is created from overlapping images captured by fish-eye or wide-angle lenses, or clusters of cameras.

B. Processing and Compression

In order to be able to compress omnidirectional video, it is essential to represent the content in a rectangular picture through sphere-to-plane projection mapping. The equirectangular projection (ERP) is the most basic and widely used. It is based on mapping longitude and latitude lines to even straight lines in the projected rectangular picture. Despite its wide support, ERP is a non-equal area projection, which means that it is not an area-preserving projection, *i.e.*, two regions of the projected rectangular picture with the same area do

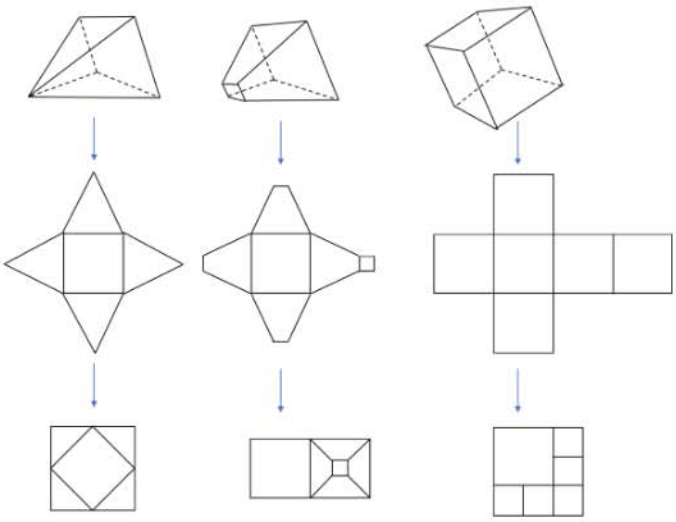

Fig. 11. Pyramid, truncated pyramid, and multi-resolution CMP generation.

not necessarily apply to an area on the sphere of the same dimension. This effect is highly noticeable around the poles, where ERP suffers from severe oversampling. This means that areas in the vicinity of the sphere poles are represented with a much higher number of pixels in the projected picture than other parts closer to the sphere equator. ERP is a projection typically used for viewport-independent streaming solutions, as discussed in the following section.

Several projections can be used for viewport-independent streaming solutions different from ERP that aim at preventing the geometric distortions inherent to ERP as they are detrimental to the coding efficiency of many codecs that employ a translatory motion model. Another commonly used projection is the CMP. In CMP, the camera surroundings are projected onto the six faces of a cube. Consequently, the sample value of each sample on a cube face stems from a rectilinear projection of the camera surroundings onto the position of that sample. The resulting pictures for each cube face are then arranged in the rectangular frame. Although CMP is also a non-equal area projection, the over-sampling and geometric distortion issues of ERP are sharply decreased. Hence, gains in coding efficiency can be demonstrated compared to ERP. Additional projections have been investigated in the last few years, achieving a projection closer to the equal area and thus reducing content discontinuities leading to an increase in coding efficiency. Although some significant gains have been achieved by such projections [65] (*i.e.*, up to around 10% compared to using ERP), such projections come at the cost of higher complexity. However, such projections are viewport-agnostic projections, *i.e.*, they do not take into account any particular viewport, and they, therefore, suffer from the problem that they include a substantial number of pixels for video areas that are not even presented to the user as they are located outside of the user's viewport.

A more efficient solution can be provided by viewport-adaptive coding and transmission schemes. This means that the content and, therefore, the streaming strategies are such that they adapt to the viewing direction of the user over time. Sphere-to-plane projections that achieve this purpose are herein referred to as viewport-specific projections. With these projections, a higher amount of pixels per degree is assigned to the content closer to the target viewport than to content farther away. Examples of such viewport-specific projections are shown in Figure 11: the pyramid projection (left), the truncated pyramid projection (middle), and a multi-resolution CMP variant (right). The figure illustrates how these three viewport-specific projections are generated. At the top, the geometric primitives are illustrated. At the same time, the second row shows the unrolled surfaces, and the bottom row gives possible arrangements of the polygon faces within a rectangular video frame. In the case of the pyramid or the truncated pyramid, the base of the polygon corresponds to the viewing direction of the user. Thus, the sampling density of the projected frame is highest in the area that the user observes. In the case of the multi-resolution CMP, the faces of the polygon not corresponding to the viewing direction of the user are downsampled before being arranged within the rectangular frame.

One of the drawbacks of viewport-specific projections is the large number of projections that need to be offered simultaneously for a service in order to be able to match any given user orientation and provide a smooth quality transition when switching from one viewing direction to another. Therefore, this solution comes with a considerable overhead cost for rendering on the content generation side, encoding, and transmission (*e.g.*, caching). An alternative solution is to offer an omnidirectional video split into several tiles and let a client choose which tiles to download, potentially at different resolutions, based on the user's current viewport, as described in the following subsection. This approach may require parallel decoders to decode multiple tiles, typically carried out in software leading to power consumption/battery issues. However, when constraint encoding is used employing motion-constraint tile sets as described in high-efficiency video coding (HEVC), a single hardware decoder can be used, provided all tiles are re-arranged in a single bitstream. For more information, the reader is referred to [66].

A further aspect to consider when encoding each of the individual tiles is that the encoded bitrate of each of the tiles needs to be accounted for a given target cumulative bitrate corresponding to the user's tile selection. Ideally, an operation point is selected that matches a reasonable target bitrate that is not exceeded for any possible viewing orientation's tile combinations. In order to avoid using a joint rate control for tiles that accounts for any of the possible combinations, which is a complex problem, Skupin *et al.* [67, 68] provide a solution that uses the spatiotemporal activity metrics, *i.e.*, the standard deviation of the Sobel filter and the standard deviation of frame differences over time. The spatiotemporal activity metrics of each of the tiles are used to estimate each tile's complexity and determine each tile's target bitrate separately while providing a solution for the combinatorial problem.

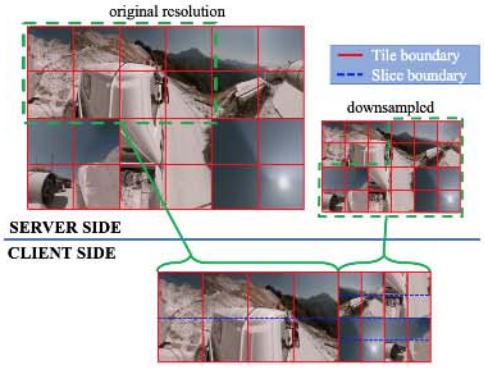

Fig. 12. Tiled high resolution (HR) and low resolution (LR) for omnidirectional video streaming.

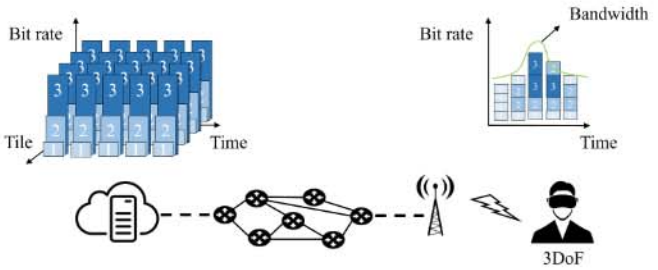

Fig. 13. The concept of HAS applied to tile-based omnidirectional video [43]. The video is encoded at different bitrates and both temporally and spatially segmented so that the client can change the quality of each tile based on, *e.g.*, available bandwidth.

C. Transmission: Video on Demand

As mentioned in Section II, omnidirectional video is typically not used for remote communications but for VoD or live scenarios in which the user enjoys 3DoF movement. This section discusses two flavors of VoD streaming for omnidirectional video: viewport-independent and viewport-dependent approaches.

1) Viewport-Independent Delivery: Viewport-independent approaches refer to streaming solutions that are agnostic to the users viewing orientation. They rely on transmitting the whole omnidirectional video to the user at a particular quality or resolution independent of the viewing orientation of a user at a particular time. Several commercial content providers, including YouTube and Facebook, use this approach. In this regard, no changes are needed compared to traditional video: the same rate adaptation heuristics can be used to deliver the content to the end user. It should be noted, however, that this approach wastes significant amounts of bandwidth on parts of the video that are never consumed. For this reason, viewport-dependent delivery has been proposed in the literature.

2) Viewport-Dependent Delivery: Viewport-dependent solutions consider information on the user's viewport when retrieving the content, allocating the most significant part of the available bandwidth to the user's region of interest. Two options exist: viewport-dependent projections and tile-based encoding.

In light of the former, Corbillon *et al.* [69] describe an approach based on viewport-specific encodings where a fixed number of streams matching different viewports are offered. When such a solution is used, two aspects need to be optimized. First, the number of streams matching a particular direction has to be determined and be high enough so that a smooth transition can be achieved when switching from one stream to another based on changes in the viewing direction of the user. Second, the streaming algorithm has to take into account the viewing direction of the user and adapt accordingly, *e.g.*, when using DASH, the client needs to change representation when the viewing direction changes as fast as possible so that the representation can be shown that has the highest visual quality on the viewport. The fact that several viewport-specific encodings are required for this approach to work correctly – and therefore, several such versions need to be stored at CDNs – poses a significant disadvantage.

In the case of tile-based encoding, the content is spatially segmented into different regions (*i.e.*, tiles). Each of the tiles is encoded at different quality representations (*e.g.*, using different resolutions or quantization parameter values) and made available in the CDN. The client can then choose at what quality to consume each tile depending on the user's viewport. This approach is illustrated in Figure 12, where the video is tiled and provided in two different resolutions. Although tile-based streaming requires encoding several bitstreams (*i.e.*, at least one per tile and quality representation), the approach comes with lower storage space requirements and overhead for coding and rendering compared to viewport-specific projections [70].

To client needs to be able to make informed decisions on the quality representation of each tile to use tile-based delivery effectively. As shown in Figure 13, this introduces an additional dimension to the rate adaptation process. In this regard, we discuss the application of the following components: *(a)* rate adaptation, *(b)* viewport prediction, and *(c)* saliency detection. Next to those, we also discuss the importance of random access.

a) Rate Adaptation: In traditional HAS-based solutions, rate adaptation is required to adapt the quality of the video to the network characteristics, the client's buffer, and the user's preferences. With tile-based solutions, an additional spatial dimension is added, resulting in higher complexity.

Tile-based streaming for panorama videos using MPEG-DASH has been studied in [71], with different tiles being downloaded at different qualities. Implementing the GPAC open-source player allows for the experimentation of different adaptation policies for tiled video content, which could consist of downloading all tiles at the same quality or prioritizing tiles within the region of interest. Hosseini and Swaminathan [72] show the benefits of using a viewport-aware adaptation technique for tile-based streaming of omnidirectional VR video. The authors identify three priority regions, assigning assigning a higher priority to tiles including or being closer to the viewport of the user, and increase the quality of each tile region per region as long as the bandwidth budget has not been exceeded following such a priority. Petrangeli *et al.* [73] propose an adaptive bitrate (ABR) heuristic for an advanced tiling scheme,

differentiating between regions close to the horizon (four tiles) and those near the zenith and nadir (one tile each). The quality of the six resulting tiles is then selected based on the user's current and predicted viewport position, considering the available bandwidth. Van der Hooft *et al.* [74] propose two ABR heuristics for generic uniform tiling schemes. These heuristics determine the great-circle distance between the viewport's center and each tile's center, and rank the available tiles accordingly. Tiles closer to the center are assigned a higher priority and, consequently, a higher quality representation. Because of this, a higher visual quality can be obtained compared to the ABR heuristics proposed by Hosseini and Swaminathan [72] and Petrangeli *et al.* [73].

Further studies for the streaming of omnidirectional video have been carried out based on reinforcement learning. For instance, Fu *et al.* [75] developed a hierarchical reinforcement-learning-based bitrate adaptation method named 360HRL. 360HRL introduces a re-downloading mechanism to tolerate the inaccurate viewport prediction and addresses the resulting complicated rate adaptation problem by two agents. The first agent is responsible for downloading a new segment for continuous playback or re-downloading an old segment to correct wrong bitrate decisions caused by inaccurate viewport estimation. The second agent determines the appropriate bitrates for the selected segments. Gains are compared to using the same algorithm without re-downloading older segments when inaccurate viewport prediction has been performed. Such an approach may only be helpful when the prediction is made for a considerable interval of a few seconds (*e.g.*, 3 s). Otherwise, re-downloading an old segment might not be feasible when predicting for shorter intervals of around 1 s in the future. However, a trade-off has yet to be compared between using a short-term prediction without a re-downloading step and a longer-term prediction with a re-downloading step.

b) Viewport Prediction: Quick adaptation to the changes in the viewing orientation requires a low-latency operation mode for DASH streaming. This requires the ABR algorithm to work with small buffer sizes, which is challenging. Algorithms working with small buffers can lead to very frequent quality switches to happen or playback interruption. Therefore, in order to provide a good quality experience to the users when using a viewport-dependent approach, it is crucial to use a streaming strategy that relies on predicting future user viewing orientation to be able to show high-quality content most of the time and still be capable of building large enough buffers to cope with throughput variations.

Several studies focus on predicting a user's viewport for omnidirectional video streaming. LaValle *et al.* [76] and Azuma [77] study the prediction of a fixation point using two methods that consider constant velocity and constant acceleration. More advanced prediction models have been proposed in the literature based on a linear regression model (LRM) or weighted LRM (WLRM) [78]. While no accurate comparison can be found among the described prediction algorithms, it can be expected that more complex algorithms outperform simpler ones. Even so, the authors show that inaccuracies always happens when predicting the future viewport: they report that when predicting 2 s in the future, only 80% of the predictions achieve an error lower than 10 degrees compared to the real viewing direction [78].

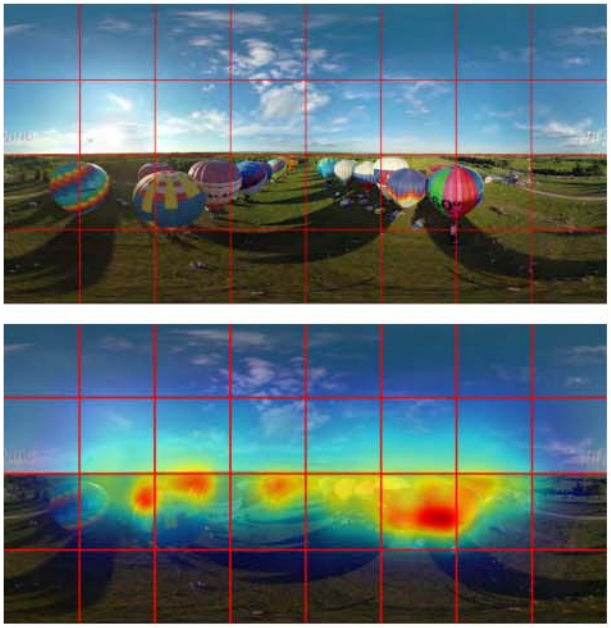

Fig. 14. Saliency detection for omnidirectional video [80], with the original content (top) and corresponding saliency map (bottom).

Irrespective of the algorithm used for the viewport prediction, the predicted value determines which tiles to download and how to prioritize those. For instance, an algorithm is developed in [79] that improves the performance of the tile-based streaming system by using a simple prediction model based on the current viewing orientation together with a movement speed (*i.e.*, very similar to [76]) that combined with a confidence value of the prediction steers an unequal quality distribution of the tiles within and outside of the predicted viewport. The confidence value is computed by deriving a correlation between the movement speed used for prediction and the time interval in the future that is predicted with the error that the prediction might make. Note that although a very simple prediction has been used in [79], a similar approach can be used with more complex prediction algorithms as long as a similar confidence value is derived.

c) Saliency Detection: While viewport prediction can be used to predict a single user's future movement and region of interest, saliency detection can be used to determine the most relevant parts of the video content based on historical data from other users. So-called saliency maps are generated by inferring regions/areas of an image that attract human attention in a scene, *e.g.*, by analyzing the viewing behavior of participants. As an example, Figure 14 shows the saliency map of a single frame from a study by Yang *et al.* [80]. The generated heatmap shows that foreground objects (*e.g.*, hot-air balloons) receive more attention than the background (*e.g.*, the sky). Knowing this, the client can anticipate the user's future focus and retrieve tiles that historically received greater visual attention at higher quality. Several studies consider such an approach tailored to omnidirectional video [81, 82, 83].

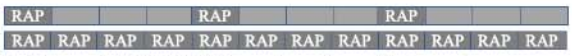

Fig. 15. Unequal RAP (URAP) configuration with two streams.

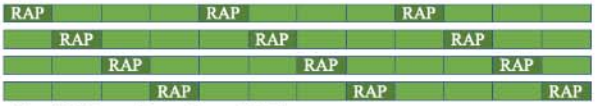

Fig. 16. SIDR configuration with four streams.

The concept of saliency detection can be combined with viewport prediction, incorporating the viewing history from prior users as well as information on the current user's movement. This requires the client to access historical data; it can thus not be used in live video streaming (see Section IV-D).

d) The Importance of Random-Access Points: The general assumption for tile-based omnidirectional video streaming approaches is that it is acceptable to show a lower quality/resolution for a short time, *e.g.*, when switching to another viewing orientation. However, fast switching capability is considered a required feature, as showing low quality/resolution content for a long time would lead to poor perceived performance. Sanchez *et al.* [79] analyze the impact of the adaptation delay in terms of the Bjontegaard delta (BD) rate. However, there has yet to be a proper subjective evaluation of the visual impact of shown lower quality/resolution for a short time. At the same time, informal experimentation points to a reasonably low acceptable value of a couple of hundreds of milliseconds (up to 300 ms).

Note that in order to be able to adapt quickly to changes in the viewing orientation, viewport-dependent approaches require the content to be encoded with frequent RAPs, *i.e.*, video frames which are encoded in such a way that they and frames following it in decoder order do not depend on previous frames. The time that a client needs to wait for a RAP to be available to switch to other streams that match the new viewport needs to be short enough, *i.e.*, lower than the mentioned 300 ms, so that low-quality content is not shown for so long. Therefore, RAPs must be available at least every 300 ms or more frequently. Since very frequent RAPs within a bitstream are detrimental to coding efficiency, a URAP configuration is typically used, where two versions of the content (each tile) are offered with different RAP periods. The client decides which one to download depending on whether a RAP is required, *e.g.*, due to the viewing orientation being changed or not. The URAP configuration is illustrated in Figure 15, where each box corresponds to a segment containing several frames, and those indicating RAP contain a RAP at the beginning.

Although the URAP configuration allows reducing the number of RAPs downloaded, once a switch to a short RAP period bitstream occurs, it is necessary to stay at that stream until a RAP is available at the long RAP period bitstream. Authors in [84] described another configuration, called SIDR, that allows to mitigate the penalty of frequent RAPs by increasing the RAP frequency available to a client without increasing the RAP frequency of individual encoded streams. The concept is illustrated in Figure 16 with four bitstreams, each of which has a long RAP period, but the RAPs are shifted so that the same availability of RAPs is achieved, as in Figure 15 for the URAP case. Authors in [85] show that using SIDR configuration compared to URAP configurations can provide gains from 3% to 6% for tile-based omnidirectional video streaming.

Irrespective of which schemes are used to provide frequent RAPs, the segment lengths for a viewport-dependent streaming service need to be kept short, *i.e.*, lower than the mentioned 300 ms, so that fast switching to match the user's viewport is possible. Note that such segment lengths are substantially shorter than what is typically used for video on demand and live video with HAS.

D. Transmission: Live Video

The solutions discussed so far cover on-demand scenarios only, with no end-to-end delay limitations. In this section, we discuss how recent developments related to low-latency streaming enable the live delivery of omnidirectional video and benchmark the experienced camera-to-display delay for the YouTube Live platform.

1) Adopting HTTP Adaptive Streaming: In live video scenarios, the content has to be released as quickly as possible once it has been captured. Complex tile-based encoding is generally not considered in this context, as it introduces a significant delay. Instead, the content is encoded and forwarded to the CDN through RTMP, as discussed in Section III. Once a new part of the content has been made available in the CDN, it can be requested by the client. To this end, HAS-based approaches can again be used. However, an essential difference with VoD scenarios is that, at any point in time, only a limited amount of new segments can be retrieved from the server. To know precisely what segments have been released, the client's MPD is typically updated periodically to announce the availability of new video segments.

By default, YouTube Live provides video through a *normal latency* mode, with a segment duration of five seconds. To establish the corresponding end-to-end delay, we run an experiment with omnidirectional video. For capturing, we use an Insta360 Pro 2 camera, which generates equirectangular video at either 1080p or 4K resolution and forwards the content to OBS Studio [86]. Then, OBS Studio uses RTMP to deliver the content to YouTube, with the recommended output bitrates of 4.5 Mb/s and 23.5 Mb/s for 1080p and 4K resolution, respectively. An Intel® Core™ i7-8850H CPU @ 2.60 GHz with an NVIDIA GeForce GTX 1050 Ti and 16 GB of random-access memory (RAM) is used to conduct the experiment.

Table III reports *(i)* the live delay between OBS and the video stream, as reported by YouTube, *(ii)* the live delay between OBS and the video stream, using a virtual clock, and *(iii)* the camera-to-display delay, using a physical clock. As can be observed, the default option for YouTube live video results in a camera-to-display delay of approximately 29.5 seconds for 1080p resolution and 44.2 seconds for 4K resolution. The difference between the two can be attributed to the cost of additional processing required to deal with the higher video resolution. Also worth observing is that the camera itself introduces an additional processing delay of 1.8 s and 3.2 s for 1080p and 4K resolution, respectively.

TABLE III
MEASURED END-TO-END DELAY FOR DIFFERENT LATENCY MODES IN YOUTUBE LIVE [87], USING AN INSTA360 PRO 2 CAMERA [88] THROUGH OBS STUDIO [86]. THE RESOLUTIONS (RES.) FOR CAPTURING (C) AND STREAMING (S) ARE REPORTED, ALONG WITH THE SEGMENT/CHUNK DURATION, THE REPORTED DELAY (RD), THE MEASURED DELAY (MD), AND THE CAMERA-TO-DISPLAY DELAY (CTD).

Res. (C)	Res. (S)	Latency	Duration (s)	RD (s)	MD (s)	CTD (s)
1080p	1080p	Normal	5 (segment)	27.3	27.7	29.5
1080p	1080p	Low	2 (segment)	5.2	5.6	7.4
1080p	1080p	Ultra-low	1 (chunk)	2.0	2.4	4.2
4K	4K	Normal	5 (segment)	40.5	41.0	44.2
4K	1440p	Low	2 (segment)	6.2	6.7	9.8
4K	1080p	Ultra-low	1 (chunk)	3.0	3.6	6.8

2) Parameter Tuning: Parameter values of both DASH and HLS can be tuned to reduce the end-to-end latency. By lowering the segment duration to two seconds, the encoding process can start as soon as two seconds' worth of frames is available. The buffer on the receiver's end can also be reduced (*e.g.*, to ten seconds) since the granularity of the segments is significantly higher. However, using a lower segment duration does come with an encoding overhead. Since segments are typically independently decodable, an I-frame is needed at the start of every segment. Thus, higher bitrates are required to achieve the same visual quality [89]. Note that, as discussed in IV-C, viewport-dependent delivery requires segments of around a few hundred milliseconds of length, which are substantially shorter than typically used for live streaming.

The reported values do not consider such short segments. Instead, when parameters are tuned with YouTube's *low-latency* profile, a segment duration of two seconds is used. As can be observed from Table III, this results in a camera-to-display delay of 7.4 seconds for 1080p resolution and 9.8 seconds for 4K resolution. Compared to the *normal-latency* mode, the delay can thus effectively be reduced by approximately 75%. However, it should be noted that the playout resolution is limited to 1440p for the low-latency profile [87]. Thus, there is a trade-off between the video quality and the delay.

3) Low-Latency Delivery: Same as for traditional video, it is possible to apply low-latency HTTP-based protocols to omnidirectional-video streaming. YouTube Live, for instance, offers support for LL-DASH with CMAF containers containing one second of video [90], and this is through the *ultra-low-latency* profile. As shown in Table III, this results in a camera-to-display delay of 4.2 seconds for 1080p resolution and 6.8 seconds for 4K resolution. Compared to the *normal-latency* mode, the delay can thus effectively be reduced by approximately 85%.

An important conclusion is that if the quality is important, a 4K resolution can be achieved with a camera-to-display delay of 44.2 s. In contrast, if latency is important, a 1080p resolution can be offered with a camera-to-display delay as low as 4.2 s. This illustrates clearly the trade-off between quality and delay for omnidirectional video streaming.

4) Partially Reliable Delivery: As discussed in Section III, QUIC can be used to lower the delay compared to HTTP-based approaches. Ravuri *et al.* [91] propose a hybrid delivery scheme based on QUIC, in which tiles within the user's viewport are sent reliably. In contrast, those not visible to the user are sent unreliably. Results show that the perceived throughput and the startup delay (*i.e.*, the time between the client requesting the video to start and the actual playout of the first segment) can be improved significantly compared to a scenario in which all data is sent reliably. For instance, adopting the proposed scheme in scenarios with 1 Tb/s throughput, 5 ms delay, and 5% packet loss reduces the startup delay by approximately 76% compared to video delivery over HTTP/2. However, the authors acknowledge that their approach suffers because HEVC, used to create independent video tiles, is less robust against packet loss. Indeed, Oztas *et al.* [92] showed that HEVC is more sensitive to packet loss than advanced video coding (AVC), especially in scenes with high motion. Consequently, non-reliable packet delivery in lossy networks can result in reduced video quality or, in some cases, a failure to render certain tiles.

E. In-Network Optimizations

While over-the-top delivery can be sufficient for omnidirectional video, some works consider the use of multi-access edge computing (MEC) to address the limited computation capability of VR devices [93]. Rather than decoding the received content on a local machine and rendering the video on the user's HMD, edge resources are used for these tasks.

In the case of non-tiled coding, the full video is transmitted from a CDN to an edge device. Here, the content is first decoded to a raw video format. Then, a 2D stream is generated that resembles the user's viewport based on the user's yaw, pitch and roll in the scene. To extract these values, the client's device can continuously forward information on the user's focus to the edge [94]. The matching viewport is then generated based on the forwarded values, and compressed using lightweight 2D video codecs. Resulting frames are sent out to the client's device as quickly as possible (*e.g.*, through UDP-based unicast) and rendered on the user's HMD.

A major drawback to this approach is the impact of the network latency between the client and the edge device on the motion-to-photon latency. By the time an update on the user's yaw, pitch and roll has arrived, the user might already have moved their focus slightly further. This can result in an increased delay between the user moving and the HMD rendering the corresponding viewport, ultimately leading to nausea and cybersickness (see Section IV-F). To address this issue, the user's viewport can be predicted on the edge device in real time. As an example, Liu *et al.* use recurrent neural networks (RNNs) to predict what part of the video will be consumed next by the user, based on historical observations. Results show that this approach reduces the interaction latency and increases the user's QoE compared to decoding the video and rendering the viewport on the client's device.

In the case of tiled coding, optimizations with respect to latency can go one step further by delaying the decoding of tiles that are currently out of scope of the user's region of interest. This results in fewer computations, ultimately reducing the decoding delay compared to an approach in which the whole scene is considered.

F. Perception

In contrast to traditional 2D video, QoE assessment and modeling for omnidirectional video have to address several additional challenges that make an adaptation of existing methods non-trivial. Beyond the impact of encoding, geometric distortions, and displays (not within the scope of this article), media delivery-related challenges are mainly driven by dynamic influencing factors such as temporal impairments like stalling, delays and quality fluctuations, different rendering strategies as well as viewport changes.

When experiencing omnidirectional video, stalling events (*i.e.*, intermittent playback freezes due to, *e.g.*, rebuffering) can strongly affect QoE and the viewer's sense of presence and immersion in ways comparable to traditional 2D video streaming. This has been demonstrated by the results of QoE studies like [95, 96, 97], in which subjects were exposed to different stalling patterns in traditional 2D screen-based and HMD omnidirectional video viewing settings. Despite some disagreement regarding the overall magnitude of the QoE impact of stalling in HMD viewing conditions (which is also influenced by end-user interaction and exploration behavior), existing research agrees that the mere presence of stalling events already significantly degrades QoE, turning stalling prevention (*e.g.*, by bitrate adaptation) into a key priority for omnidirectional video delivery.

In this respect, results of existing studies suggest that many stalling-related QoE phenomena and trade-offs known from traditional 2D video (see [98]) can be observed for omnidirectional video, too: for example, stalling is more tolerated in the beginning than toward the end of a clip (encouraging the use of larger playback buffers resulting in longer startup times), with single longer stallings being preferred over multiple shorter ones [97].

Since reducing omnidirectional video's genuinely high bandwidth requirements is critical for smooth playback, several studies have investigated the QoE impact of parameters like resolution, bit-rate, and quantization parameters (QPs). In this regard, the results of [99] and [100] suggest that at higher bitrates (above 1.5 Mb/s), choosing higher video resolutions (like UHD) generally enables higher QoE. Furthermore, acceptable QoE requires minimum bitrates ranging from 1.5 to 12 Mb/s, depending on the complexity and tempo of the content shown [100]. Moreover, bitrate changes (particularly relevant for adaptive omnidirectional video streaming) cause media quality fluctuations (including temporal distortions) that, in general, tend to have a higher QoE impact in lower bitrate playback situations [101].

While many fluctuation-related perceptual phenomena (*e.g.*, recency effects, peak-end rule) known from traditional 2D adaptive video streaming occur in omnidirectional video, too, the situation is more complex due to the impact of viewport changes and rendering strategy. In contrast to full-view streaming, viewport- and tiling-based streaming solutions introduce additional spatio-temporal fluctuations and artifacts due to selective streaming of different parts at different quality levels based on the current/predicted viewport position [102]. The resulting increase in the number of influencing factors and parameters (*e.g.*, tile grid, prediction algorithm used, *etc.*) represents a serious challenge for QoE research in terms of comparability, reproducibility, and generalizability of results (see [103] for a comprehensive overview of related studies and issues).

In addition, evaluating the consumption of omnidirectional videos with an objective metric, as done for traditional video (*e.g.*, for compression efficiency), is more complex. Some authors have been working in this field. Yu *et al.* [104] investigate how to assess the quality of omnidirectional videos using different projections. They have developed a sphere-based peak signal-to-noise ratio (S-PSNR) metric to approximate the average quality for all possible viewports. They also suggest a weighted S-PSNR metric that sets a higher weight to some sphere points than others. Such metrics have been used in standardization activities as described in [105]. The reliability and accuracy of these metrics have been verified by showing a good correlation with subjective quality assessments done by a group of experts and viewing orientation traces. Such evaluation methods allow the evaluation of the video quality of omnidirectional videos encoded using a viewport-agnostic projection without requiring viewing orientation traces. However, for viewport-dependent transmission schemes, the transmitted bitstreams vary depending on the temporary viewing direction of the user. Therefore, the peak signal-to-noise ratio (PSNR) is calculated for several viewport traces instead of using the described metrics. Thus, datasets with user traces are crucial for quality assessment and research on algorithms for improving viewport-dependent transmission (see Table IV).

As shown in Table VI, several surveys on omnidirectional video have appeared in the last years, where the topic of perception is frequently addressed. For instance, Xu *et al.* [114] focused their review on compression and perception, where perception was studied from subjective and objective perspectives. In another review, Chiariotti *et al.* [111] included an analysis of the possible factors that can affect QoE in omnidirectional video environments and a saliency analysis. While different aspects have been highlighted, the surveys agree that the open challenges regarding novel/adapted QoE techniques for omnidirectional video are related to saliency and viewport prediction. First, an adaptation of techniques from the 2D environments to omnidirectional video is required as the current 2D quality metrics have proven inaccurate. This means taking geometric distortion and viewer attention into account as well as other dynamic factors (*e.g.*, changes in quality). Second, accurate viewport prediction techniques will be critical in QoE estimation as they will clearly map the subjective assessments to the objective results derived from the content viewed by the user.

Cybersickness is usually not included in the surveys tackling the perception of omnidirectional video, but it has been analyzed independently. The SSQ has become the de-facto standard in the context of omnidirectional video [99, 116, 117, 118]. The SSQ consists of a set of questions regarding the severity of symptoms on a scale of 0–3. Scores are computed for three categories (nausea, oculomotor, and disorientation). The most comprehensive study can be found in [117], which addressed sickness in tile-based omnidirectional video stream-

TABLE IV
DATASETS RELEVANT TO OMNIDIRECTIONAL VIDEO STREAMING.

Dataset	Year	Description
360° Video Viewing [106]	2017	A dataset of both content data (such as image saliency maps and motion maps derived from omnidirectional videos) and sensor data (such as viewer head positions and orientations derived from HMD sensors).
AVTrack360 [107]	2018	A dataset of twenty different entertaining omnidirectional videos on an HTC Vive HMD in a task-free scenario.
Salient360 [108]	2018	A dataset of nineteen videos, along with 98 static images.
Wild-360 [109]	2018	An omnidirectional video saliency dataset, containing challenging videos with saliency heat-map annotations.

TABLE V
STUDIES RELEVANT TO OMNIDIRECTIONAL VIDEO STREAMING. A DISTINCTION IS MADE BETWEEN VIEWPORT-INDEPENDENT (VI) AND VIEWPORT-DEPENDENT (VD) APPROACHES. IN SOME CASES, PROTOCOLS AND EVALUATIONS ARE NOT AVAILABLE (N/A) OR NOT SPECIFIED (N/S).

	Study	Year	Target	Protocol	Evaluation	Focus
VI	Facebook [19]	2018	Live	DASH	N/A	Live streaming
	YouTube [20]	2018	Live	DASH	N/A	Live streaming
VD	Zare *et al.* [110]	2016	Both	N/A	Bitrate savings	HEVC-based tiling and encoding
	Skupin *et al.* [66]	2016	Both	DASH	N/A	HEVC-based tiling and encoding
	Hosseini and Swaminathan [72]	2016	VoD	DASH	Local (N/S)	Multi-tile bandwidth saving
	Corbillon *et al.* [69]	2017	Both	DASH	Bitrate savings	Viewport-specific encodings and adaptation
	Petrangeli *et al.* [73]	2017	VoD	DASH	Local (WiFi)	HTTP/2 server push and multi-tile rate adaptation
	van der Hooft *et al.* [74]	2019	VoD	DASH	Emulation (4G)	Multi-tile per-segment rate adaptation
	Sanchez *et al.* [79]	2019	Both	DASH	Bitrate savings	HEVC-based tiling and adaptation strategies
	Ravuri *et al.* [91]	2022	VoD	Custom (QUIC)	Emulation (5G)	Partially reliable video delivery

TABLE VI
SURVEYS RELEVANT TO OMNIDIRECTIONAL VIDEO STREAMING.

Authors	Year	Component	Description
Chariotti [111]	2021	Coding, quality perception	A survey presenting the latest developments in the relevant literature on four of the most important ones: *(i)* omnidirectional video coding and compression, *(ii)* subjective and objective QoE and the factors that can affect it, *(iii)* saliency measurement and viewport prediction, and (iv) the adaptive streaming of immersive omnidirectional videos.
Fan *et al.* [112]	2019	Capturing, delivery, rendering	A survey presenting the current literature related to omnidirectional video streaming for practical experiments. Therefore, it reviews systems built for real experiments and includes video and viewer datasets.
Ruan and Xic [103]	2021	Quality perception	A review focusing on the current state of QoE technologies applied to VR video streaming. The authors first pinpoint the main influencing factors of QoE and VR video streaming. Then, they summarize works focusing on user QoE for VR evaluation. Third, QoE modeling for VR video, QoE optimization and machine learning (ML) techniques are presented. The review finalizes with a set of current challenges and research directions.
Shafi *et al.* [113]	2020	Generic	A survey of omnidirectional video streaming on different projections, compression, and streaming techniques. This is combined with a review of the latest ongoing standardization efforts for enhanced degree-of-freedom immersive experience and an overview of the open research challenges.
Xu *et al.* [114]	2020	Compression, quality perception	A survey of on omnidirectional video/image processing from the aspects of perception, assessment and compression.
Yaqoob *et al.* [115]	2020	Generic	A survey on adaptive omnidirectional video delivery solutions considering end-to-end video streaming with a special focus on standardization efforts.

ing viewed with HMDs. Results show that while network delay clearly affects the QoE (above 47 ms RTT), the sickness propensity did not change significantly. The reason is that in tile-based streaming, only rendering behavior in motion-to-high-resolution latency is affected, but not motion-to-photon latency. Sickness is, however, influenced by the session duration [117] and the type of camera motion [119]. Even if broadly used, the SSQ has several drawbacks. Several symptoms contribute to more than one category. It can lead to oversensitivity and bias depending on the user taking the test. It takes too long to complete, making some participants lose their attention. Finally, as it was initially created for pilots in the air force [61], it lacks the generality for omnidirectional video applications.

G. Datasets, Studies, and Surveys

An overview of relevant datasets for omnidirectional video delivery is presented in Table IV. An overview of covered studies is presented in Table V, while relevant surveys are listed in Table VI.

V. 6DOF VOLUMETRIC VIDEO

This section discusses the required components for volumetric video streaming with 6DoF. First, Section V-A discusses how content is captured and preprocessed. Then, Sections V-B and V-C elaborate on the compression of both point clouds and meshes before presenting an overview of state-of-the-art approaches for transmitting volumetric video in Sections V-D to V-G. Sections V-H and V-I discuss the impact of quality degradation on the user's perception. Finally,

Fig. 17. Camera setup by Sky Studios, used to create a VR experience in the National History Museum in London, UK [120]. More than 100 depth cameras were used to create volumetric video.

Section V-J provides an overview of relevant datasets, covered studies, and surveys related to volumetric video streaming.

A. Capturing and Processing

Volumetric video is generally captured in two different ways. Light detection and ranging (LiDAR)-based cameras can be used to capture surroundings, such as an office, a building, or even an entire city [121]. However, these methods cannot be used for point cloud video as they are not suited for dynamic, close-range scenarios. Instead, specialized camera setups are typically used. Several production studios have recently built their own camera rig (see Figure 17). In these setups, specialized depth cameras (such as the Intel RealSense cameras [122]) simultaneously capture objects from different angles, merging different point clouds to form a single, unified scene.

This, of course, requires careful calibration of the different cameras. To automize this process, so-called point cloud registration techniques are used to detect the alignment of different point clouds. Today, the iterative closest point (ICP) algorithm, proposed by Besl and McKay in 1992 [123], is still one of the most effective approaches for mathematical optimization. ICP is a refinement algorithm that iteratively attempts to improve the alignment of two sets of points. To this end, it requires as input a reference and a source point cloud, a threshold τ for the reduction of the distance between consecutive iterations, and optionally an initial estimation of the required transformation to speed up the process (available when the angular distance between two cameras in a grid is known, for instance). Then, starting from either the provided transformation or the identity transformation, the following steps are repeatedly executed to refine the transformation:

1) For each point in the source point cloud, match the closest point in the reference point cloud (or a subset thereof).
2) Determine the transformation (rotation and translation) that minimizes the mean square error using the eigenvalues of the cross-covariance matrix.
3) Transform all points in the source point cloud using the resulting transformation.
4) Terminate when the difference in successive mean square errors exceeds the threshold τ, or go back to step 1 if not.

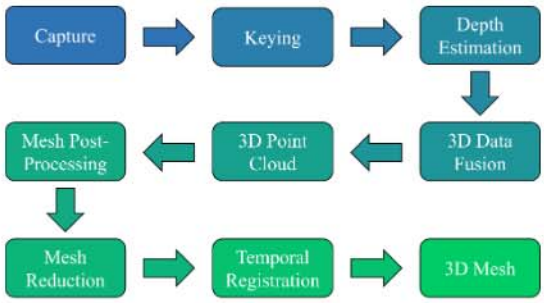

Fig. 18. 3D mesh generation and production workflow.

Fig. 19. Example of a dense depth map calculated per frame for each stereo camera pair.

It is possible to mathematically prove the convergence of this approach [123], resulting in the eventual termination of the algorithm. Because of its ability to refine transformations to high accuracy, ICP is supported by well-known tools for volumetric data processing, such as MeshLab [124] and the point cloud library (PCL) [125]. However, while optimization approaches are often used to refine the registration process of point clouds of limited size, their computational complexity does not allow them to handle the sheer volume of data (in the order of millions of points) needed for demanding applications such as autonomous driving and high-quality video delivery in real time. Furthermore, determining the optimal alignment of point clouds with limited overlap (such as those registered by different cameras) is not a straightforward task.

State-of-the-art approaches tend to apply deep learning (DL) techniques such as neural networks to find the required transformation from one point cloud to the other [126]. However, as discussed in a recent survey by Huang *et al.* [127], (deep) ML approaches currently suffer from a lower accuracy than optimization approaches. This is because many variations impact the point cloud registration process (*e.g.*, those introduced by (mobile) sensors or changes in the environment), which ML approaches are not well equipped to handle. Although the authors report that DL methods can achieve high registration accuracy on a specific dataset, their robustness and generalization ability to other datasets and applications are yet to be determined.

Once the content has been captured, a point cloud/mesh production workflow is followed (see Figure 18). First, a foreground and background segmentation process is performed in the keying stage. Then, a depth estimation process is applied in which depth information with high accuracy for each pixel is generated from each stereo pair with the 3D information, as illustrated in Figure 19. Afterward, a related 3D fusion

Fig. 20. Example of the resulting point cloud and 3D models such as meshing, simplification, and texturing (from second left to right).

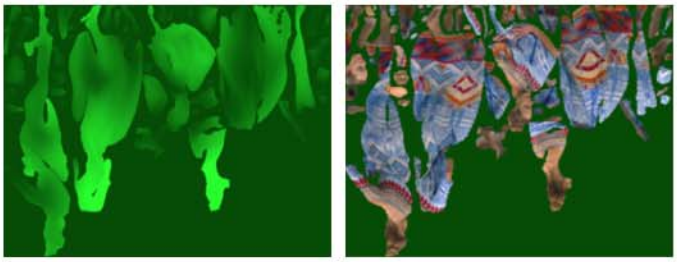

Fig. 21. Illustration of the patching process in the V-PCC codec, both for geometry (left) and texture (right) [22].

process is carried out, and the depth information from every stereo camera pair is merged, resulting in a 3D point cloud.

If a mesh representation is desired, a meshing step follows, consisting of a depth-based surface reconstruction that results in a high-density mesh with a large number of vertices and faces. A geometric simplification is performed next to simplify the resulting high-density mesh to a single consistent mesh, referred to as mesh reduction. The simplified meshes are texturized using a 2D texture map into a standard 2D image file format. In the final stage, the resulting meshes are temporally registered to obtain animated meshes. An example of the described process applied to the 3D point cloud is shown in Figure 20.

The application of volumetric video comes with limited storage and bandwidth costs since redundancy is kept to a minimum, *i.e.*, each point in space is represented at most once. Nevertheless, the size of the data is still significant: a single object comprising one million data points would require at least 6 MB if we consider three byte-sized integer coordinates (geometry) and three byte-sized integer color values (texture). The amount of data significantly increases when dynamic scenes with moving objects are considered. For instance, the 8i dataset [27] includes four dynamic point cloud objects with bitrates up to 5.7 Gb/s each. Approximately 19.2 Gb/s would be required to combine the four objects; thus, there is a need for point cloud compression (PCC).

B. Compression of Point Clouds

Many older works on PCC exist, including those by Gumhold *et al.* [128] and Merry *et al.* [129]. These works are based on a one-dimensional traversal of the point cloud, where the geometric distances between the points determine the traversal order. Compression performance is limited because it is impossible to take into account 3D spatial correlations [130] fully. For this reason, most approaches today are based on either the exploitation of three-dimensional correlations or mapping the point cloud to two-dimensional images by projection or mapping.

To exploit spatial correlations, kd- and octree-based approaches are often used. One example of the latter is the encoder proposed by Mekuria *et al.* [131], which recursively divides the bounding box of the point cloud object into eight subparts, corresponding to the child nodes of a tree-based structure. Non-empty children are subdivided further in each step, resulting in an octree of voxels. This subdivision is made for consecutive frames so that correlations can be used to achieve better compression performance. The iterative closest point algorithm computes a transformation, which is then compressed through a quaternion quantization scheme.

In 2017, MPEG launched a call for proposals on PCC [132], using the codec by Mekuria *et al.* [131] as a benchmark for point cloud video. Out of nine proposals, MPEG selected the one with the best performance (in terms of PSNR) as the reference encoder for V-PCC [22]. This codec decomposes a point cloud into a set of patches using orthogonal projection onto a two-dimensional grid. The resulting patches are merged into two separate video sequences containing the geometry and the texture information, respectively (see Figure 21). These sequences are then compressed using traditional video compression techniques, resulting in significantly higher compression rates for the same visual quality [22].

Several recent works have improved this codec by using different approaches to encode the patches generated by V-PCC more efficiently. Costa *et al.* [133] consider both patch sorting (*i.e.*, ordering the patches according to a given metric) and patch positioning (*i.e.*, placing the patches in a 2D frame) to do so. In their work, generated patches are first ordered according to a pre-defined absolute sorting metric (*e.g.*, using the patches' height as a sorting metric). Then, one by one, the best patch position is defined by performing an exhaustive search among all suitable positions and determining the one that optimizes a positioning metric (*e.g.*, the extension area, defined as the number of blocks associated with the extension of two patches). The compression efficacy can be increased by 0.8% using this approach.

Another compression approach that recently got traction is the adoption of super resolution (SR) to volumetric video. In traditional video, SR allows the creation of a high-quality version from a low-quality version and a generated model [134]. In learning-based SR, this model consists of a deep neural network (DNN), which is initially trained on high-quality frames. Then, in an inference phase, the server sends the low-quality version and the model to the client, which infers a high-quality frame. Since the combined file size of the former is typically much lower than that of the latter, bandwidth usage can be reduced significantly. Several works have proposed the application of learning-based SR to static point clouds, considering a single frame only (*e.g.*, [135, 136]). However, research on dynamic video, in which multiple, dependent

point cloud or mesh frames are consumed, is scarce. This is partly because commodity devices are generally unable to handle the computational complexity of SR on the client side. Zhang *et al.* [137] recently published a study on possible optimizations for volumetric video with SR, among which *(i)* careful model trimming in the feature extraction stage by removing several layers of the generated network, *(ii)* the adoption of motion vectors for consecutive frames, avoiding the need to infer frames one by one, and *(iii)* viewport-adaptive inference, in which a high-quality version is only inferred for objects that are within the user's viewport. However, the highest frame rate achieved in this work – through adopting a subset of the proposed optimizations – is 13 FPS for a point cloud video consisting of merely 100 thousand points per frame. The application of SR thus remains a relevant and challenging topic of research.

Also worth mentioning is that Lee *et al.* [138] recently proposed a new framework for real-time point cloud streaming, which uses parallel encoding through so-called parallel decodable trees (pd-trees). By using independent representations, individual points can be decoded in parallel, significantly reducing the decoding latency. Other approaches for point cloud compression have also been proposed, but mainly focus on static imagery [139] or LiDAR-based systems for autonomous vehicles [140, 141]. Further details on these particular applications, which are out of the scope of this tutorial since they do not deal with dynamic video, can be found in recent surveys by Pereira *et al.* [142] and Cao *et al.* [143].

C. Compression of Meshes

Regarding meshes, two different applications for immersive video streaming can be considered. On the one hand, meshes can be used for rendering only: point clouds are used to capture, encode, and decode the content, while the rendered object consists of meshes corresponding to the considered points in space. This approach is used by Zerman *et al.* [23], who evaluate differences between rendering objects as a point cloud or as a collection of meshes. On the other hand, meshes can be used to represent and render three-dimensional objects. In this case, meshes are encoded and decoded directly.

Many influential works on mesh-based compression schemes exist [144, 145]. These older works, however, typically only consider the geometry of static objects, discarding texture information or the nature of dynamic structures. Recent work on this topic is limited, yet a few notable examples exist. Pavez and Chou introduce a so-called polygon cloud, a compressible representation of 3D geometry intermediate between meshes and point clouds [146]. Polygons correspond to triangles where the object's surface is smooth but can also be represented by lines or single points if needed. By applying an adapted point cloud compression scheme based on the region-adaptive hierarchical transform [147], the authors find that, compared to static polygon clouds, the compression ratio can be improved by a factor of 2 to 5.

More recently, Nasiri *et al.* [148] propose a framework to compress three-dimensional mesh textures using a so-called geometry-aware intra-coding algorithm. This algorithm considers the topology of the associated meshes, so redundancies in the texture map can be reduced. Most notable, however, is Google's Draco [149], a C++ compression library for point clouds and meshes. Having been designed for high compression speed, it can compress meshed point cloud video in real time on commodity hardware. Even though its compression ratio is limited to a factor of 10, its fast execution makes it a deserving candidate for inclusion in real-time 6DoF video streaming systems.

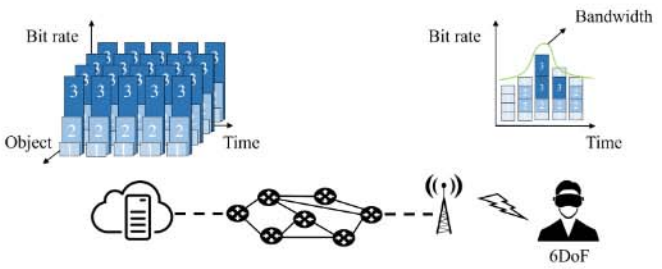

Fig. 22. The concept of HAS applied to multi-object volumetric video [150]. Each object is encoded at different bitrates and temporally, so that the client can change the quality of each object based on, *e.g.*, available bandwidth.

D. Transmission: Video on Demand

Several studies have examined the application of volumetric video streaming, where the video is streamed in an on-demand fashion. Several works propose to use solutions based on DASH, using a manifest that contains the metadata of considered point cloud objects [150, 151]. When a single point cloud or mesh object is considered, a similar strategy to traditional video can be adopted, in which the quality of this object is adjusted to match the bandwidth in the network. However, an additional spatial dimension is introduced when multiple tiles or objects are considered (see Figure 22). Similar to tile-based omnidirectional video (see Section IV-C), advanced rate adaptation is required to allocate the available network resources where they are needed, *i.e.*, to those objects that are in the current viewport of the user. To this end, the same three components can be used: *(i)* rate adaptation, *(ii)* viewport prediction, and *(iii)* saliency detection.

1) Rate Adaptation: Similar to omnidirectional video streaming, on-demand streaming of volumetric video requires the ability to deal with dynamic environments. Variable connectivity and bandwidth require adaptability, so rate adaptation is again an important factor. Initial works on rate adaptation for volumetric video consider single point cloud objects only. Hosseini and Timmerer [152] propose *DASH-PC*, a DASH-based solution for single point cloud streaming. Different quality representations are provided through point cloud sampling rather than advanced compression schemes. While this approach does not require additional coding (an advantage in the case of live and real-time video, discussed later), sampling offers limited compression. Furthermore, the approach assumes that the video quality can be adapted on a per-frame basis. Requesting frames one by one would result in excessive requests per second, which is not feasible.

Focusing on single point cloud objects, more advanced rate adaptation algorithms have been proposed that distinguish be-

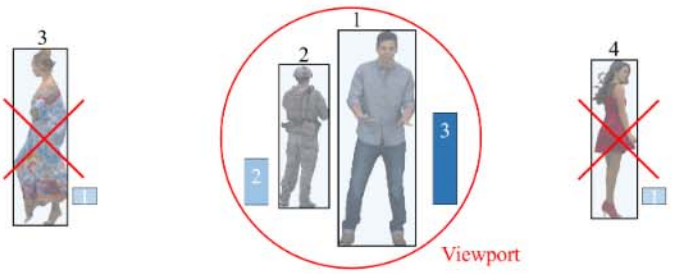

Fig. 23. Example quality decision-making for adaptive point cloud delivery. Objects are ranked according to their size in the user's viewport, and assigned a quality representation based on the amount of available bandwidth. In this case, the highest quality was assigned to the object closest to the user, while the lowest quality was assigned to objects outside of the viewport.

tween different spatial regions. For example, Park *et al.* [153] divide a point cloud object down to the level of voxels and propose a utility-based rate adaptation heuristic to determine how the available bandwidth should be allocated among them based on the user's viewport. It is worth noting that this approach does not use a traditional queue-based approach, in which temporal segments are retrieved one after the other; instead, a window-based approach is used, where voxels belonging to different segments within a given time window can be retrieved. The authors show that the proposed rate adaptation algorithm, combined with window-based buffering, significantly improves the visual quality of smaller details.

Other works followed suit, with Subramanyam *et al.* [154] proposing to divide point cloud objects into four different tiles, according to four virtual cameras placed around the object in the xz plane. Han *et al.* [155] propose a segmentation scheme that goes even further, uniformly dividing the point cloud using different sizes. A rate adaptation heuristic, which takes into account the visibility of each of the segments by considering *(i)* the viewport position, *(ii)* the occlusion of segments, and *(iii)* the distance to each of the segments and its impact on the visual perception. By ignoring points that should not be rendered, the effective bandwidth can be reduced by an average of 40% while still providing the same visual quality in terms of the structural similarity index measure (SSIM). A similar approach is used by Lee *et al.* [138], who apply advanced culling of occluded points and sampled points further away from the user, resulting in significant bandwidth savings for the same SSIM values.

Other works go one step further, considering more complex scenes consisting of multiple point cloud objects. Van der Hooft *et al.* [150] propose PCC-DASH, a DASH-based approach for delivering such scenes. In their work, objects are encoded using the V-PCC encoder, using a group of pictures (GOP) length of 30 - corresponding to one second of video - with different parameter settings to provide multiple quality representations. Several rate adaptation algorithms are proposed to allocate the available bandwidth among the considered objects, taking the user's viewport into account. To this end, point cloud objects are first ranked, using metrics such as the distance between the object and the current user's position (lower is better), the (potential) visible area of the object (higher is better), and the visible area divided by the bandwidth cost (higher is better), or a combination thereof. Then, in a second phase, the available bandwidth is allocated among these objects using three different schemes: *(i)* a greedy approach, in which the highest possible quality is given to the highest ranked point cloud object (taking the available bandwidth into account), before moving onto the next; *(ii)* a uniform approach, in which, starting with the highest ranked object, the quality of the different objects is increased one representation at a time; and *(iii)* a hybrid approach, in which first the quality of objects within the viewport is improved uniformly until either the highest quality is assigned, or no more bandwidth remains, and only then the quality of objects outside of the viewport is improved. An example of the latter approach is illustrated in Figure 23, with objects ranked according to the visual area of each object. The authors show that the best results are obtained when the available bandwidth is uniformly distributed among visible objects; when buffering is considered, this approach requires accurate prediction of the user's location and focus to identify these objects correctly. This aspect is considered next.

2) Viewport Prediction: ABR decision-making can benefit from adopting ML algorithms to deal with the high complexity of immersive media streaming. This is especially true since the client must also anticipate future movement: as different components, such as video coding and rendering, inevitably add to the end-to-end delay, it is essential to predict, as accurately as possible, where the user will be looking in the VR scene, based on their past and current location. This allows for proactively delivering only relevant content at any time, with increased video quality [156]. Recently, the first datasets containing user traces with 6DoF movement have been made available, which can be used by ML algorithms to learn from user behavior [157, 158].

In 6DoF scenarios, however, the complexity is significantly higher than in the case of 3DoF since the movement of the user's position needs to be accounted for as well. Recent work on this subject treats each of the 6DoFs independently, using linear regression to predict the user's movement based on previous coordinates [159, 155]. This approach is suboptimal, given the strong correlations typically found in human movements. Both ABR decision-making and viewport prediction can benefit from adopting more advanced ML algorithms to deal with the high complexity of immersive media streaming. Han *et al.* [155] take a first step in this direction, applying multilayer perceptron (MLP) with a single hidden layer of three neurons. However, the differences between linear regression and MLP in this work are statistically irrelevant. Further research is required to improve the state of the art of 6DoF viewport prediction.

3) Saliency Detection: Similar to omnidirectional video, saliency detection can be considered to prioritize spatial regions (*i.e.*, objects or tiles thereof). Although the topic has yet to receive significant attention in the context of volumetric video, a recent study by Li *et al.* incorporates saliency as part of a multi-tile ABR algorithm to improve the perceived quality of point cloud video [160]. Static saliency (*i.e.*, the saliency of a single frame) is detected based on the geometric

and textural features of the point cloud object. In contrast, dynamic saliency is extracted through motion estimation. Subsequently, the detected saliency is combined with the tiled video content in an attempt to provide fine-grained bitrate adaptation in a DASH-based setup. Simulation results for a 5G network scenario show that the proposed solution outperforms approaches that do not consider tiling.

E. Transmission: Live Video

Volumetric video can also be used in live scenarios, where the end-to-end delay is limited to the order of seconds. In live content capture, Jansen *et al.* [161] propose a pipeline for volumetric media-based video conferencing based on LL-DASH. The experimental setup consists of at most four RealSense cameras, which together result in point cloud frames consisting of 50 000 points, generated at a frequency of 10 Hz. A single-quality version of the point cloud streams is made available at a bitrate of approximately 10 Mb/s using the encoder proposed by Mekuria *et al.* [131]. DASH-based delivery is made possible through the GPAC toolset [162]. While the performance of the proposed system is not thoroughly evaluated, the authors conclude that the number of points per object contributes highly to the system's latency and achievable frame rate.

Another approach to reducing latency consists of the adoption of QUIC for the delivery of point cloud video. Ravuri *et al.* [91] propose a hybrid delivery scheme based on QUIC, in which point cloud objects within the user's viewport are sent reliably, while those not visible to the user are sent unreliably. Same as for tile-based omnidirectional video, results show that the perceived throughput and the startup delay can be improved significantly compared to a scenario in which all data is sent reliably. However, the authors acknowledge that their approach suffers from the fact that current compression techniques for volumetric video (*e.g.*, MPEG's reference encoder [22]) are based on HEVC, which is not robust against packet loss [92]. In the particular case of V-PCC, non-reliable packet delivery in lossy networks even results in a failure to decompress the delivered point cloud objects.

F. Transmission: Real-Time Video

Hu *et al.* [163] present a prototype for capturing point clouds with a single Azure Kinect camera. The depth and color images are decompressed and merged before being compressed by Google's Draco [149]. A TCP server sends data over a local WiFi network to a custom C# client in a Unity application through sockets. Three types of rendering procedures are evaluated: single-threaded CPU, multi-threaded CPU, and graphics processing unit (GPU). The authors show that, while GPU does outperform the other approaches, the latency per frame at 1080p is 180 ms and 80 ms on a mobile and a desktop device, respectively, resulting in frame rates lower than 13 FPS for a single point cloud object. The authors conclude that the latency and energy consumed in each stage of the capturing-to-rendering pipeline are proportional to the resolution of the point cloud, preventing the use of high-resolution point clouds on commodity devices.

Fig. 24. Setup by Dijkstra-Soudarissanane *et al.* [165]. Two users are immersed in a virtual world where they can both see and hear each other in real time.

Concerning the streaming of meshes, Orts-Escolano *et al.* [164] propose Holoportation, an end-to-end system for augmented and virtual reality telepresence. Eight pods capture RGBD streams, which are fused and transmitted to the user with limited compression. The setup requires approximately 2 Tb/s per user and dedicated hardware on the client side to render the data in real time. Dijkstra-Soudarissanane *et al.* [165] propose a multi-view system for real-time capture, transmission, and rendering of volumetric media. This system relies on a multi-point control unit (MCU) to shift processing from end devices onto a centralized server. Through a relevant demonstrator, the authors show that two end users can effectively communicate within an immersive world (see Figure 24). However, the visual quality is relatively low, *i.e.*, many artifacts can be observed due to a limited number of cameras in the setup.

G. Transmission: In-Network Optimizations

Some works address the fact that the real-timeliness of 6DoF systems is not yet feasible because the complexity of encoders and decoders clashes with the limitations of contemporary hardware [150]. Moreover, rendering complex scenes with several objects may be unfeasible on end devices due to complexity constraints and energy consumption. Similar to omnidirectional video streaming, cloud and edge processing can thus be seen as an enabler for such services, offloading computational tasks to the network. The main idea is that cloud or edge renders stereo views for volumetric objects on a plane orthogonal to the viewing direction of the viewer and on the particular position at which the object is present. These two-dimensional rasterized views can be encoded with standard video codecs. As illustrated in Figure 25, the end device can integrate the pre-rasterized media onto a plane, adjusting to the position and orientation changes of the user.

In this context, Qian *et al.* [166] propose Nebula, a volumetric video streaming system for commodity mobile devices. This system uses edge resources to transcode point cloud content into regular video, thereby alleviating the client from computational tasks and reducing the end-to-end delay. A similar approach is proposed by Gül *et al.* [159], who use the aforementioned linear regression to predict the user's movement and, based on the resulting prediction, generate traditional, two-dimensional video which can be rendered on the client side. Lee *et al.* [138] propose GROOT, a system for

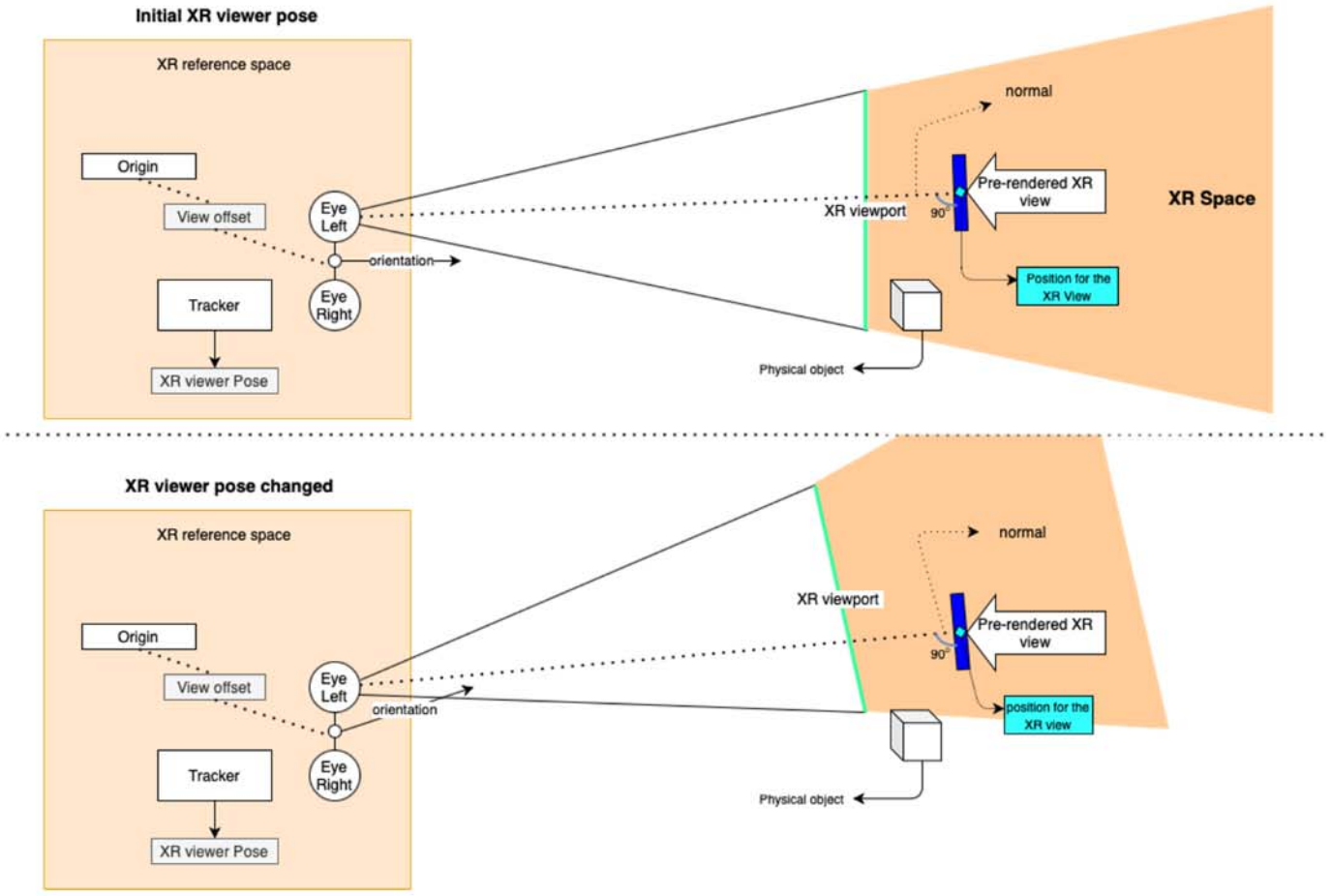

Fig. 25. Illustration of cloud- or edge-enabled immersive media streaming.

VoD delivery of high-quality volumetric media. While capturing and encoding are performed offline, advanced operations such as frustum culling, user-view adaptation, and real-time decoding through parallel optimizations make it possible to stream and render single objects from the 8i dataset [27] at 30 FPS.

H. Perception of Point Clouds

Compared to omnidirectional video, subjective quality evaluation for point clouds is still at an early stage, with standards for testing methods and procedures still to be agreed upon. Key parameters of point cloud quality evaluation study design mainly relate to the presentation (interactive vs. passive, single-stimulus vs. double-stimulus), viewing technology (2D/3D screen vs. VR HMD), and rendering scheme (*e.g.*, raw points vs. cubes/ellipsoids) [167]. Subjective point cloud quality studies have mainly featured static, non-animated point clouds, focusing on the source- and content-level influencing factors such as coding, compression, and geometry (see [168, 169, 170]). Their findings suggest that beyond the aforementioned influencing factors, stimulus presentation and model characteristics significantly affect quality rating results. In contrast, only a few recent quality evaluation studies have started to address delivery-related aspects of *dynamic* point clouds (*i.e.*, point cloud video) [36, 171, 172, 155, 173, 23].

In this respect, van der Hooft *et al.* [36, 171] have conducted subjective QoE lab studies based on a passive presentation protocol, where subjects assessed a number of video stimuli containing the generated viewport of a scene consisting of four point cloud objects featuring different programmatically defined movement paths, emulated network conditions and viewport prediction methods. Results confirm that bandwidth limitations and compression have a significant impact on point cloud video QoE and that better compression schemes for animated volumetric media are required. Comparison between single-stimulus (absolute category rating (ACR)) and double-stimulus (degradation category rating (DCR)) rating sessions show that despite similar scoring trends, ACR quality ratings were more negative than for DCR, confirming a non-negligible influence of stimulus presentation scheme on test results. Results also revealed subjects' fairly high expectations regarding the visual quality of rendered volumetric video due to traditional 2D video's role as a mental reference [171]. In a similarly designed non-interactive study, Cao *et al.* [172] quantified the QoE of mesh and point cloud content as a function of bitrate and observation distance. Results show that point cloud compression is preferred at low bitrates,

TABLE VII
DATASETS RELEVANT TO VOLUMETRIC VIDEO STREAMING.

Dataset	Year	Description
8i VFB [27]	2017	Four point cloud video sequences, each with a length of 300 frames or 10 seconds at 30 FPS. A total of 42 RGB cameras, grouped in fourteen clusters of three cameras, were used to generate the objects.
8i VSLF [176]	2017	One point cloud video sequence with a length of 300 frames or 10 seconds at 30 FPS, as well as six high-resolution static point clouds.
Owlii [177]	2017	Four point cloud video sequences with a length of 600 frames or 20 seconds each at 30 FPS.
Panoptic Studio 3D [178]	2017	54 point cloud video sequences with a total length of six hours. Ten synchronized RGB+D cameras have been used to capture the video.
CWIPC-SXR [179]	2021	45 point cloud video sequences with a length between 596 and 2768 frames at a frame rate of 30 FPS. Seven Kinect Azure DK devices have been used to capture the video.

while mesh compression scored higher at close observation distances.

In contrast, Subramanyam *et al.* [173] evaluated dynamic point clouds using interactive HMD-based setups to assess the impact of different compression types and compare them with omnidirectional video (3DoF) viewing. Their results reveal a slight subject preference for point cloud (6DoF) in the context of a quality rating task. They also found that personal preferences and model quality (realism, level of detail) have a pronounced impact on quality ratings, emphasizing the importance of new test datasets that offer a diverse range of models (see Table VII). For a more detailed discussion of the state of the art and a toolkit for subjective point cloud quality evaluation, please refer to [167, 170, 174].

I. Perception of Meshes

Several studies have evaluated the consumption and visual perception of meshes. Zerman *et al.* [23] conducted the first study on QoE for meshes, showing that meshes offer higher visual quality at larger bitrates than point clouds. At lower bitrates, however, point clouds outperform meshes. In more recent work [175], the same authors analyzed the user's behavior when consuming mesh-based augmented reality (AR) video, studying the distribution of the user's viewpoints and their location relative to the content. Their results show that users mostly watch the consumed meshes from a frontal view at an average distance of 2.37 times the object's height. While this conclusion opens up several opportunities for mesh-based video delivery (*e.g.*, changing the mesh resolution based on the expected user distance), it should be noted that only human objects were considered in this study and that participants used a smartphone (and thus, no HMD) to consume the content remotely.

Other work has focused on predicting the subjective user rating of meshes based on objective quality metrics. Cao *et al.* [172] evaluated perceptual quality utilizing ACR and DCR and used the results of their study to build a functional model for the prediction of the MOS. Results indicate that their model achieves Pearson correlation values between 0.964 and 0.972 between the observed and the estimated user ratings for different mesh-based objects. Nehmé *et al.* [184] proposed a full-reference metric for the quality assessment of 3D meshes. This metric uses statistics on curvature (*e.g.*, structure and contrast) and color (*e.g.*, chroma and hue values), thus integrating both geometry and color information. The proposed model was evaluated on a custom dataset of 480 animated meshes (made publicly available online) using five source models with geometry and color distortions. While individual features result in Pearson linear correlation coefficients (PLCCs) for the MOS between 0.30 and 0.70, the overall metric results in a PLCC between 0.86 and 0.91, depending on the considered content.

In more recent work, the same authors compared three subjective methods (*i.e.*, ACR with hidden reference (ACR-HR), the double-stimulus impairment scale (DSIS), and the subjective assessment methodology for video quality (SAMVIQ)) for the evaluation of an immersive experience in which the same five meshes are being used [185]. Evaluating the three approaches based on consistency among two groups of observers, the accuracy of the quality scores (*i.e.*, the agreement between reviewers), and the confidence intervals of the resulting scores, the authors showed that DSIS and SAMVIQ outperform ACR-HR in terms of accuracy, with DSIS achieving the highest accuracy in the shortest amount of time. While this work focuses less on the correlation between objective and subjective quality scores, it provides relevant insights into subjective experiments for volumetric video.

J. Datasets, Studies, and Surveys

An overview of relevant datasets for volumetric-video delivery is presented in Table VII. An overview of covered studies is presented in Table VIII, while relevant surveys are listed in Table IX.

VI. 6DoF Imagery Video

In this section, we will discuss image-based video streaming with 6DoF. We will begin by examining content capturing and preprocessing for light fields and holograms in Sections VI-A and VI-B. Sections VI-C and VI-D will delve into the compression of these representations. Following that, we will provide an overview of current state-of-the-art approaches for on-demand image-based video delivery in Section VI-E. We explore the effect of quality degradation on user perception in Section VI-F before discussing relevant datasets, studies, and surveys in Section VI-G.

A. Capturing of Light Fields

Image-based solutions render the view from a set of pre-acquired images, each captured at a different angle and tilt. Light fields, which describe the amount of light flowing in every direction through every point in space [186], are often

TABLE VIII
STUDIES RELEVANT TO VOLUMETRIC VIDEO STREAMING. IN SOME CASES, PROTOCOLS AND EVALUATIONS ARE NOT SPECIFIED (N/S).

	Study	Year	Target	Protocol	Evaluation	Focus
Point clouds	Jansen *et al.* [161]	2020	Live	DASH	Internet (N/S)	Preliminary architecture for low latency
	Hu *et al.* [163]	2021	Live	Sockets (TCP)	Local (WiFi)	Prototype for capturing and delivery
	Petrangeli *et al.* [151]	2019	Static	Generic	Internet (WiFi)	Multi-object quality selection
	Hosseini and Timmerer [152]	2018	VoD	DASH	Offline	Single-object per-frame rate adaptation
	Park *et al.* [153]	2019	VoD	DASH	Simulation (artificial)	Multi-tile per-segment rate adaptation
	Qian *et al.* [166]	2019	VoD	DASH	Internet (4G)	Preliminary end-to-end architecture
	van der Hooft *et al.* [150]	2019	VoD	DASH	Emulation (4G)	Multi-object per-segment rate adaptation
	Li *et al.* [160]	2022	VoD	DASH	Simulation (5G) Local (WiFi)	Multi-tile saliency-based rate adaptation
	Ravuri *et al.* [91]	2022	VoD	Custom (QUIC)	Emulation (5G)	Partially reliable video delivery
	Subramanyam *et al.* [154]	2020	VoD	Generic	Offline	Multi-tile per-segment rate adaptation
	Wang *et al.* [180]	2021	VoD	Generic	Offline	Multi-object/tile per-segment rate adaptation
	Lee *et al.* [138]	2020	VoD	Custom (TCP)	Local (WiFi)	Parallel decoding and rate adaptation
	Han *et al.* [155]	2020	VoD	Custom (TCP)	Local (WiFi) Emulation (4G) Internet (5G)	VP prediction and VP-aware optimizations
Meshes	Orts Escolano *et al.* [164]	2016	Live	N/S	Local (cable)	Prototype for capturing and delivery
	Dijkstra-Soudarissanane *et al.* [165]	2019	Live	WebRTC	Local (N/S)	Demonstrator for capturing and delivery
	Gul *et al.* [159, 156]	2020	VoD	WebRTC	Local (WiFi)	VP prediction and low-latency delivery

TABLE IX
SURVEYS RELEVANT TO VOLUMETRIC VIDEO STREAMING.

Authors	Year	Component	Description
Alkhalili *et al.* [181]	2020	Streaming	An overview of studies related to volumetric video streaming, focusing mainly on HAS and ABR algorithms. Three studies for volumetric video [152, 153, 166] and one study on omnidirectional video [182] are highlighted, discussing aspects such as bandwidth savings, robustness to network variations, and latency perceived by the user.
Cao *et al.* [130]	2019	Coding	An overview of PCC solutions, covering compression in one dimension (traversal compression), two dimensions (projection-based compression), and three dimensions (direct analysis). Four approaches are highlighted and evaluated in terms of performance (PSNR) and fidelity (lossless or lossy), including MPEG's V-PCC encoder.
Cao *et al.* [143]	2021	Coding	A survey on PCC solutions, covering both static and dynamic point cloud objects. Recent developments related to DL approaches are covered, and an evaluation of selected approaches is conducted. Coding complexity and execution times are not considered, however.
Dumic *et al.* [183]	2018	Quality perception	An overview of subjective evaluation protocols for both 2D and 3D video quality assessment. Several adaptations required for point cloud video are mentioned, along with a description of objective quality metrics (*e.g.*, based on PSNR).
Dumic and da Silva Cruz [167]	2020	Quality perception	Presents a summary of advancements related to PCC and point cloud quality assessment, along with a discussion of existing objective quality metrics. A general framework for subjective evaluations is presented, showcasing its reliability by performing the same evaluations at two different research facilities, with similar results.
Huang *et al.* [127]	2021	Capturing	A survey on point cloud registration techniques. State-of-the-art approaches are evaluated on four datasets, constructed through depth or LiDAR sensors. Evaluations are limited to static frames, rather than dynamic video.
Pereira *et al.* [142]	2020	Coding	A taxonomy for the organization of existing PCC solutions. A total of 94 works are covered in this survey, discussing, among others, the type of content (static or dynamic), the included components (*e.g.*, geometry and color), and the fidelity (lossy or lossless).

used in this case. The radiance along light rays in a 3D space with constant illumination can represent light fields. This can be described by the plenoptic function $L(x, y, z, \phi, \theta)$, which is parametrized by the coordinates x, y, and z, and the angles ϕ and θ (see Figure 26). Higher dimensions can be considered, taking into account time, wavelength, and polarization angle.

In free space, the radiance along a ray remains constant from point to point along its length. Thus, there is redundant information along one dimension, leaving a four-dimensional function that is referred to as the four-dimensional light field. It can be parametrized by the parameters u, v, s, and t, where u and v are the positions on the aperture or object plane and s and t are the positions on the image plane [187]. Thus, the function $L(u, v, s, t)$ can be considered as a collection of images on the st plane, observed from a position on the uv plane (see Figure 26). In other words, light field imagery captures both spatial and angular information. Accordingly, the uv plane refers to the angular domain, while the uv plane refers to the spatial domain (see Figure 27). At the time of writing, a light field video is considered a sequence of light field images captured at a constant rate.

A single camera device can be used to capture light field imagery from a limited range of angles [188]. These cameras multiplex the spatial and angular domains into a 2D image, known as a lenslet image. An example of such a device is the Lytro Illum camera. The multiplexed 2D image can then be converted into multi-view images. In more advanced scenarios, however, a camera array (*i.e.*, a setup in which multiple cameras are positioned on a grid) is often used [189, 190]. A Lytro ILLUM camera captures lenslet images at 3 FPS, making it suitable for capturing only static light fields. Wang *et al.* [191] developed a hybrid system using a 3 FPS Lytro ILLUM camera

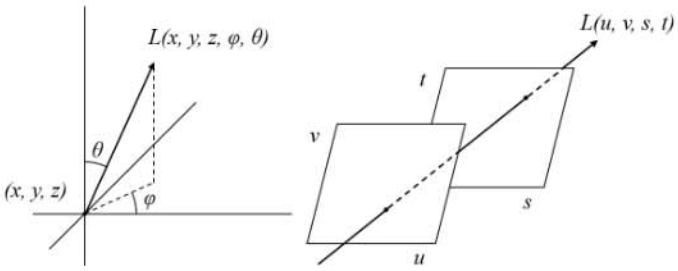

Fig. 26. Illustration of the five-dimensional (left) and the four-dimensional (right) plenoptic light field function L.

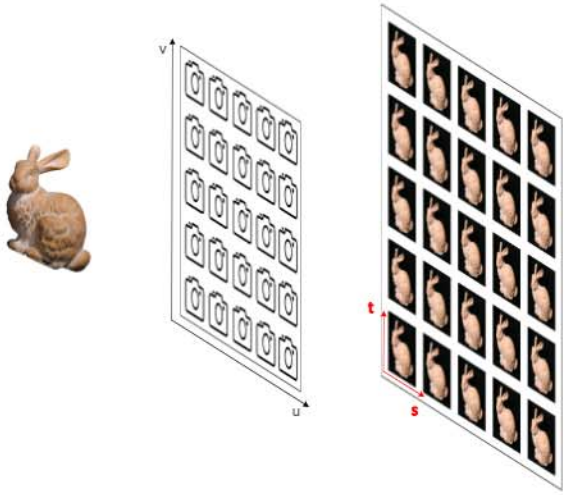

Fig. 27. Light field multi-view capturing. The *uv* plane refers to the angular domain, while the *st* plane refers to the spatial domain.

and a standard 3 FPS 2D video camera to generate a full light field video at 3 FPS. A recent example of a camera array light field camera is presented by Broxton *et al.* [28], who have built a hemisphere containing 96 cameras, each with a field of view of 120 by 90 degrees (see Figure 28), capturing at a rate of 3 FPS. In any setup, it is essential to ensure that differences between neighboring images are small enough; recent work considers an 0.3-degree difference in angle between images in order to provide a smooth transition [5].

Light fields offer two main advantages. First, capturing content does not require complex preprocessing tasks; it suffices to capture and store the different images, which are then ready for transmission. Second, displaying content based on the user's position and focus requires modest computational resources since the requisite images are readily available. However, image-based solutions suffer from ample storage and bandwidth requirements. Even when the content arrives on time, contemporary handheld devices cannot load the resources in real time. This is illustrated by Wijnants *et al.* [192], who show that a high number of GPUs and cache optimizations are required to stream a single, static object captured through light fields. Nevertheless, light fields are a promising approach for immersive video streaming.

Fig. 28. Illustration of the light field capture rig proposed by Broxton *et al.* [28].

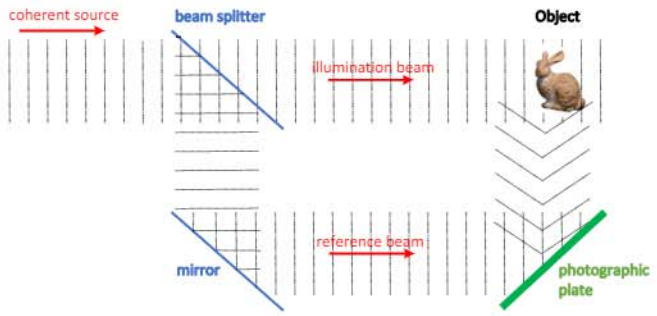

Fig. 29. Recording a hologram.

B. Capturing of Holograms

Holography is the ultimate solution for 3D image technology as it can record light fields scattered off objects and reproduce them later in the absence of original objects, similar to sound recording, whereby a sound field can be reproduced without the presence of the original sound generator. The invention of holography dates back to 1948, and it is attributed to Denis Gabor, who was awarded the Nobel Prize for this discovery [193]. However, its development was delayed due to the lack of coherent light sources until the invention of the laser in the early 1960s. To capture holograms, a coherent light beam is split into two beams. The first beam, *i.e.*, the object beam, is redirected toward the object(s), and the second beam, *i.e.*, the reference beam, is redirected toward the recording medium. The object beam is reflected off the object(s) and interferes with the reference beam on the recording medium, resulting in a recorded interference pattern. The same reference beam is used to reconstruct the object field along with all its properties, *i.e.*, light intensity, parallax, and depth. Figure 29 and Figure 30 show how a hologram is recorded and reconstructed, respectively. Recording the object beam relative to the reference beam enables recording the phase in addition to amplitude, allowing the reconstruction of 3D images.

Digital holography was enabled by the emergence of required sophisticated electronic devices, *i.e.*, charged-coupled device (CCD) image sensors, spatial light modulators (SLMs),

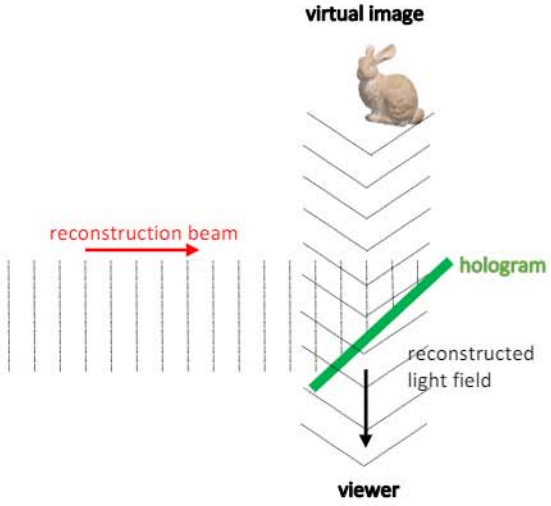

Fig. 30. Reconstructing a hologram.

and powerful computers [194]. The potential applications of digital holography include, but are not limited to, microscopy [195], interferometry [196], quantitative phase imaging (QPI) [197], surface measurements [198], 3D object recognition [199], and 3D display systems [200]. Despite significant progress in end-to-end holographic imaging, there are still challenges in the holographic signal processing chain, *i.e.*, from acquisition to display, namely holographic display, capturing, transform and coding, and quality evaluation [201].

Digital holograms can be recorded either by using an optical setup, which yields the optically generated hologram (OGH) or by using an electronic setup, which yields the computer-generated hologram (CGH). The OGHs typically come with three main challenges: *(i)* resolution limitations, *(ii)* physical limitations, and *(iii)* optical distortions [201]. At the same time, the CGHs using one of the following methods, *i.e.*, *(i)* point cloud method [202], *(ii)* polygon method [203], *(iii)* RGB+Depth method [204, 205], or *(iv)* ray-based method [206], suffer from expensive computation, and realistic scene rendering problems.

While commercial light field cameras are available, recording digital holograms at high resolutions requires specialized optical setups and expertise to build and operate [201]. However, the CGHs are much more compute-intensive than other classical image renderings.

C. Compression of Light Fields

Multiple compression techniques have recently been proposed to exploit a light field's spatial and angular redundancy to compress the large amounts of required data. Viola *et al.* [187] compare several compression techniques applied directly to lenslet images and after conversion to multi-view images. The authors show that the latter approach results in better performance, achieving both higher PSNR values and subjective user ratings. This is primarily due to the fact that, once all content has been processed, similarities between images can be used to compress data using existing video codecs such as HEVC efficiently. Light field image compression methods are mainly of three groups: *(i)* transform-based, *(ii)* traditional prediction-based, and *(iii)* learning-based.

1) Transform-Based Compression: In transform-based approaches, the redundancy in a 4D light field is exploited in a transform domain. These approaches typically involve using a transform, such as the discrete cosine transform (DCT) or the discrete wavelet transform (DWT), to convert the light field data into a different domain, where the redundancy is more pronounced. This allows for more efficient compression by exploiting the statistical properties of the transformed data. For example, the multidimensional light field encoder (MuLE) [207], which has been adopted by the joint pictures expert group (JPEG) Pleno standardization committee, initially converts partitions of the four-dimensional (4D) light field into 4D blocks. 4D DCT is computed for each block, and the transformed block is then encoded using an adaptive arithmetic coder. In addition to DCT, other transforms such as DWT [208], Karhunen-Loève Transform (KLT) [209], and graph Fourier transform (GFT) [210] are also used to transform light fields into a transform domain.

2) Prediction-Based Compression: Several studies have since proposed frame prediction schemes that include the different images in an optimized order, resulting in a pseudo-temporal video sequence that can then be compressed [211, 212]. Other works perform a second preprocessing step, creating a low-rank representation that aligns the light fields according to the disparity across views from one depth plane to the other [213]. The resulting representation is then again encoded using HEVC, allowing higher compression rates for the same visual quality. Ahmad *et al.* [214] interpret light fields as multi-view sequences and use a multi-view extension of HEVC (*i.e.*, MV-HEVC) for improved compression.

3) Learning-Based Compression: Learning-based view synthesis approaches are a recent development in light field compression. These approaches leverage the power of machine learning techniques to improve the encoding efficiency of light fields. One of the key advantages of these methods is that they can synthesize new views from a sparse set of input views, thereby reducing the amount of data that needs to be encoded. This can significantly improve the compression efficiency and the quality of the synthesized views. Hou *et al.* [215] first encode four corner views using HEVC. These views are then decoded and fed to a learning-based view synthesis method to reconstruct the remaining views. To improve the quality of synthesized views, the prediction residuals between the synthesized images and their corresponding original images are converted to a pseudo-temporal video sequence and encoded with HEVC. Jia *et al.* [216] first convert sparsely sampled light field image views to a pseudo-temporal video sequence and encode the generated pseudo-video with HEVC. A generative adversarial network (GAN) is then used to reconstruct unsampled image views. Amirpour *et al.* [217] present a light field compression method based on a video frame interpolation network. They formulate the task of view interpolation as frame interpolation in order to effectively compress light field images. The method involves classifying

image views into different layers and using the network to synthesize the views within each layer, utilizing image views encoded in previous layers as inputs.

However, efficient light field compression is still an issue and remains a relevant topic of research. In addition to the encoding efficiency, providing viewport scalability and random access to the viewports are among the critical challenges in light field compression [217].

In light field video compression, temporal redundancy is exploited in addition to spatial and angular redundancy. This is achieved using multi-view coding solutions, which leverage spatial, temporal, and inter-view predictions to compress light field videos. For example, Wang *et al.* [218] propose a new prediction structure that extends inter-view prediction into a two-directional parallel structure. Mehajabin *et al.* [219] propose a method that utilizes the similarity among views for prediction structure. Additionally, Wang *et al.* [220] propose a learning-based method that utilizes a CNN to synthesize views for improved compression efficiency.

D. Compression of Holograms

Compression of holographic data is a significant challenge in holographic signal processing, especially for dynamic hologram transmission. The large amount of data associated with holograms can lead to a bottleneck in terms of bandwidth requirements, with dynamic holography streaming potentially requiring as much as 14 Tb/s [201]).

The CGHs can be encoded prior to conversion to holograms considering their representations. For example, RGBD images are encoded and transmitted in [221]. On the decoder side, the bitstream is decoded, and the hologram is generated. Raw holograms can be encoded in either the hologram or object plane. To encode in the hologram plane, each hologram that consists of complex numbers is divided into *(i)* real and imaginary parts or *(ii)* amplitude and phase parts. They are treated as 2D images and then encoded using conventional image/video encoders. To encode in the object plane, holograms are first backpropagated to the object plane, typically by using numerical Fresnel diffraction or angular spectrum method, and then similar to the hologram plane, they are encoded by dividing into *(i)* real and imaginary parts, or *(ii)* amplitude and phase parts. Encoding holograms in the hologram and object planes has been compared in [222].

The JPEG Pleno Holography project aims to improve digital hologram compression. This includes *(i)* collecting test data for experiments, *(ii)* supporting hologram compression considering its complex data representation, and *(iii)* designing quality assessment procedures [223].

E. Transmission: Video on Demand

Recently, several studies have considered light fields and holography as a means to on-demand immersive video delivery with 6DoF. Similar to tile-based omnidirectional video and volumetric video, the presence of multiple views requires the client to make informed decisions on the quality at which these are retrieved. In this context, three components are considered: *(i)* rate adaptation, *(ii)* viewport prediction, and *(iii)* saliency detection. Recent studies on each of these components are discussed next.

1) Rate Adaptation: Wijnants *et al.* [192] propose a DASH-compliant framework for the interactive delivery of static light fields, focusing on single objects. In this approach, light field source images are encoded as segmented pseudo-videos with multiple quality representations, allowing for quality-variant access to specific portions of the light field. The framework can render still content in real time by leveraging video decoding to contemporary consumer-grade GPUs and using disk-versus-GPU caching in order to retrieve source images more quickly. Overbeck *et al.* [224] present a complete system for acquiring, processing, and rendering light fields using two light field camera rigs designed for portability. The authors also propose a compression scheme based on the VP9 codec, allowing high compression rates with real-time, random-access decompression. The renderer decodes the images on demand and reconstructs stereo views at a consistent rate of 90 Hz on commodity hardware. However, it should be noted that these works only consider static scenes and are, therefore, not suited for immersive video.

Considering dynamic scenes, Daniel *et al.* [225] are the first to propose an open streaming media standard for light field video. This standard allows compliant displays to consume a video stream consisting of three-dimensional frame descriptions and use these to render scenes without specialized HMDs. Several challenges to realize a functional framework are identified in this work, focusing on encoding, streaming, and rendering of light field video.

Considering the limitations of contemporary hardware, a novel end-to-end system for light field video streaming has recently been proposed by Broxton *et al.* [28]. Rather than using a multi-plane image scene representation, the authors use a collection of spherical shells to represent panoramic light field content. This approach leads to higher compression rates than traditional solutions, allowing the compressed representation to be rendered in real time on a consumer-grade gaming laptop.

Lievens *et al.* [226] design, implement, and evaluate the performance of a web-based static light field consumption system. This system adaptively streams static light fields over the network and then visualizes them in a vanilla web browser. The evaluations show that static light fields can be consumed in real time at AR/VR-compatible frame rates of 90 FPS or more on commercial off-the-shelf hardware.

El Rhammad *et al.* [227] propose a scalable progressive compression framework based on Gabor-wavelets decomposition for holography streaming. In this framework, the observer plane is divided into different blocks, and for each block, Gabor atoms are assigned, considering the duality between Gabor wavelets and diffracted light rays. Based on the importance of different blocks for reconstruction, each group of atoms is encoded in different packets. The packets are decoded progressively on the decoder side based on the viewer's position, and the corresponding hologram is reconstructed using a GPU implementation [227]. In [228], a progressive coding method that combines quality scalability with viewpoint scalability based on Gabor wavelets decomposition is proposed.

Amirpour *et al.* [229] propose a DASH-compliant view-aware adaptive streaming for holography streaming. Four different strategies for holography streaming have been studied. These strategies include *(i)* monolithic, *(ii)* single view, *(iii)* adaptive view, and *(iv)* non-real-time streaming. These four strategies were investigated and compared in terms of *(a)* bandwidth requirements, *(b)* encoding time-complexity, and *(c)* bitrate overhead. It remains to be investigated, however, how the proposed schemes can be applied in live streaming scenarios and to what extent they can reduce the delay in end-to-end 6DoF imagery video streaming frameworks.

2) Viewport Prediction: Viewport prediction has not yet been widely adopted or deployed in practical imagery video streaming. Despite this, some efforts are underway to provide random access to arbitrary views to avoid streaming the whole video and instead stream the desired viewport. Random access enables encoding and decoding a set of viewports rather than the entire set, though it may decrease compression efficiency. Amirpour *et al.* [230] propose a light field image compression method that allows for viewport scalability, quality scalability, spatial scalability, random access, and uniform quality distribution while maintaining high compression efficiency. The light fields are divided into sequential viewport layers, and each layer is encoded using the previous layer. Since a few references are used to encode each view, the proposed method improves the flexibility of light field streaming, provides random access to viewports, and increases error resilience. The experimental results demonstrate that accessing a desired viewport requires less than 5% of the whole bitstream. The method's error resiliency is demonstrated by the ability to reconstruct all viewports on the decoder side using deep neural networks, even with a limited number of streamed views. Avramelos *et al.* [231] demonstrate that using only the central view as a reference for inter-coding other viewports in light fields provides a desirable trade-off between random access to the desired viewport and compression efficiency.

3) Saliency Detection: To further reduce execution times, the application of saliency detection can be considered for 6DoF video streaming. Similar to 3DoF, predicting the user's consumption pattern could be used to optimize video streaming, decoding, and rendering. Some works have already performed saliency detection for light field video, using ML approaches to identify relevant parts of the video [232, 233]. Recent work by Wang *et al.* [234] considers user movement and gesture analysis to predict the future position of the user and consecutively encodes and transmits a limited set of views only. The proposed approach is evaluated on a limited 5×5 camera grid, showing that compression bitrates for the same visual quality can be reduced by 27% compared to the approach proposed in [218].

F. Rendering and Perception

A passive approach is typically used to evaluate the quality of degraded light fields subjectively. Light field multi-view images are converted into a pseudo-video and assessed by subjects. This way, the interactivity between content and subjects (which allows changing perspectives, refocusing, *etc.*) is ignored. Viola *et al.* [235] proposed an interactive setup to evaluate the quality of the light field and compared both interactive and passive methodologies in [236]. The statistical analysis showed that both methodologies are highly correlated; however, the interactive methodology leads to larger confidence intervals.

The reliability of conventional objective metrics, including full reference (FR) [237] and no reference (NR) image quality assessment (IQA) approaches, 3D IQA methods, and video metrics were assessed to evaluate light field images [238]. A FR light field quality metric is proposed in [239], which considers *(i)* global spatial quality based on view structure matching, *(ii)* local spatial quality based on near-edge mean square error, and *(iii)* angular quality based on multiview quality analysis. Another index is introduced by considering the angular-spatial characteristic of the light field based on focus stack [240]. Some NR light field image quality metrics were also designed [241].

Image-based rendering systems are classified into three types, namely *(i)* rendering without geometry, *(ii)* rendering with implicit geometry, and *(iii)* rendering with explicit geometry [242]. Light field rendering takes image rendering toward a "no-geometry-required" solution but uses multiple image views. A light field rendering method was proposed in [243], which generates new views without depth information by interpreting the input images as 2D slices of a 4D function. In this approach, the intersection points of the novel ray with the *uv* and *st* planes (see Figure 26) are first calculated. The nearest sampling rays in the light slab around the novel ray are then selected to interpolate the novel ray.

Evaluating the perceived visual quality of rendered holograms is one of the core challenges in holographic signal processing due to their inherent difference from photographic imagery. Ahar *et al.* [244] objectively and subjectively studied the impact of standard compression techniques on the numerically reconstructed holograms. The suitability of different displays, namely holographic, light field, and 2D displays, was studied, and it was shown that all displays show high correlations. The performance of standard codecs at different distances and perspectives was objectively evaluated in [245]. The quality of digital holography images encoded on the object plane was evaluated subjectively in [246], and the performance of objective metrics was assessed. Considering the 3D properties of digital holograms, the quality of compressed holograms was evaluated as a sequence of multiple views in [247], and the performance of objective metrics was assessed. Ahar *et al.* [248] conducted a dynamic subjective quality testing of holograms considering focus and viewing angle changes. Note that holographic signal processing is still in its early stages of development, and there are ongoing research activities to address its challenges [201].

G. Datasets, Studies, and Surveys

An overview of relevant datasets for 6DoF image-based video delivery is presented in Table X, while an overview of covered studies is presented in Table XI. Table XII provides a summary of important surveys and overviews that address 6DoF imagery video from various viewpoints.

TABLE X
DATASETS RELEVANT TO 6DOF IMAGERY-BASED STREAMING.

	Dataset	Year	Type	Description
Light fields	4DLFVD [249]	2021	Dynamic	The dataset contains a total of nine groups of light field videos taken by a 10 x 10 light field capture matrix composed of 100 cameras. The resolution of each camera is 1920 x 1056.
	EPFL [250]	2016	Static	The Ecole Polytechnique Fédérale de Lausanne (EPFL) dataset contains 118 light field images taken by Lytro Illum light field camera.
	HCI [251]	2017	Static	24 synthetic, densely sampled 4D light fields with highly accurate disparity ground truth.
	Light Field Intrinsic Dataset [252]	2018	Both	Real-world and synthetic light fields images and videos. The ground-truth intrinsic data comprises albedo, shading and specularity layers for all sub-aperture images. In case of synthetic data, ground-truth depth, normals and further decomposition of shading into direct and indirect components are also provided.
	MPI Light Field Archive [253]	2017	Static	Nine synthetic and five captured real-world scenes, with scenes spanning a large variety of conditions in terms of lighting. All light fields are of identical spatial and angular resolution (960 x 720 x 101).
	Raytrix (R8) [254]	2018	Dynamic	The R8 Raytrix dataset is composed of three video sequences recorded with a R8 Raytrix video camera fitted with a 35 mm lens.
	SINTEL [255]	2020	Dynamic	A medium-scale synthetic 4D light field video dataset consists of 24 synthetic 4D light field videos with 1204 x 436 pixels, 9 x 9 views, and 20–50 frames, and has ground-truth disparity values.
	SMART [256]	2016	Static	Sixteen light field images from both indoor and outdoor category. They cover general image content related features but also LF specific aspects.
Holography	B-com [257]	2016	Dynamic	Hologram computed from the multiview-plus-depth data or a synthetic scene.
	EmergIMG [222]	2018	Static	The sets of four phase-shifted holograms obtained by the phase shifting holography technique.
	Interfere-II [258]	2016	Static	The dataset consists of six diffuse and six specular holograms generated from 3D point clouds.
	Tensor Holography [205]	2021	Static	Computer-generated holography (MIT-CGH-4K) with 4000 pairs of RGB-D images and corresponding 3D hologram.

TABLE XI
STUDIES RELEVANT TO 6DOF IMAGERY-BASED VIDEO STREAMING.

	Study	Year	Target	Focus
Light fields	Wijnants *et al.* [192]	2018	Static	A standards-compliant architecture
	Overbeck *et al.* [224]	2018	Static	A system for acquiring, processing, and rendering
	Daniel *et al.* [225]	2018	Dynamic	An open streaming media standard for light field video of light field displays
	Broxton *et al.* [28]	2020	Dynamic	An end-to-end system for high quality immersive light field video streaming
	Lievens *et al.* [226]	2021	Static	A web-based static light field consumption system, enabling real-time consuming at AR/VR-compatible frame rates of 90 FPS
Holography	El Rhammad *et al.* [227]	2019	Static	Progressive streaming of digital holograms with a low latency using viewport scalability
	El Rhammad *et al.* [228]	2019	Static	Progressive streaming of digital holograms that combines quality and viewpoint scalability
	Amirpour *et al.* [229]	2020	static	A DASH-compliant view-aware adaptive streaming system for holography streaming

TABLE XII
SURVEYS RELEVANT TO 6DOF IMAGERY-BASED VIDEO STREAMING.

Authors	Year	Component	Description
Conti *et al.* [259]	2020	Coding	A comprehensive survey of light field coding solutions, focusing on angularly dense light fields. It includes special attention to a thorough description of the different light field coding methods and to the main concepts related to this relevant area. Additionally, comprehensive insights into open research challenges and future research directions for light field coding are presented.
Zhou *et al.* [260]	2021	Imaging	This survey reviews light field imaging from the following aspects: depth estimation, content editing, image quality, scene reconstruction and view synthesis, and industrial products.
Wu *et al.* [261]	2017	Imaging	A comprehensive overview and discussion of light field imaging over the past 20 years is presented. This overview focuses on all aspects of light field imaging, including basic light field representation and theory, acquisition, super-resolution, depth estimation, compression, editing, processing algorithms for light field display, and computer vision applications of light field data.
Sahin *et al.* [262]	2020	Synthesis	A comprehensive survey of methods for synthesis of computer-generated holograms is presented. They are classified into two broad categories: wavefront-based methods and ray-based methods. Their modern implementations in terms of the quality of reconstruction and computational efficiency are examined.
Haleem *et al.* [263]	2022	Application	An exploration of holography and its significant benefits through various development processes, features, and applications, where the focus is on 'holography for Industry 4.0'.
Blinder *et al.* [201]	2019	Imaging	An overview of the end-to-end chain from digital content acquisition to display, involving the efficient generation, representation, coding, and quality assessment of digital holograms is presented.

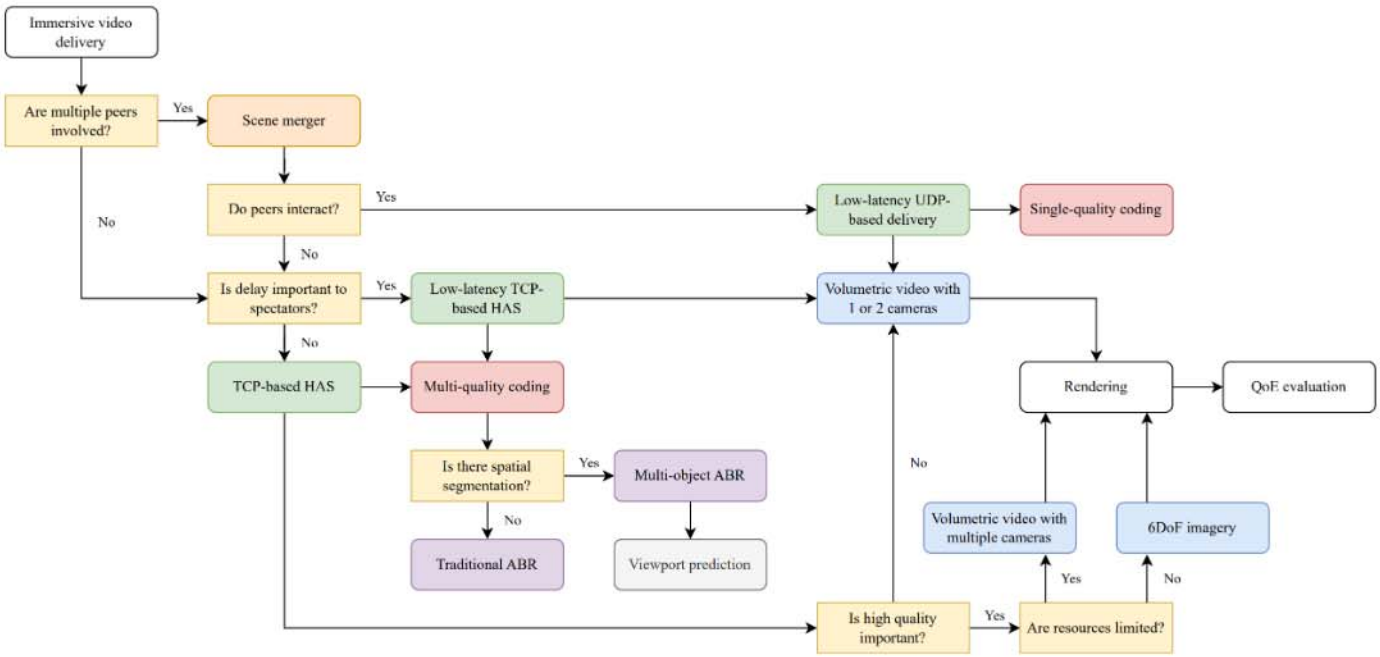

Fig. 31. Flowchart for technology and protocol selection for 6DoF video content delivery. Content representations are indicated in blue, coding approaches in red, streaming protocols in green, and ABR algorithms in purple.

VII. A Brief Guide to Immersive Video Deployments

This section presents a guide to immersive multimedia deployments, from which media researchers and practitioners can quickly understand what, where, and how they need to decide and do to deploy an end-to-end immersive multimedia system. Figure 31 presents a flow chart illustrating step-by-step decisions to provide a remote immersive experience, focusing on technology and protocol selection.

1) If **multiple peers** participate in the end-to-end communication, their respective captures (be it a human object or another type of content) must be merged on the server or client side. A **static background** and **a descriptor** (*e.g.*, a modified MPD [43]) are to be used to merge the content.
2) If **multiple peers interact**, a high-throughput and low-latency network infrastructure is required. Regarding delivery, **UDP-based** solutions such as WebRTC or unreliable QUIC are recommended to avoid packet retransmissions in lossy networks. Moreover, **Single-quality coding** will aid in getting the content ready as soon as possible.
3) If **peers do not interact**, or only a single peer is present, but the **delay is still essential** to spectators (e.g., when commenting on live events), **low-latency TCP-based** approaches, such as DASH with CMAF, could be used.
4) If **delay is crucial**, lightweight capturing and processing are required to achieve limited delays. Thus, using volumetric video with at most two depth cameras is recommended. Culling and sampling of points can limit the amount of data to process. Moreover, real-time compression requires **dedicated hardware** and parameter optimizations.
5) If the **delay is insignificant** (*i.e.*, in VoD scenarios), **TCP-based HAS** can be used. **Multiple quality representations** must be provided. Parameter quantization for content compression can be considered to create these different representations (*e.g.*, V-PCC [22]).
6) **Spatial segmentation** (*e.g.*, voxels for point clouds) can result in better usage of the available bandwidth. This approach requires **multi-object ABR** algorithms [43, 153], which are of higher complexity than traditional ABR algorithms and require accurate viewport prediction.
7) So far, only a few works on **6DoF viewport prediction** exist. These works consider relatively straightforward approaches, treating each of the 6DoFs independently [159, 155]. While these approaches can improve the client's quality decision-making, they should be used with caution.
8) When latency is of no concern and the **visual quality** is of the utmost importance, two approaches for content capturing can be adopted. If **resources are limited**, **sparse volumetric video** representations with multiple cameras can be considered. This comes with limitations related to the visual quality (*e.g.*, blur and other artifacts). **Otherwise**, **6DoF imagery-based** solutions can be adopted, requiring advanced camera rigs for dynamic video and significant processing times [28]).
9) Despite the advances in device capabilities, **rendering** complex scenes with several objects is still **unfeasible** on end devices due to complexity constraints and energy consumption. **Cloud and edge processing** can enable such services, offloading computational tasks to the network.
10) Finally, the **user's perception and QoE** can be assessed in terms of **objective metrics** (such as video quality metrics like VMAF or SSIM) and **subjective studies (i.e., questionnaires)**. Moreover, **cybersickness** and related discomfort are critical phenomena that can severely impair a person's immersive media experience, potentially causing abrupt termination of a viewing session or even preventing the adoption of immersive media.

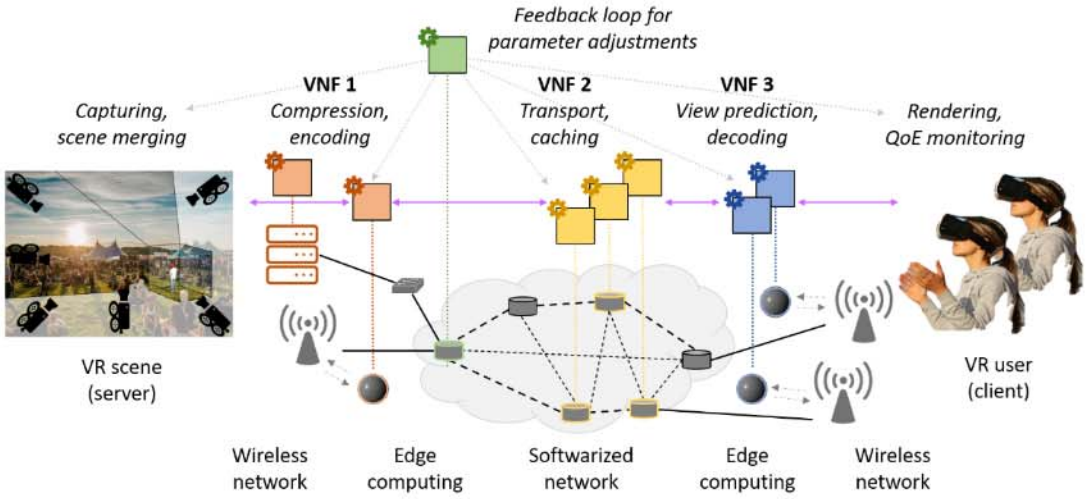

Fig. 32. Future immersive media architecture.

VIII. Outlook and Open Challenges

As discussed in Section VII, limitations in terms of throughput, latency, and packet loss currently impede the adoption of over-the-top video streaming solutions at scale. However, new opportunities arise with the deployment of 5G networks and the advent of 6G. In Figure 32, we propose an updated version of the initial architecture presented in Section III. This architecture is based on NFV, where virtual network functions (VNFs) are all composed of dedicated tasks (*e.g.*, caching, viewport prediction, or decoding) handled in the network delivery chain. This requires advancements to programmable networking and poses challenges for the different components involved. Below, we identify five major challenges for immersive video streaming, providing an outlook on future research.

A. Low-Latency Content Delivery

System settings are often rigidly and statically defined at design time. However, the user's perception of 6DoF video heavily depends on the ever-changing context. A fixed setting cannot handle different contexts and requirements: configuration settings should be dynamically tuned to the considered application, network characteristics, and user's prerequisites. Initial works on 6DoF video streaming have considered this aspect by making content available at multiple quality representations, similar to traditional video. However, more advanced approaches are needed to cope with the stringent interactivity and latency requirements of 6DoF video applications.

One way forward involves incorporating measurements from both the client-side application and the server. For instance, when measurements indicate a limited bandwidth to the client, encoding and transcoding parameters could be tuned on the fly, providing higher compression of the video content or even discarding irrelevant content as part of a culling process. As more users join the video session, the amount of generated quality representations could also be increased to improve the granularity of ABR decisions. Providing feedback from multiple clients, encoder settings can be tuned to clusters of users, adapting encoder profiles and caching strategies to meet aggregated needs.

More than these changes alone will be required to deliver immersive video at low latency, *e.g.*, in the context of live video or video conferencing. Although low-latency video streaming protocols such as LL-DASH and WebRTC are now commonly used for traditional 2D video, limitations in terms of processing power, network throughput, and latency impede their adoption to immersive video. As a result, only a few studies have attempted to provide real-time content capture, streaming, and rendering at high video quality. Recently, the Internet Engineering Task Force (IETF) initiated the media over QUIC (MOQ) working group aiming to develop a simple low-latency media delivery solution addressing use cases ranging from live video streaming over gaming to media conferencing at scale [264]. However, at the time of writing this paper only early drafts of MOQ are available without considering immersive media use cases [265]. With the ongoing deployment of 5G infrastructures and the advent of 6G, more ground-breaking systems and studies on these topics are expected.

B. Improved In-Network Solutions

Current transport mechanisms require computations to be executed at the server or client side, resulting in many end-to-end exchanges. Consecutively, any short-term changes in network characteristics or user interactions cannot be responded to in a timely manner, causing inadequate content or quality selection. Therefore, future research should incorporate complementary in-network optimizations to meet the requirements of even more stringent remote 6DoF experiences.

First, dynamically adjusting network configurations is possible through SDN, which separates the data layer from the control layer. Routing packets softwarematically allows for intelligent decision-making in the network, including bandwidth shaping, packet prioritization, rerouting, and caching. In the context of immersive video streaming, SDN can be adopted to meet stringent requirements in terms of bandwidth and latency.

Segment routing can play an important role here: this form of networking allows to prepend a header to packets that contain a list of instructions (*e.g.*, on forwarding packets to a specific destination), called a segment [266] (not to be confused with a DASH segment). Together, segments can create dynamic and unconstrained network paths, making segment routing highly responsive to network changes. This is beneficial for immersive media since guarantees on the network latency, throughput, and jitter can be given. Thus far, research has yet to consider the concept as an enabler of immersive video delivery.

Second, some works already consider the availability of edge and cloud resources to improve the decoding and rendering of immersive video content [159, 166]. Using these resources, however, requires smart orchestration of network and computing resources on the network path, efficiently allocating network resources to VNFs along the delivery path (see Figure 32). Current research considers optimization techniques such as integer linear programming (ILP) for optimal placement, where objectives aim to minimize the total cost or end-to-end delay [267]. The problem complexity grows exponentially with the number of services and network resources, hampering scalability. Furthermore, a static problem is typically considered, where the service requirements do not change over time – an unrealistic assumption in the case of 6DoF video delivery. Given these limitations, there is a strong need for advanced ML algorithms for dynamic and intelligent VNF placement. Such algorithms should be able to allocate and modify resources to different VNFs based on their computational or/and traffic load so that the system can adapt to the end users' needs. Given the large number of network and encoding parameters, we envision the application of DL techniques, such as multilayer neural networks trained with reinforcement learning (RL), to surge in the next few years.

Finally, the actual processing of immersive media content, including – but not limited to – transcoding, transrating, transmuxing, *etc.*, is becoming a vital option thanks to more computational resources being deployed on the edge, resulting in a cloud computing continuum that needs to be orchestrated and utilized efficiently.

C. Scalable and Portable Capturing Devices

As discussed in this tutorial, professional studios can capture volumetric videos with professional equipment using camera arrays, some of them relying on 3D sensors or multiview cameras. This typically requires that cameras are properly calibrated, lighting is properly set, and potentially adapting to changes (*e.g.*, objects moving), which might not be fully automated. The vast amount of data captured with these camera arrays to ensure high-quality volumetric videos poses a challenge for real-time processing, *e.g.*, for live events. In particular, when such systems generate point clouds or meshes, temporal consistency of their topology is desired, *i.e.*, the number of points and faces does not change continuously over time. In addition, some applications require meshes to be rigged so they can be transformed, such as when animated, which requires a complex fitting process [268]. Fast algorithms are required for this purpose and are currently being studied.

Additionally, applications such as volumetric video streaming require more affordable, portable, and fully automated solutions with consumer-grade capturing systems. It is expected that in the near future, research will be carried out in this field so that it is not always required to rely on professional studios.

D. Increased Compression Performance

Higher coding efficiency is crucial to enable services for immersive video. Currently, video codecs are widely used to compress immersive media, for instance, for omnidirectional videos or even for 6DoF with V-PCC. With the increasing performance of video codecs, immersive media can be compressed with a higher compression ratio. However, tools specifically considering immersive media when developing new video codecs could further improve their performance in immersive media compression. The new video coding standard, *i.e.*, versatile video coding (VVC) [269], supports higher versatility, including immersive media applications. While VVC supports the compression of omnidirectional videos, the versatility can be extended to support other types of immersive media. At the same time, existing codecs are suboptimal for some immersive media types, such as holographic media types, due to their substantial difference from natural images. Therefore, new *transforms* are required to adopt video codecs to different content types. Furthermore, video-based coding standards may only sometimes be the best fit for specific applications. For instance, sparse point clouds benefit from geometry-based coding, namely geometry-based PCC (G-PCC) [22], rather than a video-based coding solution. However, currently, the G-PCC standard only supports intra prediction, while temporal prediction tools are being investigated and will undoubtedly be added in the near future. In addition, meshes are widely used to represent immersive content instead of point clouds. Although technologies and some standards exist to compress such a format, time-varying attribute maps and connectivity information have yet to be considered or full covered. MPEG has issued a call for proposals for a new mesh compression standard to directly handle dynamic meshes with time-varying connectivity information and time-varying attribute maps.

Due to the bulky nature of the immersive media on the one hand, and the need for personalized and adaptive streaming of the immersive media on the other, more support for scalability and random access compression of immersive media is required.

Nowadays, DL-based image/video compression methods can achieve comparable or even better performance than traditional coding solutions [270]. In immersive video coding, DL can be used for *(i)* end-to-end compression or *(ii)* improving existing coding solutions.

An overview of design criteria and outlook on emerging media compression standards is given in [271]. In particular, optimization of existing coding tools and end-to-end deep neural network-based coding [272] are currently subject to research and standardization (*e.g.*, within MPEG). Furthermore, it suggests a better understanding of the human visual system (HVS) and the usage of perceptual coding tools. Finally,

the MPEG immersive video coding standard [273] seems to become an alternative to other approaches introduced in this paper, yet to be integrated and validated in end-to-end systems.

E. Evaluation of the User's Perception

A general problem in the domain of perceptual quality assessment is that while methods relying on explicit user feedback (*i.e.*, questionnaires, prompts, *etc.*) are well established, they are also known to suffer from individual biases (*e.g.*, personal preferences, scale usage) and from the intrusive nature of questions and prompts in terms of disrupting a person's experience. The latter is particularly critical in evaluating immersive media experiences since breaks in presence and immersion tend to significantly alter one's experience and subjective evaluation [274]. For this reason, alternative assessment methods that do not require conscious introspection must be investigated and integrated into immersive media experience evaluation. Such less intrusive alternatives include behavioral (based on observing and tracking user behaviors) as well as psycho-physiological (based on electroencephalogram (EEG), electrocardiogram (ECG), eye-tracking, *etc.*) assessment. Due to their complementary strengths and weaknesses, combining these three methodological strands has the potential to generate novel multi-method approaches that capture the experience and perception of immersive media at higher levels of accuracy and validity [33].

Another assessment challenge relates to the highly interactive nature of immersive video. In this medium, viewers are expected to freely move around within the scene or at least be able to change gaze direction. This genuine freedom to choose one's individual path through the media experience challenges subjective evaluation design because it requires trading off external validity and realism (as typically enabled by interactive test protocols) against reliability and reproducibility (as enabled by passive evaluation protocols, *e.g.*, using pre-rendered viewport trajectories). To resolve this trade-off, new subjective assessment methods are required that utilize novel approaches for analyzing and clustering user attention and behavior (see [275, 276]).

Finally, with the increasing quality and fidelity of immersive media delivery, the research community needs to converge regarding the design of a critical subjective assessment benchmark: analog to the Turing test known from the artificial intelligence (AI) domain, the purpose of this benchmark would be to determine a system's ability to provide an immersive experience that is indistinguishable from reality. We envisage that this ultimate benchmark draws from research on reality perception, judgment, and presence and integrates existing proposals in related domains such as computer graphics [277] and VR [278] in order to determine whether and when immersive media technology has fully delivered on its promises.

ACKNOWLEDGMENT

Jeroen van der Hooft is funded by the Research Foundation Flanders (FWO), with grant number 1281021N. The financial support of the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology, and Development, and the Christian Doppler Research Association is gratefully acknowledged. Christian Doppler Laboratory ATHENA: https://athena.itec.aau.at. The authors would like to thank the IDLab-DSLab research group for the use of their Insta360 Pro 2 camera.

ACRONYMS

2D two-dimensional. 1, 5, 6, 9, 14, 15, 18, 22, 24, 25, 27, 28, 31
3D three-dimensional. 1–4, 9, 17–19, 22–26, 28, 29, 32
3DoF three degrees of freedom. 1, 3, 5, 8, 11, 20, 23, 28
4D four-dimensional. 26, 28, 29
6DoF six degrees of freedom. 1, 3–6, 8, 16, 19–21, 23, 27–32

ABR adaptive bitrate. 11, 12, 20, 24, 30, 31
ACR absolute category rating. 22, 23
ACR-HR ACR with hidden reference. 23
AI artificial intelligence. 33
AR augmented reality. 23, 27
AVC advanced video coding. 14

BD Bjontegaard delta. 13

CCD charged-coupled device. 25
CDN content delivery network. 7, 11, 13, 14
CGH computer-generated hologram. 26, 27
CGI computer-generated imagery. 2
CMAF common media application format. 7, 14, 30
CMP cubemap projection. 10
CPU central processing unit. 4, 13, 21
CTE chunked transfer encoding. 7

DASH dynamic adaptive streaming over HTTP. 7, 11, 12, 14, 16, 19–21, 24, 27–30, 32
DCR degradation category rating. 22, 23
DCT discrete cosine transform. 26
DL deep learning. 17, 24, 32
DNN deep neural network. 18
DSIS double-stimulus impairment scale. 23
DWT discrete wavelet transform. 26

ECG electrocardiogram. 33
EEG electroencephalogram. 33
ERP equirectangular projection. 9, 10

FoV field of view. 4
FPS frames per second. 3
FR full reference. 28

G-PCC geometry-based PCC. 32
GAN generative adversarial network. 26
GFT graph Fourier transform. 26
GOP group of pictures. 20
GPU graphics processing unit. 21, 25, 27

HAS HTTP adaptive streaming. 6, 7, 11, 13, 19, 24, 30
HEVC high-efficiency video coding. 10, 14, 16, 21, 26
HLS HTTP live streaming. 7, 14

HMD head-mounted display. 1, 2, 8, 14–16, 22, 23, 27
HOL head-of-line. 8
HTTP hypertext transfer protocol. 6, 7, 14, 16
HVS human visual system. 32

ICP iterative closest point. 17
IETF Internet Engineering Task Force. 31
ILP integer linear programming. 32
IMU inertial measurement unit. 2
IoT Internet of things. 8
IQA image quality assessment. 28

JPEG joint pictures expert group. 26, 27

KLT Karhunen-Loève Transform. 26

LiDAR light detection and ranging. 17, 19, 24
LL-DASH low-latency DASH. 7, 14, 21, 31
LL-HLS low-latency HLS. 7
LRM linear regression model. 12

MCU multi-point control unit. 21
MEC multi-access edge computing. 14
ML machine learning. 16, 17, 20, 28, 32
MLP multilayer perceptron. 20
MOQ media over QUIC. 31
MOS mean opinion score. 8, 23
MPD media presentation description. 7, 13, 30
MPEG media pictures expert group. 7, 11, 18, 21, 24, 32, 33
MSS Microsoft smooth streaming. 7
MuLE multidimensional light field encoder. 26

NAT network address translation. 6
NFV network function virtualization. 8, 31
NR no reference. 28

OGH optically generated hologram. 26

PCC point cloud compression. 18, 20, 24
PCL point cloud library. 17
PLCC Pearson linear correlation coefficient. 23
PSNR peak signal-to-noise ratio. 15, 18, 24, 26

QoE quality of experience. 4, 8, 9, 14–16, 22, 23, 30
QP quantization parameter. 15
QPI quantitative phase imaging. 26

RAM random-access memory. 13
RAP random-access point. 5, 13
RL reinforcement learning. 32
RNN recurrent neural network. 14
ROI region of interest. 6
RTMP real-time messaging protocol. 7, 13
RTP real-time transport protocol. 8
RTT round-trip time. 8, 16

S-PSNR sphere-based PSNR. 15
SAMVIQ subjective assessment methodology for video quality. 23
SDN software-defined networking. 8, 31
SFC service function chain. 8
SIDR shifted instantaneous decode refresh. 13
SLM spatial light modulator. 25
SR super resolution. 18, 19
SSIM structural similarity index measure. 20
SSQ simulator sickness questionnaire. 9, 15, 16

TCP transmission control protocol. 6–8, 21, 24, 30
TLS transport layer security. 8

UDP user datagram protocol. 8, 14, 30
URAP unequal RAP. 13
URL uniform resource locator. 7

V-PCC video-based PCC. 18, 20, 21, 24, 30, 32
VNF virtual network function. 31, 32
VoD video on demand. 1, 6, 8, 9, 11, 13, 16, 22, 24, 30
VR virtual reality. 2, 8, 9, 11, 14, 16, 17, 20, 22, 27, 33
VRET VR exposure therapy. 2
VVC versatile video coding. 32

WebRTC web real-time communication. 8, 24, 30, 31
WLRM weighted LRM. 12

REFERENCES

[1] Sandvine, "Global Internet Phenomena Report 2023," https://www.sandvine.com/phenomena, accessed: 2023-02-09.

[2] S. Minns *et al.*, "Immersive 3D Exposure-Based Treatment for Spider Fear: A Randomized Controlled Trial," *Journal of Anxiety Disorders*, vol. 61, 2019.

[3] B. M. Kyaw *et al.*, "Virtual Reality for Health Professions Education: Systematic Review and Meta-Analysis by the Digital Health Education Collaboration," *Journal of Medical Internet Research*, vol. 21, no. 1, 2019.

[4] P. Kaliraj and T. Devi, *Innovating with Augmented Reality: Applications in Education and Industry*. CRC Press, 2022.

[5] A. Clemm *et al.*, "Toward Truly Immersive Holographic-Type Communication: Challenges and Solutions," *IEEE Communications Magazine*, vol. 58, no. 1, 2020.

[6] "Cambridge Dictionary - Immersion," https://dictionary.cambridge.org/dictionary/english/immersion, accessed: 2023-01-16.

[7] "ITI VR Crane & Equipment Virtual Reality Simulation Training," https://www.iti.com/vr, accessed: 2022-06-21.

[8] "Training to Craning in 60 Minutes: Putting My VR-learned Skills to the Test with a Real 22 Ton Crane," https://www.roadtovr.com/iti-vr-crane-training-simulator-test/, accessed: 2022-06-21.

[9] C. Jacobs, *Interactive Panoramas: Techniques For Digital Panoramic Photography, Volume 1*. Springer, 2004.

[10] "A Brief History of Panoramic Photography," https://www.loc.gov/collections/panoramic-photographs/articles-and-essays/a-brief-history-of-panoramic-photography, accessed: 2022-06-22.

[11] "Vintage 360 - Doo Interactive Offices in France - 2003," https://www.youtube.com/watch?v=EBYvU3hE4M4, accessed: 2023-02-09.
[12] "Ricoh Theta m15," https://theta360.com/en/about/theta/m15.html, accessed: 2022-06-21.
[13] M. E. *et al.*, "Virtual Reality-Based Interventions for Patients With Paranoia: A Systematic Review," *Psychiatry Research*, vol. 307, 2022.
[14] "Boosted by Virtual Reality and AI, Telesurgery Is on the Rise," https://www.healthcareitnews.com/news/boosted-virtual-reality-and-ai-telesurgery-rise, accessed: 2022-06-21.
[15] "YouTube," https://www.youtube.com, accessed: 2023-09-02.
[16] "Facebook," https://www.facebook.com, accessed: 2023-02-09.
[17] "Introducing 360 Video on Facebook," https://about.fb.com/news/2015/09/introducing-360-video-on-facebook/, accessed: 2023-02-09.
[18] "XSplit: Live Streaming & Recording Software," https://www.xsplit.com, accessed: 2023-02-09.
[19] "Introducing the Live 360 Ready Program and New Features for Live 360," https://www.facebook.com/formedia/blog/introducing-the-live-360-ready-program-and-new-features-for-live-360, accessed: 2023-02-09.
[20] "Encoder Settings for Live 360 Degree Videos," https://support.google.com/youtube/answer/6396222, accessed: 2023-02-09.
[21] "ABBA Announces Hologram Concert For Spring 2022," https://www.xliveglobal.com/fan-experience/abba-announces-hologram-concert-spring-2022, accessed: 2022-06-21.
[22] S. Schwarz *et al.*, "Emerging MPEG Standards for Point Cloud Compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, 2018.
[23] E. Zerman *et al.*, "Textured Mesh vs Coloured Point Cloud: a Subjective Study for Volumetric Video Compression," in *International Conference on Quality of Multimedia Experience*, 2020.
[24] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," in *Conference on Computer Graphics and Interactive Techniques*, 1994.
[25] "Kinect for Windows," https://developer.microsoft.com/en-us/windows/kinect/, accessed: 2023-02-09.
[26] M. R. Desselle *et al.*, "Augmented and Virtual Reality in Surgery," *Computing in Science & Engineering*, vol. 22, no. 3, 2020.
[27] E. d'Eon *et al.*, "8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, 2017.
[28] M. Broxton *et al.*, "Immersive Light Field Video with a Layered Mesh Representation," *ACM Transaction on Graphics*, vol. 39, no. 4, 2020.
[29] M. Torres Vega *et al.*, "Immersive Interconnected Virtual and Augmented Reality: A 5G and IoT Perspective," *Journal of Network and Systems Management*, 2020.
[30] E. Bastuğ *et al.*, "Toward Interconnected Virtual Reality: Opportunities, Challenges, and Enablers," *IEEE Communications Magazine*, vol. 55, no. 6, 2017.
[31] "3GPP TR 26.918: "Virtual Reality (VR) Media Services Over 3GPP"," https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3053, accessed: 2022-09-05.
[32] K. Brunnström *et al.*, "QUALINET White Paper on Definitions of Quality of Experience," 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00977812
[33] A. Perkis *et al.*, "QUALINET White Paper on Definitions of Immersive Media Experience (IMEx)," *CoRR*, vol. abs/2007.07032, 2020.
[34] O. Schreer *et al.*, "Capture and 3D Video Processing of Volumetric Video," in *IEEE International Conference on Image Processing*, 2019.
[35] Wowza, "2021 Video Streaming Latency Report," https://www.wowza.com/blog/2021-video-streaming-latency-report, Wowza, Tech. Rep., 2021.
[36] J. van der Hooft *et al.*, "From Capturing to Rendering: Volumetric Media Delivery With Six Degrees of Freedom," *IEEE Communications Magazine*, vol. 58, no. 10, 2020.
[37] "Keyframes, InterFrame & Video Compression," https://blog.video.ibm.com/streaming-video-tips/keyframes-interframe-video-compression, accessed: 2022-06-22.
[38] M. W. Akhtar *et al.*, "The Shift to 6G Communications: Vision and Requirements," *Human-Centric Computing and Information Sciences*, vol. 10, no. 53, 2020.
[39] H. Riiser *et al.*, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," in *ACM Multimedia Systems Conference*, 2013.
[40] J. van der Hooft *et al.*, "HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks," *IEEE Communications Letters*, vol. 20, no. 11, 2016.
[41] A. Narayanan *et al.*, "A Variegated Look at 5G in the Wild: Performance, Power, and QoE Implications," in *ACM SIGCOMM Conference*, 2021.
[42] A. Aijaz *et al.*, "Realizing the Tactile Internet: Haptic Communications over Next Generation 5G Cellular Networks," *IEEE Wireless Communications*, vol. 24, no. 2, 2017.
[43] J. van der Hooft, "Low-Latency Delivery of Adaptive Video Streaming Services," Ph.D. dissertation, Ghent University, 2019.
[44] A. Bentaleb *et al.*, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, 2019.
[45] "Microsoft Smooth Streaming," https://www.microsoft.com/silverlight/smoothstreaming/, accessed: 2022-07-12.
[46] "Enabling Low-Latency HTTP Live Streaming (HLS)," https://developer.apple.com/documentation/http_live_

streaming/enabling_low-latency_http_live_streaming_hls, accessed: 2022-07-01.

[47] Bitmovin, "2021 Bitmovin Video Developer Report," https://go.bitmovin.com/video-developer-report-2021, Bitmovin, Tech. Rep., 2021.

[48] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, 2011.

[49] "Real-Time Messaging Protocol (RTMP)," https://rtmp.veriskope.com/docs/spec/, accessed: 2022-06-29.

[50] "dash.js – Low Latency Streaming with CMAF," https://websites.fraunhofer.de/video-dev/dash-js-low-latency-streaming-with-cmaf/, accessed: 2022-07-07.

[51] "DASH Industry Forum - Part 4: Live and Low-Latency Services," https://dashif.org/guidelines/iop-v5/#part-4-live-and-low-latency-services, accessed: 2023-02-09.

[52] C. Timmerer and A. Bertoni, "Advanced Transport Options for the Dynamic Adaptive Streaming over HTTP," *CoRR*, vol. abs/1606.00264, 2016.

[53] J. Herbots *et al.*, "Cross-layer metrics sharing for QUICker video streaming," in *International Conference on Emerging Networking Experiments and Technologies*, 2020.

[54] "Real-Time Communication for the Web," https://webrtc.org/, accessed: 2022-07-07.

[55] S. Petrangeli *et al.*, "A Scalable WebRTC-Based Framework for Remote Video Collaboration Applications," *Multimedia Tools and Applications*, vol. 78, no. 6, 2019.

[56] F. Bannour *et al.*, "Distributed SDN Control: Survey, Taxonomy, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, 2018.

[57] F. Tashtarian *et al.*, "CoDeC: A Cost-Effective and Delay-Aware SFC Deployment," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, 2020.

[58] "VIVE," https://www.vive.com/, accessed: 2023-02-09.

[59] "Meta Quest," https://www.meta.com/be/en/quest/, accessed: 2023-02-09.

[60] L. Skorin-Kapov *et al.*, "A Survey of Emerging Concepts and Challenges for QoE Management of Multimedia Services," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 14, no. 2s, 2018.

[61] L. Rebenitsch and C. Owen, "Review on Cybersickness in Applications and Visual Displays," *Virtual Reality*, vol. 20, no. 2, 2016.

[62] M. S. Anwar *et al.*, "Subjective QoE of 360-Degree Virtual Reality Videos and Machine Learning Predictions," *IEEE Access*, vol. 8, 2020.

[63] J. Kim *et al.*, "Multisensory Integration and the Experience of Scene Instability, Presence and Cybersickness in Virtual Environments," *Computers in Human Behavior*, vol. 113, 2020.

[64] "OmniCam-360," https://www.hhi.fraunhofer.de/en/departments/vit/technologies-and-solutions/capture/panoramic-uhd-video/omnicam-360.html, accessed: 2023-02-09.

[65] H. Y. *et al.*, "JVET AHG Report: 360 Video Conversion Software Development," in *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-I0006*, 2018.

[66] R. Skupin *et al.*, "Tile-Based HEVC Video for Head Mounted Displays," in *IEEE International Symposium on Multimedia*, 2016.

[67] R. Skupin *et al.*, "Tile-Based Rate Assignment for 360-Degree Video Based on Spatio-Temporal Activity Metrics," in *IEEE International Symposium on Multimedia*, 2018.

[68] R. Skupin *et al.*, "Rate Assignment in 360-Degree Video Tiled Streaming Using Random Forest Regression," in *International Conference on Acoustics, Speech, and Signal Processing*, 2020.

[69] X. Corbillon *et al.*, "Viewport-Adaptive Navigable 360-Degree Video Delivery," in *IEEE International Conference on Communications*, 2017.

[70] M. Graf *et al.*, "Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP: Design, Implementation, and Evaluation," in *ACM Multimedia Systems Conference*, 2017.

[71] J. Le Feuvre and C. Concolato, "Tiled-Based Adaptive Streaming Using MPEG-DASH," in *ACM Multimedia Systems Conference*, 2016.

[72] M. Hosseini and V. Swaminathan, "Adaptive 360 VR Video Streaming: Divide and Conquer!" in *IEEE International Symposium on Multimedia*, 2016.

[73] S. Petrangeli *et al.*, "An HTTP/2-Based Adaptive Streaming Framework for 360° Virtual Reality Videos," in *ACM International Conference on Multimedia*, 2017.

[74] J. van der Hooft *et al.*, "Tile-Based Adaptive Streaming for Virtual Reality Video," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 4, 2019.

[75] J. Fu *et al.*, "360HRL: Hierarchical Reinforcement Learning Based Rate Adaptation for 360-Degree Video Streaming," in *International Conference on Visual Communications and Image Processing*, 2021.

[76] S. M. LaValle *et al.*, "Head Tracking for the Oculus Rift," in *IEEE International Conference on Robotics and Automation*, 2014.

[77] R. T. Azuma, "Predictive Tracking for Augmented Reality," Ph.D. dissertation, University of North Carolina at Chapel Hill, 1995.

[78] F. Qian *et al.*, "Optimizing 360 Video Delivery Over Cellular Networks," in *Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016.

[79] Y. Sanchez *et al.*, "Delay Impact on MPEG OMAF's Tile-Based Viewport-Dependent 360 Video Streaming," in *IEEE Journal on Emerging and selected Topics in Circuits And Systems*, 2019.

[80] J. Yang *et al.*, "QoE-Driven Resource Allocation Optimized for Uplink Delivery of Delay-Sensitive VR Video Over Cellular Network," *IEEE Access*, vol. PP, 2019.

[81] Z. Zhang *et al.*, "Saliency Detection in 360○ Videos," in *ECCV European Conference on Computer Vision*, 2018.

[82] P. Lebreton and A. Raake, "GBVS360, BMS360, ProSal: Extending existing saliency prediction models

from 2D to omnidirectional images," in *Elsevier Signal Processing: Image Communication*, vol. 69, 2018.

[83] D. Zhu *et al.*, "A Lightweight Saliency Prediction Model for Omnidirectional Images," in *IEEE/ICME International Conference on Multimedia and Expo*, 2021.

[84] Y. Sanchez *et al.*, "Shifted IDR Representations for Low Delay Live DASH Streaming using HEVC Tiles," in *IEEE International Symposium on Multimedia*, 2016.

[85] Y. Sanchez *et al.*, "Encoding Configurations for Tile-Based 360° Video," in *IEEE International Symposium on Multimedia)*, 2019.

[86] "Open Broadcaster Software," https://obsproject.com/, accessed: 2022-07-01.

[87] "Live Streaming Latency," https://support.google.com/youtube/answer/7444635, accessed: 2022-07-04.

[88] "Insta360 Pro 2," https://www.insta360.com/product/insta360-pro2, accessed: 2022-07-08.

[89] J. van der Hooft *et al.*, "An HTTP/2 Push-Based Approach for Low-Latency Live Streaming with Super-Short Segments," *Journal of Network and Systems Management*, vol. 26, no. 1, 2017.

[90] S. C. Madanapalli *et al.*, "Modeling Live Video Streaming: Real-Time Classification, QoE Inference, and Field Evaluation," *CoRR*, vol. abs/2112.02637, 2021.

[91] H. K. Ravuri *et al.*, "Partially Reliable Transport Layer for QUICker Interactive Immersive Media Delivery," in *Submitted to ACM International Conference on Interactive Media Experiences*, 2022.

[92] B. Oztas *et al.*, "A Study on the HEVC Performance Over Lossy Networks," in *IEEE International Conference on Electronics, Circuits, and Systems*, 2012.

[93] X. Liu and Y. Deng, "Learning-Based Prediction, Rendering and Association Optimization for MEC-Enabled Wireless Virtual Reality (VR) Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, 2021.

[94] S. Mangiante *et al.*, "Vr is on the edge: How to deliver 360° videos in mobile networks," in *ACM Workshop on Virtual Reality and Augmented Reality Network*, 2017.

[95] R. Schatz *et al.*, "Towards Subjective Quality of Experience Assessment for Omnidirectional Video Streaming," in *International Conference on Quality of Multimedia Experience*, 2017.

[96] W. Zhang *et al.*, "The Impact of Stalling on the Perceptual Quality of HTTP-based Omnidirectional Video Streaming," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[97] M. S. Anwar *et al.*, "Measuring Quality of Experience for 360-Degree Videos in Virtual Reality," *Science China Information Sciences*, vol. 63, no. 10, 2020.

[98] M. Seufert, "Quality of Experience and Access Network Traffic Management of HTTP Adaptive Video Streaming," Ph.D. dissertation, IEEE, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8406136/

[99] A. Singla *et al.*, "Comparison of Subjective Quality Evaluation for HEVC Encoded Omnidirectional Videos at Different Bit-rates for UHD and FHD Resolution," in *Thematic Workshops of ACM Multimedia*, 2017.

[100] H. T. T. Tran *et al.*, "A Subjective Study on QoE of 360 Video for VR Communication," in *IEEE International Workshop on Multimedia Signal Processing*, 2017.

[101] B. Zhang *et al.*, "Subjective and Objective Quality Assessment of Panoramic Videos in Virtual Reality Environments," in *IEEE International Conference on Multimedia Expo Workshops*, 2017.

[102] R. Schatz *et al.*, "Tile-based Streaming of 8K Omnidirectional Video: Subjective and Objective QoE Evaluation," in *International Conference on Quality of Multimedia Experience*, 2019.

[103] J. Ruan and D. Xie, "A Survey on QoE-Oriented VR Video Streaming: Some Research Issues and Challenges," *Electronics*, vol. 10, no. 17, 2021.

[104] M. Yu *et al.*, "A Framework to Evaluate Omnidirectional Video Coding Schemes," in *IEEE International Symposium on Mixed and Augmented Reality*, 2015.

[105] V. Zakharchenko *et al.*, "AhG8: Suggested Testing Procedure for 360-Degree Video," in *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-D0027*, 2016.

[106] W. Lo *et al.*, "360° Video Viewing Dataset in Head-Mounted Virtual Reality," in *ACM Multimedia Systems Conference*, 2017.

[107] S. Fremerey *et al.*, "AVtrack360: An Open Dataset and Software Recording People's Head Rotations Watching 360° Videos on an HMD," in *ACM Multimedia Systems Conference*, 2018.

[108] J. Gutiérrez *et al.*, "Introducing UN Salient360! Benchmark: A Platform for Evaluating Visual Attention Models for 360° Contents," in *International Conference on Quality of Multimedia Experience*, 2018.

[109] H. Cheng *et al.*, "Cube Padding for Weakly-Supervised Saliency Prediction in 360° Videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[110] A. Zare *et al.*, "HEVC-Compliant Tile-Based Streaming of Panoramic Video for Virtual Reality Applications," in *ACM Multimedia Conference*, 2016.

[111] F. Chiariotti, "A Survey on 360-Degree Video: Coding, Quality of Experience and Streaming," *CoRR*, vol. abs/2102.08192, 2021.

[112] C. Fan *et al.*, "A Survey on 360° Video Streaming: Acquisition, Transmission, and Display," *ACM Computing Surveys*, vol. 52, no. 4, 2019.

[113] R. Shafi *et al.*, "360-Degree Video Streaming: A Survey of the State of the Art," *Symmetry*, vol. 12, no. 9, 2020.

[114] M. Xu *et al.*, "State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, 2020.

[115] A. Yaqoob *et al.*, "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, 2020.

[116] A. Singla *et al.*, "Comparison of Subjective Quality Evaluation Methods for Omnidirectional Videos With

DSIS and Modified ACR," in *Human Vision and Electronic Imaging*, 2018.

[117] A. Singla *et al.*, "Subjective Quality Evaluation of Tile-based Streaming for Omnidirectional Videos," in *ACM Multimedia Systems Conference*, 2019.

[118] M. S. Anwar *et al.*, "Evaluating the Factors Affecting QoE of 360-Degree Videos and Cybersickness Levels Predictions in Virtual Reality," *Electronics*, vol. 9, no. 99, 2020.

[119] A. Singla *et al.*, "Measuring and Comparing QoE and Simulator Sickness of Omnidirectional Videos in Different Head Mounted Displays," in *International Conference on Quality of Multimedia Experience*, 2017.

[120] "Factory 42 and Hold the World," https://www.immerseuk.org/case-study/factory-42-and-hold-the-world, accessed: 2023-02-09.

[121] X. Roynard *et al.*, "Paris-Lille-3D: A Large and High-Quality Ground-Truth Urban Point Cloud Dataset for Automatic Segmentation and Classification," *The International Journal of Robotics Research*, vol. 37, no. 6, 2018.

[122] "Intel RealSense Technology," https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html, accessed: 2022-07-08.

[123] P. J. Besl and N. D. McKay, "Method for Registration of 3-D Shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, 1992.

[124] "MeshLab," https://www.meshlab.net/, accessed: 2022-07-28.

[125] "Point Cloud Library," https://pointclouds.org/, accessed: 2022-07-28.

[126] W. Yuan *et al.*, "DeepGMR: Learning Latent Gaussian Mixture Models for Registration," in *European Conference on Computer Vision*, 2020.

[127] X. Huang *et al.*, "A Comprehensive Survey on Point Cloud Registration," *Computing Research Repository*, vol. abs/2103.02690, 2021.

[128] S. Gumhold *et al.*, "Predictive Point-Cloud Compression," in *ACM SIGGRAPH Sketches*, 2005.

[129] B. Merry *et al.*, "Compression of Dense and Regular Point Clouds," in *International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, 2006.

[130] C. Cao *et al.*, "3D Point Cloud Compression: A Survey," in *International Conference on 3D Web Technology*, 2019.

[131] R. Mekuria *et al.*, "Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, 2017.

[132] MPEG, "MPEG 3DG and Requirements - Call for Proposals for Point Cloud Compression V2," 2017.

[133] A. Costa *et al.*, "Improved Patch Packing for the MPEG V-PCC Standard," in *IEEE International Workshop on Multimedia Signal Processing*, 2019.

[134] S. C. Park *et al.*, "Super-Resolution Image Reconstruction: A Technical Overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, 2003.

[135] A. Kappeler *et al.*, "Video Super-Resolution With Convolutional Neural Networks," *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, 2016.

[136] R. Li *et al.*, "PU-GAN: a Point Cloud Upsampling Adversarial Network," *CoRR*, vol. abs/1907.10844, 2019.

[137] A. Zhang *et al.*, "Efficient Volumetric Video Streaming Through Super Resolution," in *International Workshop on Mobile Computing Systems and Applications*, 2021.

[138] K. Lee *et al.*, "GROOT: A Real-Time Streaming System of High-Fidelity Volumetric Videos," in *International Conference on Mobile Computing and Networking*, 2020.

[139] P. A. Chou *et al.*, "A Volumetric Approach to Point Cloud Compression-Part I: Attribute Compression," *IEEE Transactions on Image Processing*, vol. 29, 2020.

[140] X. Sun *et al.*, "A Novel Point Cloud Compression Algorithm Based on Clustering," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, 2019.

[141] C. Tu *et al.*, "Real-Time Streaming Point Cloud Compression for 3D LiDAR Sensor Using U-Net," *IEEE Access*, vol. 7, 2019.

[142] F. Pereira *et al.*, "Point Cloud Coding: A Privileged View Driven by a Classification Taxonomy," *Signal Processing: Image Communication*, vol. 85, 2020.

[143] C. Cao *et al.*, "Compression of Sparse and Dense Dynamic Point Clouds - Methods and Standards," *Proceedings of the IEEE*, vol. 109, no. 9, 2021.

[144] J. Rossignac, "Edgebreaker: Connectivity Compression for Triangle Meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, 1999.

[145] P. Alliez and C. Gotsman, "Recent Advances in Compression of 3D Meshes," in *Advances in Multiresolution for Geometric Modelling*, 2005.

[146] E. Pavez and P. A. Chou, "Dynamic Polygon Cloud Compression," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[147] R. L. de Queiroz and P. A. Chou, "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform," *IEEE Transactions on Image Processing*, vol. 25, 2016.

[148] F. Nasiri *et al.*, "A Geometry-Aware Framework for Compressing 3D Mesh Textures," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[149] "Draco," https://github.com/google/draco, accessed: 2023-02-09.

[150] J. van der Hooft *et al.*, "Towards 6DoF HTTP Adaptive Streaming Through Point Cloud Compression," in *ACM International Conference on Multimedia*, 2019.

[151] S. Petrangeli *et al.*, "Dynamic Adaptive Streaming for Augmented Reality Applications," in *IEEE International Symposium on Multimedia*, 2019.

[152] M. Hosseini and C. Timmerer, "Dynamic Adaptive Point Cloud Streaming," in *Packet Video Workshop*, 2018.

[153] J. Park *et al.*, "Rate-Utility Optimized Streaming of Volumetric Media for Augmented Reality," *IEEE Jour-*

nal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, 2019.

[154] S. Subramanyam *et al.*, "User Centered Adaptive Streaming of Dynamic Point Clouds With Low Complexity Tiling," in *ACM International Conference on Multimedia*, 2020.

[155] B. Han *et al.*, "ViVo: Visibility-Aware Mobile Volumetric Video Streaming," in *International Conference on Mobile Computing and Networking*, 2020.

[156] S. Gül *et al.*, "Kalman Filter-Based Head Motion Prediction for Cloud-Based Mixed Reality," in *International Conference on Multimedia*, 2020.

[157] "Point Cloud Compression," https://www.dis.cwi.nl/pointcloud/, accessed: 2023-02-09.

[158] J. Chakareski *et al.*, "6DOF Virtual Reality Dataset and Performance Evaluation of Millimeter Wave vs. Free-Space-Optical Indoor Communications Systems for Lifelike Mobile VR Streaming," in *Asilomar Conference on Signals, Systems, and Computers*, 2020.

[159] S. Gül *et al.*, "Low-Latency Cloud-based Volumetric Video Streaming Using Head Motion Prediction," in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2020.

[160] J. Li *et al.*, "Optimal Volumetric Video Streaming with Hybrid Saliency-Based Tiling," *IEEE Transactions on Multimedia*, 2022.

[161] J. Jansen *et al.*, "A Pipeline for Multiparty Volumetric Video Conferencing: Transmission of Point Clouds over Low Latency DASH," in *ACM Multimedia Systems Conference*, 2020.

[162] "GPAC — Multimedia Open Source Project," https://gpac.wp.imt.fr/, accessed: 2022-07-29.

[163] J. Hu *et al.*, "Characterizing Real-Time Dense Point Cloud Capture and Streaming on Mobile Devices," in *ACM Workshop on Hot Topics in Video Analytics and Intelligent Edges*, 2021.

[164] S. Orts-Escolano *et al.*, "Holoportation: Virtual 3D Teleportation in Real-Time," in *ACM Symposium on User Interface Software and Technology*, 2016.

[165] S. Dijkstra-Soudarissanane *et al.*, "Multi-Sensor Capture and Network Processing for Virtual Reality Conferencing," in *ACM Multimedia Systems Conference*, 2019.

[166] F. Qian *et al.*, "Toward Practical Volumetric Video Streaming on Commodity Smartphones," in *International Workshop on Mobile Computing Systems and Applications*, 2019.

[167] E. Dumic and L. A. da Silva Cruz, "Point Cloud Coding Solutions, Subjective Assessment and Objective Measures: A Case Study," *Symmetry*, vol. 12, no. 12, 2020.

[168] A. Javaheri *et al.*, "Subjective and Objective Quality Evaluation of 3D Point Cloud Denoising Algorithms," in *IEEE International Conference on Multimedia Expo Workshops*, 2017.

[169] E. Alexiou *et al.*, "Point Cloud Subjective Evaluation Methodology based on 2D Rendering," in *International Conference on Quality of Multimedia Experience*, 2018.

[170] E. Alexiou *et al.*, "A Comprehensive Study of the Rate-Distortion Performance in MPEG Point Cloud Compression," *APSIPA Transactions on Signal and Information Processing*, vol. 8, 2019.

[171] J. van der Hooft *et al.*, "Objective and Subjective QoE Evaluation for Adaptive Point Cloud Streaming," in *International Conference on Quality of Multimedia Experience*, 2020.

[172] K. Cao *et al.*, "Visual Quality of Compressed Mesh and Point Cloud Sequences," *IEEE Access*, vol. 8, 2020.

[173] S. Subramanyam *et al.*, "Comparing the Quality of Highly Realistic Digital Humans in 3DoF and 6DoF: A Volumetric Video Case Study," in *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2020.

[174] E. Alexiou *et al.*, "PointXR: A Toolbox for Visualization and Subjective Evaluation of Point Clouds in Virtual Reality," *International Conference On Quality Of Multimedia Experience*, 2020.

[175] E. Zerman *et al.*, "User Behaviour Analysis of Volumetric Video in Augmented Reality," in *International Conference on Quality of Multimedia Experience*, 2021.

[176] M. Krivokuća *et al.*, "8i Voxelized Surface Light Field (8iVSLF) Dataset," in *ISO/IEC JTC1/SC29 WG11 m42914*, 2017.

[177] M. Krivokuća *et al.*, "Owlii Dynamic Human Mesh Sequence Dataset," in *ISO/IEC JTC1/SC29/WG11 m41658*, 2017.

[178] H. Joo *et al.*, "Panoptic Studio: A Massively Multiview System for Social Motion Capture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, 2017.

[179] I. Reimat *et al.*, "CWIPC-SXR: Point Cloud Dynamic Human Dataset for Social XR," in *ACM Multimedia Systems Conference*, 2021.

[180] L. Wang *et al.*, "QoE-Driven and Tile-Based Adaptive Streaming for Point Clouds," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

[181] Y. Alkhalili *et al.*, "A Survey of Volumetric Content Streaming Approaches," in *IEEE International Conference on Multimedia Big Data*, 2020.

[182] L. Sun *et al.*, "Multi-Path Multi-Tier 360-Degree Video Streaming in 5G Networks," in *ACM Multimedia Systems Conference*, 2018.

[183] E. Dumic *et al.*, "Subjective Evaluation and Objective Measures for Point Clouds - State of the Art," in *International Colloquium on Smart Grid Metrology*, 2018.

[184] Y. Nehmé *et al.*, "Visual Quality of 3D Meshes With Diffuse Colors in Virtual Reality: Subjective and Objective Evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, 2021.

[185] Y. Nehmé *et al.*, "Comparison of Subjective Methods for Quality Assessment of 3D Graphics in Virtual Reality," *ACM Transactions on Applied Perception*, vol. 18, no. 1, 2021.

[186] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," in *Computational Models of Visual Processing*, 1991.

[187] I. Viola *et al.*, "Comparison and Evaluation of Light Field Image Coding Approaches," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, 2017.

[188] T. Georgiev *et al.*, "Lytro Camera Technology: Theory, Algorithms, Performance Analysis," in *Multimedia Content and Mobile Devices*, 2013.

[189] "The Stanford Multi-Camera Array," http://graphics.stanford.edu/projects/array/, accessed: 2023-02-09.

[190] "Fraunhofer IIS Light-Field Technology," https://www.iis.fraunhofer.de/en/ff/amm/for/forschbewegtbildtechn/lichtfeld.html, accessed: 2023-02-09.

[191] T. Wang *et al.*, "Light Field Video Capture Using a Learning-Based Hybrid Imaging System," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[192] M. Wijnants *et al.*, "Standards-Compliant HTTP Adaptive Streaming of Static Light Fields," in *ACM Symposium on Virtual Reality Software and Technology*, 2018.

[193] D. Gabor, "A New Microscopic Principle," *Nature*, vol. 161, no. 4098, 1948.

[194] T. Tahara *et al.*, "Digital Holography and Its Multidimensional Imaging Applications: A Review," *Microscopy*, vol. 67, no. 2, 2018.

[195] M. K. Kim, "Principles and Techniques of Digital Holographic Microscopy," *SPIE Reviews*, vol. 1, 2010.

[196] F. Charrière *et al.*, "Characterization of Microlenses by Digital Holographic Microscopy," *Applied Optics*, vol. 45, no. 5, 2006.

[197] C. J. Mann *et al.*, "High-Resolution Quantitative Phase-Contrast Microscopy by Digital Holography," *Optics Express*, vol. 13, no. 22, 2005.

[198] I. Yamaguchi *et al.*, "Surface Shape Measurement by Phase-Shifting Digital Holography With a Wavelength Shift," *Applied Optics*, vol. 45, no. 29, 2006.

[199] B. Javidi and E. Tajahuerce, "Three-Dimensional Object Recognition by Use of Digital Holography," *Optics Letters*, vol. 25, no. 9, 2000.

[200] F. Yaraş *et al.*, "State of the Art in Holographic Displays: A Survey," *Journal of Display Technology*, vol. 6, no. 10, 2010.

[201] D. Blinder *et al.*, "Signal Processing Challenges for Digital Holographic Video Display Systems," *Signal Processing: Image Communication*, vol. 70, 2019.

[202] D. Blinder and P. Schelkens, "Accelerated Computer Generated Holography Using Sparse Bases in the STFT Domain," *Optics Express*, vol. 26, no. 2, 2018.

[203] H. Nishi and K. Matsushima, "Rendering of Specular Curved Objects in Polygon-Based Computer Holography," *Applied Optics*, vol. 56, no. 13, 2017.

[204] N. Okada *et al.*, "Band-Limited Double-Step Fresnel Diffraction and Its Application to Computer-Generated Holograms," *Optics Express*, vol. 21, no. 7, 2013.

[205] L. Shi *et al.*, "Towards Real-Time Photorealistic 3D Holography With Deep Neural Networks," *Nature*, vol. 592, 2021.

[206] S. Igarashi *et al.*, "Efficient Tiled Calculation of Over-10-Gigapixel Holograms Using Ray-Wavefront Conversion," *Optics Express*, vol. 26, no. 8, 2018.

[207] M. B. de Carvalho *et al.*, "A 4D DCT-Based Lenslet Light Field Codec," in *IEEE International Conference on Image Processing*, 2018.

[208] A. Aggoun, "Compression of 3D Integral Images Using 3D Wavelet Transform," *Journal of Display Technology*, vol. 7, no. 11, 2011.

[209] H. Kang *et al.*, "Compression Scheme of Sub-Images Using Karhunen-Loeve Transform in Three-Dimensional Integral Imaging," *Optics Communications*, vol. 281, no. 14, 2008.

[210] V. R. M. Elias and W. Martins, "On the Use of Graph Fourier Transform for Light-Field Compression," *Journal of Communication and Information Systems*, vol. 33, 2018.

[211] J. Khoury *et al.*, "A New Prediction Structure for Efficient MV-HEVC-Based Light Field Video Compression," in *International Conference on Computing, Networking and Communications*, 2019.

[212] H. Amirpour *et al.*, "High Efficient Snake Order Pseudo-Sequence Based Light Field Image Compression," in *Data Compression Conference*, 2018.

[213] E. Dib *et al.*, "Super-Ray-Based Low Rank Approximation for Light Field Compression," in *Data Compression Conference*, 2019.

[214] W. Ahmad *et al.*, "Interpreting Plenoptic Images as Multi-View Sequences for Improved Compression," in *IEEE International Conference on Image Processing*, 2017.

[215] J. Hou *et al.*, "Light Field Image Compression Based on Bi-Level View Compensation With Rate-Distortion Optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, 2019.

[216] C. Jia *et al.*, "Light Field Image Compression Using Generative Adversarial Network-Based View Synthesis," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, 2019.

[217] H. Amirpour *et al.*, "SLFC: Scalable Light Field Coding," in *Data Compression Conference*, 2021.

[218] G. Wang *et al.*, "Light Field Multi-View Video Coding With Two-Directional Parallel Inter-View Prediction," *IEEE Transactions on Image Processing*, vol. 25, no. 11, 2016.

[219] N. Mehajabin *et al.*, "An Efficient Pseudo-Sequence-Based Light Field Video Coding Utilizing View Similarities for Prediction Structure," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, 2022.

[220] B. Wang *et al.*, "Learning-Based High-Efficiency Compression Framework for Light Field Videos," *Multimedia Tools and Applications*, vol. 81, no. 6, 2022.

[221] T. Senoh *et al.*, "Multiview Image and Depth Map Coding for Holographic TV System," *Optical Engineering*, vol. 53, no. 11, 2014.

[222] B. M. V. *et al.*, "Holographic Representation: Hologram Plane vs. Object Plane," *Signal Processing: Image Communication*, vol. 68, 2018.

[223] P. Astola *et al.*, "JPEG Pleno: Standardizing a Coding

Framework and Tools for Plenoptic Imaging Modalities," *ITU Journal: ICT Discoveries*, vol. 3, 2020.

[224] R. S. Overbeck *et al.*, "A System for Acquiring, Processing, and Rendering Panoramic Light Field Stills for Virtual Reality," *ACM Transactions on Graphics*, vol. 37, no. 6, 2018.

[225] J. Daniel *et al.*, "Initial Work on Development of an Open Streaming Media Standard for Field of Light Displays (SMFoLD)," in *International Symposium on Electronic Imaging*, 2018.

[226] H. Lievens *et al.*, "Adaptive Streaming and Rendering of Static Light Fields in the Web Browser," in *International Conference on 3D Immersion*, 2021.

[227] A. El Rhammad *et al.*, "Towards Practical Hologram Streaming Using Progressive Coding," in *Applications of Digital Image Processing XLII*, 2019.

[228] A. El Rhammad *et al.*, "Scalable Coding Framework for a View-Dependent Streaming of Digital Holograms," in *IEEE International Conference on Image Processing*, 2019.

[229] H. Amirpour *et al.*, "Towards View-Aware Adaptive Streaming of Holographic Content," in *IEEE International Conference on Multimedia Expo Workshops*, 2020.

[230] H. Amirpour *et al.*, "Advanced Scalability for Light Field Image Coding," *IEEE Transactions on Image Processing*, vol. 31, 2022.

[231] V. Avramelos *et al.*, "Random Access Prediction Structures for Light Field Video Coding With MV-HEVC," *Multimedia Tools and Applications*, 2020.

[232] J. Zhang *et al.*, "Saliency Detection on Light Field: A Multi-Cue Approach," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 13, no. 3, 2017.

[233] T. Wang *et al.*, "Deep Learning for Light Field Saliency Detection," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[234] B. Wang *et al.*, "User-Dependent Interactive Light Field Video Streaming System," *Multimedia Tools and Applications*, 2021.

[235] I. Viola *et al.*, "A New Approach to Subjectively Assess Quality of Plenoptic Content," in *Applications of Digital Image Processing XXXIX*, 2016.

[236] I. Viola *et al.*, "Impact of Interactivity on the Assessment of Quality of Experience for Light Field Content," in *International Conference on Quality of Multimedia Experience*, 2017.

[237] H. Amirpour *et al.*, "Reliability of the Most Common Objective Metrics for Light Field Quality Assessment," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[238] S. Mahmoudpour and P. Schelkens, "On the Performance of Objective Quality Metrics for Lightfields," *Signal Processing: Image Communication*, vol. 93, 2021.

[239] X. Min *et al.*, "A Metric for Light Field Reconstruction, Compression, and Display Quality Evaluation," *IEEE Transactions on Image Processing*, vol. 29, 2020.

[240] C. Meng *et al.*, "Objective Quality Assessment of Lenslet Light Field Image Based on Focus Stack," *IEEE Transactions on Multimedia*, 2021.

[241] J. Xiang *et al.*, "Pseudo Video and Refocused Images-Based Blind Light Field Image Quality Assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, 2021.

[242] S. B. Shum, H.and Kang, "Review of Image-Based Rendering Techniques," in *Visual Communications and Image Processing*, 2000.

[243] M. Levoy and H. Pat, "Light Field Rendering," in *Conference on Computer Graphics and Interactive Techniques*, 1996.

[244] A. Ahar *et al.*, "Suitability Analysis of Holographic vs Light Field and 2D Displays for Subjective Quality Assessment of Fourier Holograms," *Optics Express*, vol. 28, no. 24, 2020.

[245] R. Corda and C. Perra, "Hologram Domain Data Compression: Performance of Standard Codecs and Image Quality Assessment at Different Distances and Perspectives," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, 2020.

[246] H. Amirpour *et al.*, "Quality Evaluation Of Digital Holographic Data Encoded On The Object Plane Using State Of The Art Codecs," in *IEEE International Conference on Image Processing*, 2020.

[247] H. Amirpourazarian *et al.*, "Quality Evaluation of Holographic Images Coded With Standard Codecs," *IEEE Transactions on Multimedia*, 2021.

[248] A. Ahar *et al.*, "Validation of dynamic subjective quality assessment methodology for holographic coding solutions," in *International Conference on Quality of Multimedia Experience*, 2021.

[249] X. Hu *et al.*, "4DLFVD: A 4D Light Field Video Dataset," 2021.

[250] M. Řeřábek and T. Ebrahimi, "New Light Field Image Dataset," *International Conference on Quality of Multimedia Experience*, 2016.

[251] K. Honauer *et al.*, "A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields," in *Computer Vision - ACCV 2016*, 2017.

[252] S. Shekhar *et al.*, "Light-Field Intrinsic Dataset," in *British Machine Vision Conference*, 2018.

[253] V. K. Adhikarla *et al.*, "Towards a Quality Metric for Dense Light Fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[254] L. Guillo *et al.*, "Light Field Video Dataset Captured by a R8 Raytrix Camera (With Disparity Maps)," in *ISO /IEC JTC1/SC29/WG11 MPEG2018/m42468, ISO/IEC JTC1/SC29/WG1 JPEG2018/m79046, ISO/IEC JTC1/SC29/WG1 & WG11, April 2018*, 2018.

[255] T. Kinoshita and S. Ono, "Sintel 4D Light Field Video Dataset," 2020.

[256] P. Paudyal *et al.*, "SMART: A Light Field Image Quality Dataset," in *ACM Multimedia Systems Conference*, 2016.

[257] A. Gilles *et al.*, "Computer Generated Hologram from

Multiview-Plus-Depth Data Cnsidering Specular Reflections," in *IEEE International Conference on Multimedia Expo Workshops*, 2016.

[258] A. Symeonidou *et al.*, "Speckle Noise Reduction for Computer Generated Holograms of Objects With Diffuse Surfaces," in *Optics, Photonics and Digital Technologies for Imaging Applications IV*, vol. 9896, 2016.

[259] C. Conti *et al.*, "Dense Light Field Coding: A Survey," *IEEE Access*, vol. 8, 2020.

[260] S. Zhou *et al.*, "Review of Light Field Technologies," *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 1, 2021.

[261] G. Wu *et al.*, "Light Field Image Processing: An Overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, 2017.

[262] E. Sahin *et al.*, "Computer-Generated Holograms for 3D Imaging: A Survey," *ACM Computing Surveys*, vol. 53, no. 2, 2020.

[263] A. Haleem *et al.*, "Holography and its Applications for Industry 4.0: An Overview," *Internet of Things and Cyber-Physical Systems*, vol. 2, 2022.

[264] "Media Over QUIC," https://datatracker.ietf.org/wg/moq/about/, accessed: 2023-02-09.

[265] J. Gruessing and S. Dawkins, "Media Over QUIC - Use Cases and Requirements for Media Transport Protocol Design," Internet Engineering Task Force, Internet-Draft draft-gruessing-moq-requirements-03, 2022, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-gruessing-moq-requirements/03/

[266] Z. N. Abdullah *et al.*, "Segment Routing in Software Defined Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, 2019.

[267] J. Santos *et al.*, "SRFog: A flexible architecture for Virtual Reality content delivery through Fog Computing and Segment Routing," in *IFIP/IEEE International Symposium on Integrated Network Management*, 2021.

[268] N. Villanueva, *Beginning 3D Game Assets Development Pipeline: Learn to Integrate from Maya to Unity*. Apress, 2022, ch. Rigging the Mech.

[269] B. Bross *et al.*, "Overview of the Versatile Video Coding (VVC) Standard and its Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, 2021.

[270] Y. Zhang *et al.*, "Machine Learning Based Video Coding Optimizations: A Survey," *Information Sciences*, vol. 506, 2020.

[271] C. Timmerer *et al.*, "Special Issue on Open Media Compression: Overview, Design Criteria, and Outlook on Emerging Standards," *Proceedings of the IEEE*, vol. 109, no. 9, 2021.

[272] D. Ding *et al.*, "Advances in Video Compression System Using Deep Neural Network: A Review and Case Studies," *Proceedings of the IEEE*, vol. 109, no. 9, 2021.

[273] J. M. Boyce *et al.*, "MPEG Immersive Video Coding Standard," *Proceedings of the IEEE*, vol. 109, no. 9, 2021.

[274] M. Slater and A. Steed, "A Virtual Presence Counter," *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 5, 2000.

[275] M. Mu *et al.*, "User Attention and Behaviour in Virtual Reality Art Encounter," *CoRR*, vol. abs/2005.10161, 2020.

[276] S. Rossi *et al.*, "Spherical Clustering of Users Navigating 360° Content," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[277] M. Borg *et al.*, "Practical Implementation of a Graphics Turing Test," in *Advances in Visual Computing*, 2012.

[278] T. Renshaw *et al.*, "Fundamentals for a Turing Test of Virtual Reality," *Human Factors and Ergonomics Society Annual Meeting*, vol. 60, no. 1, 2016.