

## Research Article

# Mobile Edge Assisted Live Streaming System for Omnidirectional Video

Xinjue Hu <sup>1,2</sup>, Wei Quan,<sup>1,2</sup> Tao Guo,<sup>1,2</sup> Yu Liu,<sup>1,2</sup> and Lin Zhang <sup>1,2</sup>

<sup>1</sup>*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*

<sup>2</sup>*Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518055, China*

Correspondence should be addressed to Lin Zhang; zhanglin@bupt.edu.cn

Received 28 January 2019; Accepted 14 April 2019; Published 21 May 2019

Guest Editor: Tara A. Yahiya

Copyright © 2019 Xinjue Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a popular form of virtual reality (VR) media, omnidirectional video (OV) has been continuously developed in recent years. OV contains the view of the scene in every direction, which will ask for around 120 Mbps with 8k resolution and 25 fps (frames per second). Although there has been a lot of work to optimize the transmission for on-demand of OV, the research on the live streaming of OV is still very lacking. Another big challenge for the OV live streaming system is the huge demand for computing resources. The existing terminal devices are difficult to completely carry tasks such as stitching, encoding, and rendering. This paper proposes a mobile edge assisted live streaming system for omnidirectional video (MELiveOV); the MELiveOV can intelligently offload the processing tasks to the edge computing enabled 5G base stations. The MELiveOV consists of an omnidirectional video generation module, a streaming module, and a viewpoint prediction module. A prototype system of MELiveOV is implemented to prove its complete end-to-end OV live streaming service. Evaluation result demonstrates that compared with the traditional solution, MELiveOV can reduce the network bandwidth requirement by about 50% and the transmission delay of more than 70% while ensuring the quality of the user's experience.

## 1. Introduction

According to the report [1] commissioned by Intel and conducted by Ovum, VR and AR applications will account for 90 percent of 5G data use over the next decade. Omnidirectional video (OV) is one of the most mature forms of VR, and it is expected to become the killer application for the future of 5G networks [2]. Driven by more powerful network performance, the 5G-powered OV not only focuses on realistic visual effects but also emphasizes the user's interactive experience. The basis of the interactive OV application is the ability to implement a complete end-to-end live streaming service system, which is the core problem that this work wants to solve. The popularity of OV technology brings viewers a novel immersive multimedia experience, but this new experience is supported by video contents with very high resolution (usually 4k or 8k) multiplied by 360-degree panoramic viewpoint. The transmission of OV usually

consumes 4~6x the bandwidth of a regular video with the same viewable resolution, which means a huge challenge to traditional video streaming architecture. The more amount of data and more complex computing tasks are the two major challenges that the OV live streaming system needs to address.

On the one hand, OV uses head mount displays (HMDs) with stereoscopic capabilities to provide the immersive experience. When the omnidirectional content is viewed by users, only a subset of the entire video frame will be displayed on the HMD's screen. To reduce the waste of network bandwidth caused by the redundancy of OV data, various improved solutions have been proposed both in academic and industrial communities. Some research works [3–5] have designed the tile-based coding scheme, which can effectively optimize the OV transmission. Many relevant Standards Development Organizations (SDOs) also have started work on the scope of OV [6]. But most of these works are carried out around the application for on-demand of

OV. Now more and more users are paying attention to the live streaming experience, which is also the development trend of digital multimedia technology in the future. Therefore, there is an urgent need for a new feasible solution that can minimize the bandwidth requirement of OV while simultaneously maximizing the user's experience.

On the other hand, in addition to the high bandwidth consumption during transmission, the huge demand for computing resources is another big challenge for the design of the OV live streaming system. The acquisition and generation of OV content require extensive stitching and encoding work. Especially when performing OV live streaming service, these computing works need to be completed in real time, which puts extremely high demands on the performance of the processing platform. When the OV streaming is viewed, the system can customize the process of rendering according to different fields of view (FOV) for multiple users. Accurate prediction of the user's viewpoint can bring great benefits to the optimization of the OV live streaming system. By predicting the FOV areas that users may be watching in the near future, the transfer of data in those useless areas can be avoided. And the OV live streaming system is able to use limited bandwidth to maximize the image quality of the FOV area. Viewpoint prediction relies on deep learning neural network algorithm, which also has a high demand for computational power. Offloading computationally intensive tasks to resourceful cloud/fog servers is necessary to reduce the pressure of the users' devices while saving the cost of the OV devices. Compared with the traditional central cloud server, the mobile edge computing (MEC) architecture can bring computing resources closer to users, thus greatly reducing the response delay of users requesting services.

A mobile edge assisted live streaming system for omnidirectional video (MELiveOV) is presented in this work to address the above challenges. As shown in Figure 1, it consists of the omnidirectional video generation module, the streaming module, and the viewpoint prediction module. Through the collaborative work of each module, MELiveOV achieves the complete end-to-end live streaming service of the omnidirectional video. Meanwhile, its edge computing architecture closely matches the needs of the 5G network and has a very broad application prospect. We implement the prototype system of MELiveOV and evaluate various performance metrics for it. The evaluation result shows that MELiveOV can effectively reduce the network bandwidth requirement and the transmission delay during the OV live streaming.

The contributions of this paper are summarized as follows:

- (i) We build an end-to-end mobile edge assisted live streaming system for omnidirectional video (MELiveOV). With the help of the MEC architecture, MELiveOV is able to perform well in both service latency and bandwidth requirements.
- (ii) In order to speed up the real-time generation of the omnidirectional video after the acquisition, we design an improved stitching algorithm based on the overall mapping table.

- (iii) A tile-based omnidirectional video transmission scheme is introduced to MELiveOV to reduce the pressure on network bandwidth during OV live streaming.
- (iv) In order to enhance the user's quality of experience and reduce the service delay, we design a user's viewpoint prediction algorithm, which enables MELiveOV to provide proactive service for users.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 introduces the system architecture of MELiveOV. Section 4 presents the design of omnidirectional video generation module based on the overall mapping stitching table. Section 5 presents the structure of tile-based streaming module. Section 6 introduces the architecture of the viewpoint prediction module using deep learning. Section 7 describes our implementation and evaluation. Section 8 concludes the paper and discusses the future work.

## 2. Related Works

Live streaming of events has been traditionally done by using broadcast TVs. DASH can also be applied to live streaming over the Internet [7, 8], despite the tighter latency constraints compared to on-demand video services. The challenge of live streaming is to minimize the end-to-end delay between the content generation (at the server) and presentation (at the client). The main research in video streaming areas focuses on optimization of different aspects like higher resolution (e.g., omnidirectional video and virtual reality streaming), lower latency, higher compression ratio, and better quality of experience (QoE). Many focus on adaptive streaming to fit as many network situations as possible and make full usage of possible bandwidth. In [9], subjective studies that cover QoE aspects of adaptation dimensions and strategies are revisited. As a result, QoE influence factors of HAS and corresponding QoE models are identified, and open issues and conflicting results are also discussed. The tiled video source is a sounding way to obtain adaptive streaming [10]; they describe how spatial access can be performed in an adaptive HTTP streaming context, using MPEG-DASH and its SRD extensions. They describe a configurable implementation of these technologies, within the GPAC open-source player, allowing experimentations of different adaptation policies for tiled video content. New scenarios enabled by the development of technology like virtual reality (VR) have attracted great attention. Ozcinar et al. [5] proposed an end-to-end streaming system implementation that contains tiling, a novel extension of the MPD, and DASH bitrate level selection in a viewport-aware manner, which can bring significant quality enhancements compared with the traditional streaming approach. In [11], a novel wireless video transmission method is developed, where the authors jointly investigate how to conquer the problem of source video's huge size, how to efficiently satisfy a user's view switch request, and how to handle packet loss. In [12], they develop a prototype of VR live architecture that combines RTP and DASH to deliver 360° VR content to a

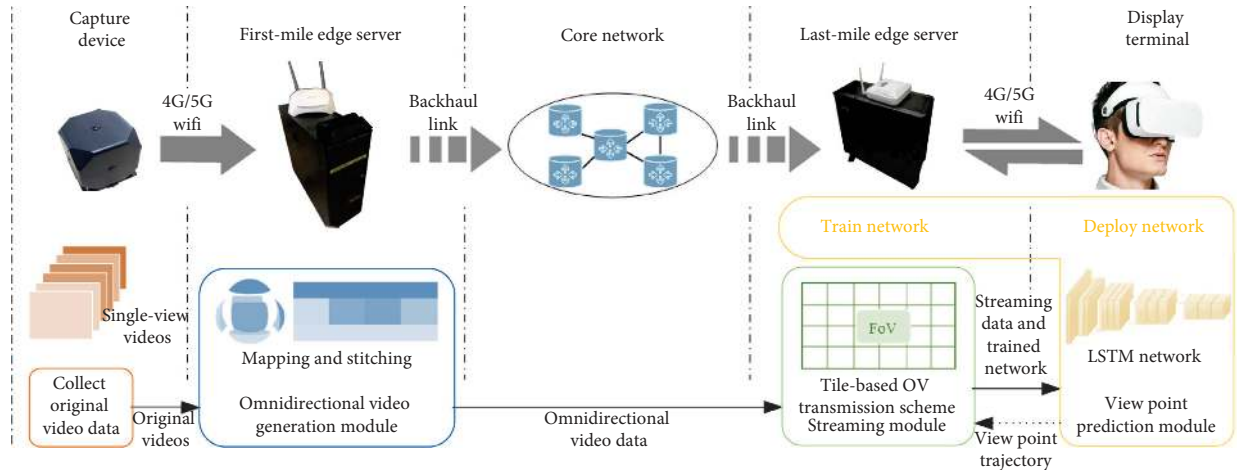


FIGURE 1: Overview of mobile edge assisted live streaming system for omnidirectional video (MELiveOV).

Huawei set-top-box and a Samsung Galaxy S7. The system multiplexes a single HEVC hardware decoder to provide faster quality switching than at the traditional group of pictures (GOP) boundaries.

As for the QoE aspect, in [13], it is believed that cellular operators and content providers can tremendously improve video QoE by predicting available bandwidth and sharing it through APIs. To be more specific, when combined with rate stabilization functions, prediction outperforms existing video streaming algorithms and reduces the gap with optimal to 4%. Besides, in [14], a layered framework for migrating active service applications that are encapsulated either in virtual machines (VMs) or containers is presented. This layering approach allows a substantial reduction in service downtime. The framework is easy to implement using readily available technologies, and one of its key advantages is that it supports containers, which is a promising emerging technology that offers benefits over VMs. Reducing delay is an attracting field as well. Machen et al. [15] presented a layered framework for migrating active service to MEC. This layering approach allows a substantial reduction in service downtime [16]. ENA develops a novel transmission scheduling framework dubbed AdaPtive HFR vIdEO Streaming (APHIS). It is proved by intensive experiments that APHIS framework is able to appropriately filter video frames and adjust data protection levels to optimize the quality of HFR video streaming. Sanchez et al. [17] presented a video coding and slicing scheme for OV streaming. In a delay-constrained circumstance, their scheme significantly reduces the transmission cost and enhances the quality of the reconstructed video sequences compared with the nonadaptive transmission scheme.

Omnidirectional video (OV) enables direct surround immersive viewing of a scene by warping the original image into the correct perspective given a viewing direction. A live streaming system for OV has been achieved in [18]. They design periodic and adaptive optimization frameworks to adapt to the bandwidth variations and FoV prediction errors in real time. OV can offer immersive visual experience when a user equipped with HMD, but

transmit OV with high bitrate will bring a heavy burden to transmission system especially in a real-time scenario. So, how to compress a video without affecting the user experience is very important. Chen et al. [19] reviewed the recent advances in the pipeline of omnidirectional video processing including projection and evaluation. An efficient way was achieved to facilitate motion-constrained HEVC tiles. Sreedhar et al. and Skupin et al. [20, 21] investigated various viewpoint dependent projection schemes, and they developed a methodology for comparing the rate-distortion performance of these projections. Yu et al. and Lee et al. [22, 23] considered the problem of evaluating the coding efficiency in the context of viewing with a HMD. They compared the original and coded videos on the viewpoint after sphere-to-plane mappings. It is observed that the equal-area mapping yields around 8.3% bitrate savings relative to the commonly used equirectangular mapping. Ghaznavi-Youvalari et al. and Curcio et al. [24, 25] adopted subjective assessment results of experiments using a tile-based streaming system for OV. This work reduces streaming bitrates by an average of 44% under a subjective DMOS value of 4.5. Yu et al. [26] showed a computationally efficient solution using Lagrangian optimization by separating the sampling and bit allocation constraints and got coding gains over standard representations. Graf et al. [27] described the usage of tiles in HEVC/H.265 and VP9, enabling bandwidth efficient adaptive streaming of omnidirectional video over HTTP. Various streaming strategies have been defined, which can effectively improve the quality of OV streaming service. Li et al. [28] proposed a tile-based omnidirectional video segmentation scheme which can save up to 28% of the pixel area and 20% of BD-rate averagely compared to the traditional equirectangular projection-based approach. Gudumasu et al. [29] showed a viewing orientation tracking and real-time viewpoint extraction platform. Generally, the user can only view a restricted field of view of the content. This means that a significant part of the bandwidth is wasted by transmitting quality video in regions that are not being visualized. So, there appears tile-based transmit method along with user's

viewpoint prediction. Ozcinar and Smolic [30] created a new visual attention user dataset for OV, investigated the behavior of viewers when consuming the content, and analyzed the prediction performance of state-of-the-art visual attention models. Ninan and Atluru [31] generated a second reconstructed image with view direction of the viewer when the user watched the first reconstructed image. Ghaznavi-Youvalari and Aminlou [32] proposed a geometry-based motion vector scaling method in order to compress the motion information of omnidirectional content efficiently. The result shows a 2.2% bitrate reduction with Versatile Video Coding (H.266/VVC) standard. Ghaznavi-Youvalari and Aminlou [33] divided the image into tiles and set different priorities with FOV information. The high-priority tiles will encode with a high bitrate. To measure the objective quality of omnidirectional video in observation space more accurately, a weighted-to-spherically-uniform quality evaluation method has been proposed in [34].

Many Standards Development Organizations in the field of multimedia and communications have also begun working on OV [6]. In R15 [35], 3GPP has started to consider the application of virtual reality (VR) media services in the next generation of mobile network. The Digital Video Broadcasting (DVB) Project established a VR-related commercial module to follow up on this area [36]. The Video Coding Experts Group (VCEG, ITU-T Q6/16) and the Moving Picture Experts Group (MPEG, ISO/IEC JTC 1/SC 29/WG 11) began the standardization process for OV, respectively, which started from the research of OV coding and transmission technology, and are expected to guide the development of the entire OV application ecosystem. There have been some joint groups to carry out some works in the field of OV, for example, the Joint Collaborative Team on Video Coding (JCT-VC), responsible for developing the High-Efficiency Video Coding (HEVC) standard and its extensions [37] and the Joint Video Exploration Team (JVET) that investigates new video coding approaches for coding efficiency beyond HEVC [38]. The coding of the omnidirectional video has attracted enough attention and has gradually become the focus of multimedia technology development. Compared with the VCEG, MPEG concentrated more on the technologies of delivery and display. MPEG established the subgroup of Omnidirectional Media Format (OMAF) [2], which is envisioned to become Part 2 of the emerging ISO/IEC 23090 MPEG-I.

### 3. System Architecture of MELiveOV

As an end-to-end service system, MELiveOV covers the entire service chain from acquisition to playback. As shown in Figure 1, we have designed the corresponding modified functional module at every stage of the OV live streaming service. In the conventional scheme, after the raw data are collected by the omnidirectional camera, the OV is generated directly locally. Constrained by the limited computing capability of the capture device, the omnidirectional video generation process will be extremely time

consuming, which greatly affects the real-time performance of the live streaming service. In MELiveOV, we offload the computational tasks required for the omnidirectional video generation process to the first-mile edge server. It is usually deployed at the access point of the first hop in the mobile communication network to provide the most timely service to the capture device. The mapping and stitching operations are then performed by the omnidirectional video generation module to obtain the OV data containing omnidirectional scene information.

Similarly, we deployed the last-mile edge server at the access point of the 5G network closest to the viewer. Before the last hop, the streaming module on the last-mile edge server will optimize the transmission in real-time based on the viewer's viewpoint trajectory fed back by the display terminal, which can effectively reduce the bandwidth requirement during the OV download process. The streaming module mainly applies a tile-based OV transmission scheme. By dividing the complete video into multiple tiles by spatial region, we can control the quality of different tiles to optimize transmission. We use a high bitrate for the tiles in the user's field of view (FOV) and a lower bitrate for the area outside the user's FOV. Through the interaction with the display terminal, the streaming module can minimize the transmission bandwidth requirement of the OV without sacrificing the quality of the user viewing area.

This can save the power consumption of VR devices and the costs of traffic for the users. On the display terminal of MELiveOV, we introduced the architecture of the proactive service. In the traditional reactive service architecture, the server can only respond and process after waiting for the user's request to arrive. For example, when our system is without the proactive service architecture, the optimization process performed by the streaming module can only rely on the user's past viewpoint data. This lag in user information can result in reduced performance of the streaming module. Therefore, we deploy the viewpoint prediction module to proactively predict the user's possible viewpoint location in the future. This can further improve the QoE (quality of experience) of MELiveOV's users. The viewpoint prediction module is designed based on the LSTM (long short-term memory) network. The LSTM is a model often used in deep learning to process time series predictive data problems. Our prediction model can not only learn the information of the user's personalized viewing habits but also can perceive the statistical distribution of video saliency only through the multiuser viewpoint data.

### 4. Omnidirectional Video Generation Module

The omnidirectional camera is generally composed of multiple cameras so that the image data of the scene can be collected from various directions. The most representative camera is a 6-lens omnidirectional camera that can capture up, down, left, right, front, and back six channels of video. These raw data need to be mapped and stitched to generate OV content. There are a variety of omnidirectional image

unfolding methods, such as equirectangular projection (ERP, the upper part of Figure 2), which is the most familiar rendering method for the average user, achieving unfolding by transforming the spherical image into rectangular space according to longitude and latitude; cube maps (the lower part of Figure 2) transform the sphere into cubes and then expand the six faces of the cube; Equi-Angular Cubemap (EAC) is an optimization of the traditional cube expansion, correcting the deformation of the cube expansion by keeping the pixels evenly sampled.

Usually, the omnidirectional camera needs to be processed offline for several hours after the acquisition to finally generate the omnidirectional video. This is obviously unacceptable for OV's live streaming service system, so we need to develop a dedicated fast real-time stitching algorithm for MELiveOV. Next, we will introduce the functional design of the omnidirectional video generation module in detail.

**4.1. Overview Structure of the Module.** Traditional omnidirectional image stitching requires dynamic estimation of the input of each camera at each moment. Firstly, feature point matching is needed to estimate the intrinsic parameters and extrinsic parameters of the camera, and then the overall white balance is performed on each image to facilitate deriving the best stitching mask between images. Finally, with best stitching masks between every two pictures found, all original pictures can be merged into the same coordinate system to form the omnidirectional frame. Apparently, the stitching procedure is considerably time consuming. However, real-time processing of omnidirectional images requires both high resolution and image quality and low latency simultaneously. Due to the computational power and the high complexity of the algorithm itself, quality and efficiency are a pair of mutually exclusive indicators. Limited by this situation, the traditional stitching method can hardly be implemented, resulting in the current fisheye image real-time stitching technology's lack of variety. In our scheme, the stitching mapping table, which describes the projection from pixel coordinate in each unit lens' image to the pixel coordinate in the final omnidirectional frame, is firstly decided, and then the mapping table is embedded in the image processing algorithm to achieve omnidirectional image stitching in real time each frame by each frame.

The procedure for obtaining the parameter mapping table in our scheme is as follows:

- (1) Input fisheye images and separately estimate the camera model to obtain the mapping of points on two-dimensional fisheye images to corrected three-dimensional points on hemispheres.
- (2) Scale three-dimensional corrected images, with the equirectangular projection (ERP) method, to unfolding pattern to prepare for subsequent processions.
- (3) Extract feature points and estimate to find the best math. Then, accordingly calculate the intrinsic parameters and the extrinsic parameters to register the spatial positional relationship between the images.

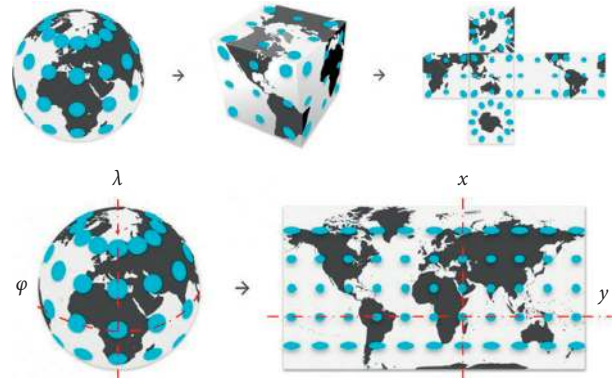


FIGURE 2: Cube maps and equirectangular projection.

- (4) According to the registration result, adjust the spatial positional relationship between the five hemispherical planes, superimpose the five-way corrected hemisphere in the world coordinate system, fuse image pixels on the overlap part, and then convert the three-dimensional image into an omnidirectional frame with ERP.
- (5) Extract and save the overall homography matrix of the coordinate on the fisheye image to the coordinate on the final omnidirectional frame for subsequent real-time processing.

#### 4.2. Camera Calibration and Camera Model Estimation.

The process of calibrating a camera model is actually a transformation estimate of a two-dimensional vector  $p$  in the original fisheye plane to a three-dimensional vector  $P$  in the world coordinates. Namely, to accomplish this process, the intrinsic and extrinsic parameters of cameras and the distortion parameters of lenses need to be estimated. The most commonly used technique for lens distortion parameter correction is polynomial fitting, and the pose estimation parameters of the unit lens are  $3 \times 3$  dimensional matrices. The relationship between the parameters of these two parts is a composite function. This composite optimization technique has a strong dependence on the initial value of the parameters, and the mutual interference is obvious as well, leading to difficulty in achieving global optimization. Our scheme treats the camera parameters and lens distortion as a combined system and estimates the transformation process as a whole. Map two-dimensional points on the fisheye plan to three-dimensional vectors and then convert them to points on the surface of a unit sphere to give a coordinate.

The camera model we use is presented in Figure 3. Let  $p$  be a pixel point in the original fisheye image,  $(u, v)$  is the pixel coordinate in term of the image center point as the origin, let  $P$  be its corresponding three-dimensional vector emanated from the single effective viewpoint, and  $(x, y, z)$  is the unit point in term of the optical axis as the origin. Since the plane coordinate transformation is an affine transformation, the relationship between  $p$  and  $P$  can be expressed as

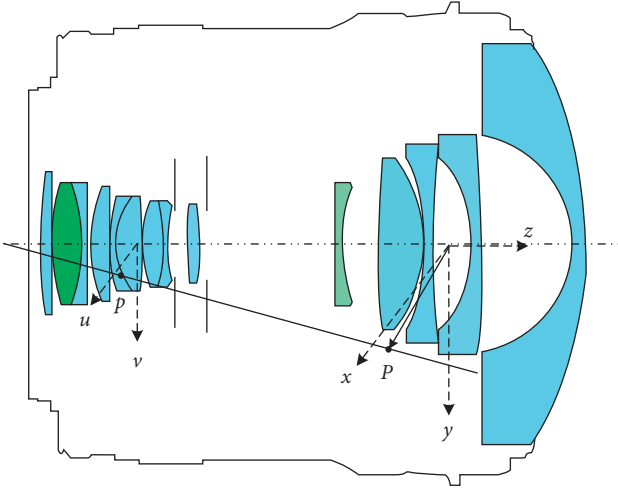


FIGURE 3: Camera model of fisheye camera.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} u \\ v \end{bmatrix}. \quad (1)$$

Then, the overall mapping of the two-dimensional plane coordinate to the three-dimensional vector can be written as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \\ f(u, v) \end{bmatrix} = \begin{bmatrix} u \\ v \\ f'(u, v) \end{bmatrix} = \begin{bmatrix} u \\ v \\ f'(\rho) \end{bmatrix}, \quad (2)$$

$f(u, v)$  is a function of two-dimensional coordinates,  $\rho = \sqrt{u^2 + v^2}$ .  $f'(\rho)$  is the polynomial function that needs fitting:

$$f' = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_4 x^4 + \dots \quad (3)$$

The polynomial fitting process is assisted by the Matlab toolbox `ocam_calib`. After a lot of experiments, the number of polynomial terms to be fitted is not as good as possible. To be more specific, the phenomenon of fitting degradation will occur for the sake of too many polynomial terms. Finally, it is determined that the four-term polynomial is used for fitting.

**4.3. Unfolding of the Spherical Image.** After obtaining the corrected three-dimensional hemisphere image from the fisheye image, we need to perform spherical unfolding for subsequent processing. In our scheme, the most widely used ERP is implemented to achieve unfolding.

As can be seen in the lower part of Figure 2,  $\lambda$  is the longitude of the location to project;  $\varphi$  is the latitude of the location to project;  $\varphi_1$  are the standard parallels (north and south of the equator) where the scale of the projection is true;  $\lambda_1$  is the central meridian of the map;  $x$  is the horizontal coordinate of the projected location on the map;  $y$  is the vertical coordinate of the projected location on the map. It can be concluded that

forward mapping:

$$x = (\lambda - \lambda_0) \cos(\varphi_0), y = (\varphi - \varphi_0), \quad (4)$$

reverse mapping:

$$\lambda = \frac{x}{\cos(\varphi_0)} + \lambda_0, \varphi = y + \varphi_0. \quad (5)$$

**4.4. Spatial Registration.** For physically setting the five cameras to be mutually orthogonal, theoretically, using the central camera's coordinate system as the world coordinate system and, respectively, rotating the corresponding coordinate system of other cameras by  $90^\circ$ , namely, multiplying the original three-dimensional coordinate matrix by the corresponding rotation matrix can guarantee a strict registered system, achieving three-dimensional space registration. However, considering the physical placement and the camera lens may introduce errors, and the center estimated by the fisheye correction process is not sufficiently the center of the original image; the edge may be misaligned when stitching, so the corrected three-dimensional spherical image needs to be registered again.

To perform registration, we first need to match and filter the feature points of the two images, select the best matching points, calculate the homography matrix, and then calculate the rotation matrix between adjacent two corrected pictures according to the homography matrix.

According to the principle of pinhole imaging, points in the camera coordinate system can be mapped to the world coordinate system via rotation and translation. The translation can be written as follows:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (6)$$

$$R = \begin{bmatrix} 0 & \theta_z & \theta_y \\ \theta_z & 0 & \theta_x \\ \theta_x & \theta_y & 0 \end{bmatrix},$$

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix},$$

where  $R$  is the rotation matrix and  $\theta_x, \theta_y$ , and  $\theta_z$  represent the angles at which the camera rotates around three coordinate axes.  $t$  is the translation vector, and  $t_x, t_y$ , and  $t_z$  are the translation distances of the camera along three coordinate axes.

Through calibrating, the extrinsic parameters matrix  $[R|t]$  of each camera relative to the center camera can be obtained, thereby completing the spatial registration.

**4.5. Generating the Overall Mapping Table.** Finally, the results of previous parts are combined, and overlapped pixels on the spherical surface are fused to produce a mapping table. The table describes how the source coordinate on the overall fisheye map will be transformed to the destination coordinate on the omnidirectional frame. With the mapping table fixed, by preparing multiple threads simultaneously

and executing pixel mapping operations on different regions of the panoramic frame, real-time stitching is available. The function display diagram of the omnidirectional video generation module is shown in Figure 4. Our module can complete the stitching process of an OV frame within 20 ms.

## 5. Streaming Module

High resolution and low transmission delay are the key points in the OV live streaming system. When the transmission delay reaches up to 13 ms or the bitrate is too low, users will feel tired and dizzy [39]. To ensure a good watching experience, the best way is to transmit omnidirectional video to display terminal, but this transmission method does not consider that viewer only watches a small portion of the entire full image. In fact, if OV player offers a 90 rectangle-view when the user looks at a certain direction, only one of the six spheres appear in the user's version and other parts will be out of sight. Transmitting non-FOV with a high bitrate will cause a huge waste of the network bandwidth. Therefore, we adopt a two-layer tile-based transmission mechanism to reduce the heavy burden on the transmission system.

*5.1. Projection of the OV Content.* After the omnidirectional video generation process, we get a spherical omnidirectional video which we cannot encode with existing coding standards such as H.264/AVC, H.265/HEVC. Since the encoder can only encode rectangular pictures, the omnidirectional video must be mapped into rectangular user view. A common method called equirectangular projection is to map the 3D sphere image to a 2D rectangular plane with longitude as a reference. However, the different visual angle on a panoramic sphere will result in different map areas. The closer to the two poles of the sphere, the more serious the image distortion. As Figure 5 shows, when the user looks at the equator position of the sphere, the projection area corresponding to the 2D plane is 11.9% of the entire panoramic frame. And the area gets the maximum value of 26.7% when view at the poles and is very distorted [40].

*5.2. Two-Layer Streaming Scheme for OV.* One method is to intercept the FOV area and transmit the FOV image with high bitrate to the client alone. Though it does not consider that the real-time OV system is latency sensitive, if user's head movement is too fast or the image cannot reach to display terminal in time, the display terminal will not have enough time to match the image properly. Users may see a blank area in this view, and it will seriously reduce the user's QoE.

So, we adopt a two-layer tile-based transmission mechanism, and Figure 6 shows the detailed process. First, after the equirectangular projection, the panoramic frame will be encoded by H.265/HEVC. And a low bitrate layer which we called basic layer (BL) will be generated. The BL represents the omnidirectional view at a low bitrate. At the same time, the panoramic frame is divided into  $4 \times 6$  tiles, and the tiles in the FOV area will be extracted by the encoder and encoded with high bitrate as tile enhanced layer (TEL). The FOV area

information such as the coordinate of the screen center collected by video client will return to the encoder side. BL and TEL will be transmitted to the client, and these two layers are superimposed on the client side to display.

In this two-layer tile-based transmission mechanism, encoder needs to encode the FOV tiles according to information which returned from display terminal to ensure system performance of MELiveOV. However, due to the random nature of viewer motion, it is very difficult to predict a long-term movement of the user's head. The accuracy will drop from 92% down to 71% when the time for prediction increases from 1 second to 2 seconds [41]. So, the prediction time we set is 1 second. According to the visual movement trajectory of the user in the first few seconds, the prediction algorithm shows the position of the user's viewpoint in the next second.

If the TEL has not arrived in time or TEL matches error, the client can display the BL to ensure a basic view experience rather than generate a blank area. Although this method will cause huge computation, transmission bandwidth is a more valuable resource. With the two-layer tile-based method, we solve the problem of unexpected head movement and network flow. The client can obtain a panoramic frame with the high bitrate of the FOV area and the low bitrate of the non-FOV area, which saves about 55% of the bandwidth consumption without affecting the user's QoE.

*5.3. Adaptive FOV Size Selection.* In the above two-layer transmission mechanism, a fixed FOV area is used. If the size of the FOV area can be dynamically selected according to different network condition, the system will be more adaptive. When the network is in a good condition, a larger area of high bitrate omnidirectional video can be obtained by the display terminal so that the user can get a better QoE.

Therefore, we adopt an adaptive strategy which allows the encoder to choose different FOV region size based on the network condition. Focusing on the user's viewpoint, we set the FOV area to  $90^\circ$  and  $120^\circ$ -degree FOV areas, respectively. When the network is in a bad condition, the encoder selects the FOV with  $90^\circ$ , a smaller panoramic frame area is encoded with the high bitrate. When the network condition is ideal, the encoder selects the FOV with  $120^\circ$  so that the larger panoramic frame area will be encoded with a relatively high bitrate.

The actual function of our two-layer OV transmission scheme is shown in Figure 7. It can be easily observed from the panoramic frame of OV that there is a significant difference in the video quality between FOV and non-FOV.

## 6. Viewpoint Prediction Module

Accurately predicting the viewer's future viewpoint trajectory can help MELiveOV to better enhance the user experience. Thus, we designed a special prediction model, which can provide users with effective viewpoint prediction at long intervals by using the local historical data and global multiuser information.

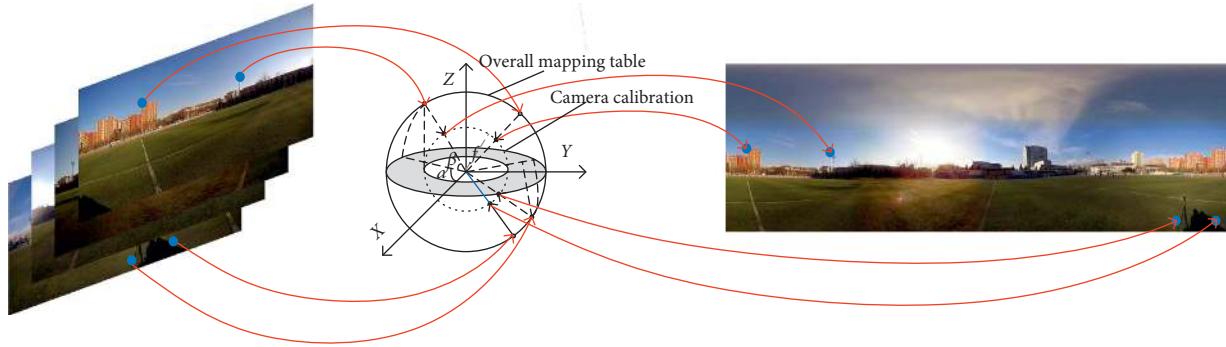


FIGURE 4: Function display of omnidirectional video generation module.

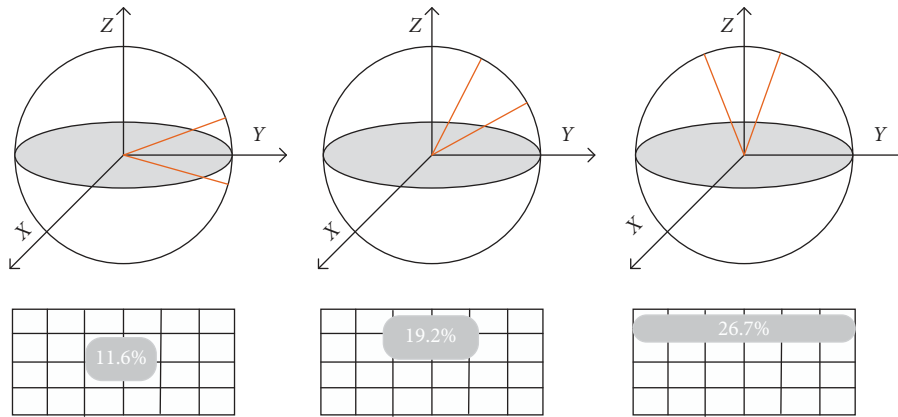


FIGURE 5: Projection from different angles.

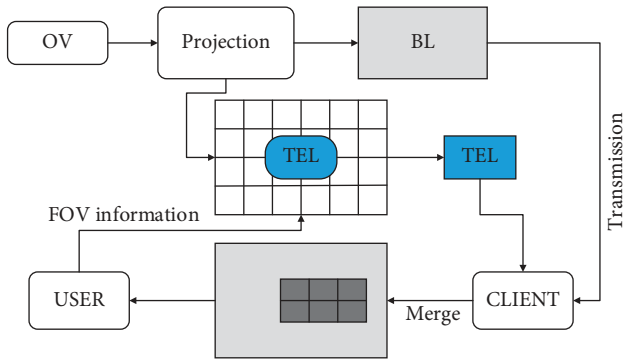


FIGURE 6: Two-layer streaming process.



FIGURE 7: Function display of two-layer OV transmission.

6.1. Overview of the Module. The problem of viewpoint prediction is considered from two perspectives in the viewpoint prediction module. On the one hand, most users are not watching the OV for the first time. Therefore, the historical viewpoint data of the OVs they have seen may contain some information about the user’s viewing habits. For example, some users may prefer to move their viewpoints slowly and smoothly, while other users prefer faster viewpoint movements. This customized information allows our module to be adaptable to different users. On the other hand, the OV content provider may already have collected the viewpoint trajectory data from multiple users for the same OV source. Through the analysis of the dataset, it can

be found that when different users watch the same OV, their viewpoint trajectory will have a similar movement pattern. This is because some frames of the OV have the content that can arouse most users’ interest. When viewing these frames, different users tend to focus on the same region of interest, so the viewpoint trajectory will have a similar movement pattern. In this way, these existing models will help provide more accurate viewpoint prediction services as new users begin to watch.

The overview flow of viewpoint prediction module is shown in Figure 8. In the proposed method, the viewpoint prediction system includes two independent channels, one



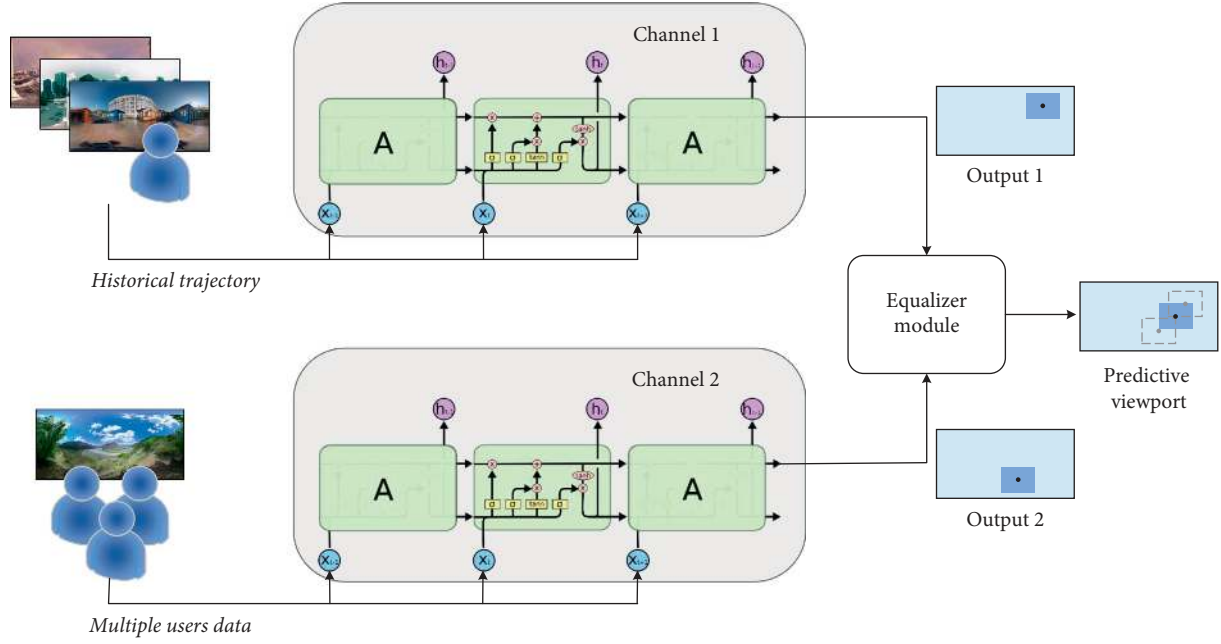


FIGURE 8: System flow of CPVp-LSTM.

of which makes the prediction based on historical viewpoint data of the single user. And the second channel will use the trajectory data of other people from the same OV content to predict the viewpoint. After the output of both channels passes through the equalizer model, the final prediction result can be obtained.

As shown in Figure 8, both channels of viewpoint prediction module implement prediction functions through the LSTM (long short-term memory) network. The LSTM network is often used to implement the prediction of time series data in deep learning. It is a good way to detect and fit the deep rules of the data. Based on these advantages, the LSTM network is well suited as the basic predictor for the proposed module.

**6.2. Basic Predictor Based on LSTM.** As shown in Figure 8, both channels of CPVp-LSTM implement prediction functions through the LSTM network. The LSTM network is often used in the prediction of time series signals. It can well detect and fit to the in-depth features of the dataset. Based on these advantages, the LSTM network is well suited as the basic predictor for the proposed algorithm. Suppose that the time series of the user's viewport can be expressed by  $\{\vec{V}_{t_1}, \vec{V}_{t_2}, \dots, \vec{V}_{t_{n-1}}, \vec{V}_{t_n}\}$ .  $\vec{V}_{t_i}$  represents the viewport coordinates of the user at time  $t_i$ . The core function of the basic predictor is to calculate  $\vec{V}_{t_{i+M}}$  from  $\vec{V}_{t_{i-N}:t_i}$  with LSTM networks,  $N$  is the length of the input sequence, and  $M$  is the length of the predicted interval. The historical viewport coordinate sequence from time  $t_{i-N}$  to time  $t_i$  is used to predict the position of the viewport at the time  $t_{i+M}$  in the future.

The proposed basic predictor contains two hidden layers and three LSTM layers, as shown in Figure 9. The rectified

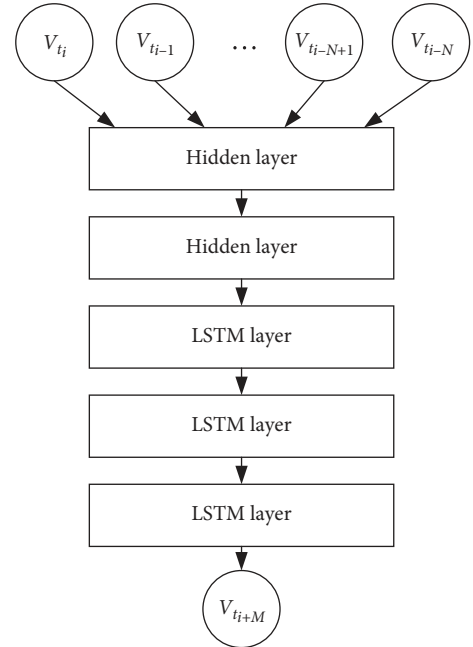


FIGURE 9: LSTM network of basic predictor.

linear unit (ReLU) activation function is used after the hidden layer to enhance nonlinearity. The LSTM layer is composed of  $N$  LSTM units. Each unit generates two values simultaneously; one is the output of the current unit, and the other one is the collection of memory information from all previous units. Both of these two output values will be sent into the next unit as the input, so the LSTM layer can be memorable. The loss function is modified based on cross entropy, which is used to update various parameters of the network during each iteration of the training. The user's viewport position  $\vec{V}_{t_i}$  can be described by its Euler angle

coordinates, which includes 3 degrees of freedom, pitch, yaw, and roll (i.e.,  $X$ ,  $Y$ , and  $Z$  angles).  $X$  and  $Y$  angles are within  $-90^\circ \sim 90^\circ$  and  $-180^\circ \sim 180^\circ$ , respectively. In 90% of time,  $Z$  angles are within  $-11^\circ \sim 9^\circ$ . Based on this special range of values for viewport coordinates, we define an improved cross entropy loss function  $L$ . Its definition of the  $Y$  component is shown as equations (7) and (8).  $T_h$  is a threshold used to determine whether an out-of-bounds condition has occurred, which is generally set to a default value of 10.  $Y$  is the predicted output and  $\hat{Y}$  is the actual value:

$$\hat{Y} = \begin{cases} Y - 360 & \text{where } 0 < 360 - (Y - \hat{Y}) < T_h, \\ Y + 360 & \text{where } 0 < 360 + (Y - \hat{Y}) < T_h, \\ Y & \text{otherwise.} \end{cases} \quad (7)$$

After normalizing  $\hat{Y}$  and  $\hat{Y}$ ,

$$L = -[\hat{Y} \log(\hat{Y}) + (1 - \hat{Y}) \log(1 - \hat{Y})]. \quad (8)$$

The cross entropy definition of the  $X$  component is similar to the  $Y$  component. Due to the small distribution range of the  $Z$  component, there is no out-of-bounds condition in most cases, so the cross entropy of the  $Z$  component does not change.

In CPVp-LSTM, the predictors used in the two channels are similar in structure, but the size and some parameters of each layer are adjusted according to the difference between the input sequence.

**6.3. Prediction Model Based on User Viewing Habits.** The difference in viewing habits between different users is enormous, which needs to be fully considered when making viewport predictions based on personal historical data. We use the user's ID as an index to create a separate viewport trajectory database for each user. The database will contain historical viewport data for all OVs that the user has viewed. Since the user's behavioral habit information is mainly included in the relative movement of the user's viewport (slow or fast) and is not closely related to the absolute position of the user's viewport, we extract the differential data of the user's viewport trajectory and send them to the LSTM network for training.

At time  $t_i$ , its difference value can be obtained by the following formula:

$$D_{t_i} = V_{t_i} - V_{t_{i-1}}, \quad (9)$$

where  $V_{t_i}$  is the current viewport coordinate at time  $t_i$  and  $V_{t_{i-1}}$  is the last coordinate at time  $t_{i-1}$ . The LSTM network finally obtains the predicted value  $D_M$  of the viewport coordinate change amount, and the final output result of channel 1 is  $V_{t_i+M} = V_{t_i} + D_M$ .

**6.4. Prediction Model Based on ROI Perception of OV Content.** Inspired by some existing viewport prediction schemes, they are able to improve the accuracy of prediction by acquiring regions of interest (ROI) in OV frames. This type of method first locates the ROI by performing image feature

extraction on each frame that is predecoded and then simultaneously sends the ROI coordinates into the prediction model along with the viewpoint coordinates acquired by the sensor of the display terminal. The ROI information of every frame can effectively improve the accuracy of the prediction model, but this operation of predecoding and extracting features is very expensive in terms of resource consumption of most display devices.

In this paper, we consider that the information of this ROI should also be included in the time series of viewpoint coordinates. When a frame of the OV has an ROI that attracts the attention of most users, the user's viewport position should tend to converge at this moment. In order to get that ROI information, we cluster the set of viewpoint coordinates of each frame in one OV. These viewpoint data are collected from all users when they independently watch this OV. Because the number of ROIs contained in one frame cannot be predetermined, the DBSCAN (density-based spatial clustering of applications with noise) algorithm is used for clustering. DBSCAN can automatically determine the number of clusters by specifying the distance between members and the maximum boundary of the cluster.

Figure 10 shows the analysis results of two typical frames. The left side of Figure 10(a) is the picture of the OV frame, and the right side is the clustering result of the frame viewport coordinates of this frame. It can be seen that most of the points are clustered to cluster-1, which are colored yellow. The remaining isolated points are shown in blue and their number is too small to be grouped together. The area indicated by the yellow box in the OV frame on the left corresponds to cluster-1 in the clustering result. It can be clearly observed that the concentration of the viewport at this time is due to the presence of the diver in the area of the yellow box. Similarly, the cluster-1 of the clustering results in Figure 10(b) is caused by the diver in the yellow box of the OV frame, and the cluster-2 is caused by the underwater wreckage of the green box.

Because channel 2 mainly refers to the information of the absolute coordinates of the user's viewport, the  $V_t$  sequence is directly used as the input of the predictor. At the same time, we introduce the clustering results of each frame into the prediction model to improve accuracy. In actual deployment, after the viewport prediction module collects the viewport data from different users according to the OV ID, the clustering operation can be completed with only a small amount of resources. Channel 2 will directly output the predicted viewport coordinates.

## 7. Implementation and Evaluation

In this section, we will show the implementation of the MELiveOV prototype system and discuss the performance of it.

**7.1. Experimental Prototype System.** Figure 11 shows the capture device of the prototype system. It consists of a

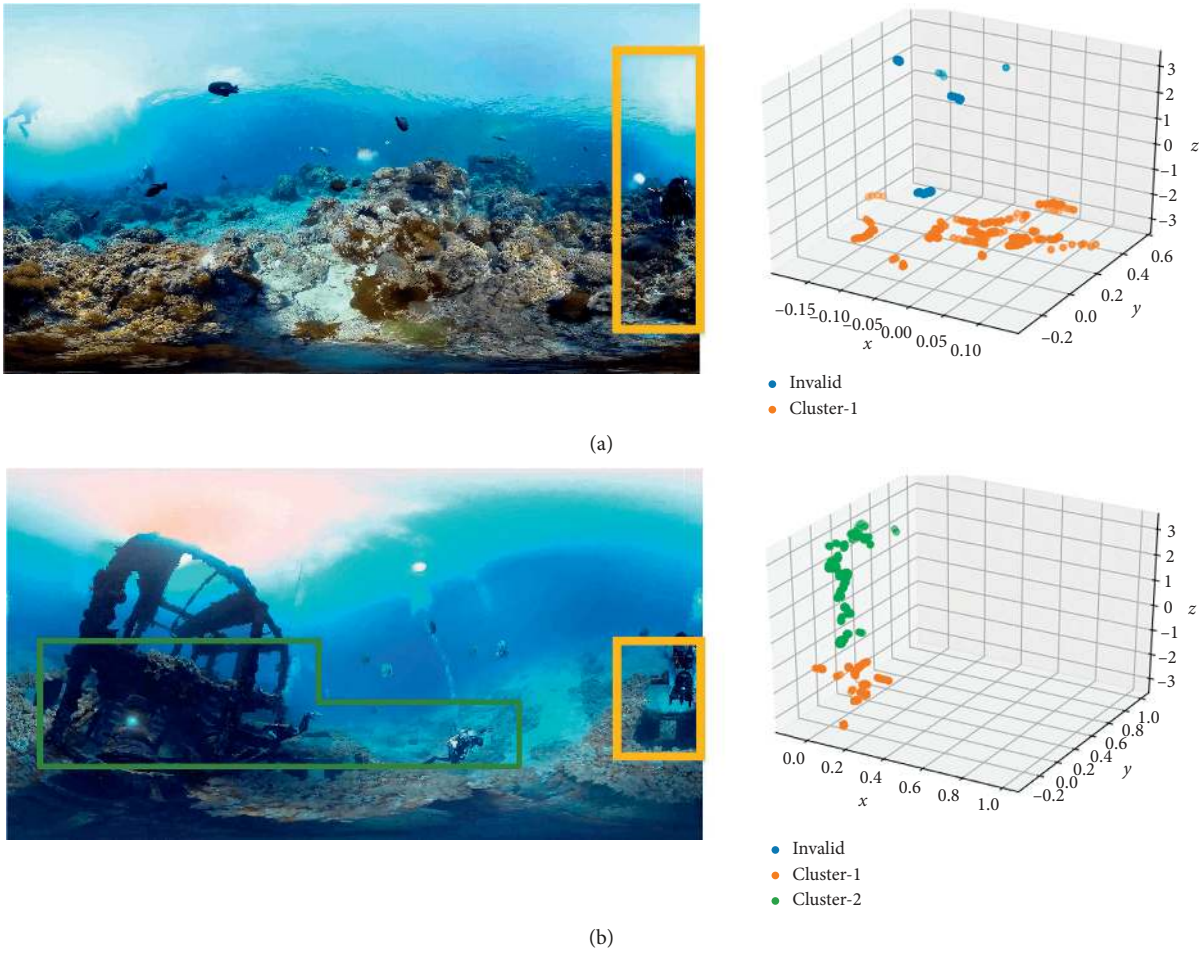


FIGURE 10: Cluster result of OV diving.

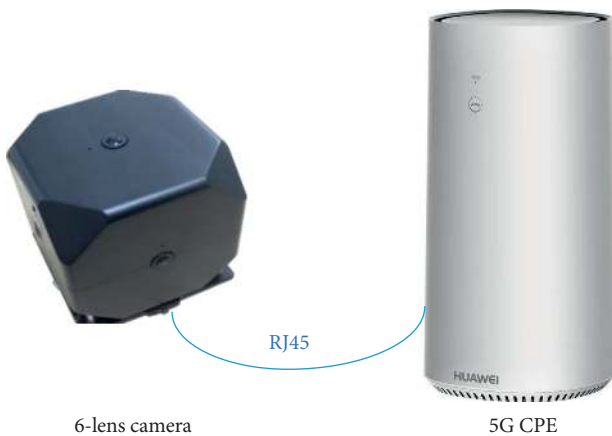


FIGURE 11: Omnidirectional capture device.

customized omnidirectional camera with 6 lenses that can simultaneously capture video data in 6 directions (up, down, left, right, front, and back) and a 5G CPE. They communicate through the RJ45 network ports. The structure of the customized camera is shown in Figure 12. We use HiSilicon’s Hi3559AV100 as the control board, which is responsible for collecting all the original lens data and generating

standardized video sequences. Data are transmitted between lens and control board through MIPI interface.

Our prototype system also includes two edge servers, as shown in Figure 13. The edge server consists of a 5G small cell and a regular server. The regular server has an Intel(R) Xeon(R) CPU E5-2630 v4 and six GTX 1080TI 11G; the size of the server is 32G. We modified the forwarding strategy of the 5G small cell so that after the data arrives, it will be processed by the server before forwarding. There are two sets of such edge servers, one as the first-mile edge server and the other as the last-mile edge server. Communication between them is achieved through a virtual core network inside the lab.

On the display terminal, the prototype system supports access to multiple heterogeneous playback devices. Such as Android phones, PC, and HMD. We have designed dedicated player software on each platform to implement the functionality of the viewpoint prediction module. All player software can collect the user’s viewpoint data with the sampling frequency of 30 Hz.

As presented in Figure 14, the prototype system of MELiveOV implements the end-to-end live streaming service of OV. The left part of Figure 14 is the picture inside the FOV, which can be seen by the user on the display terminal

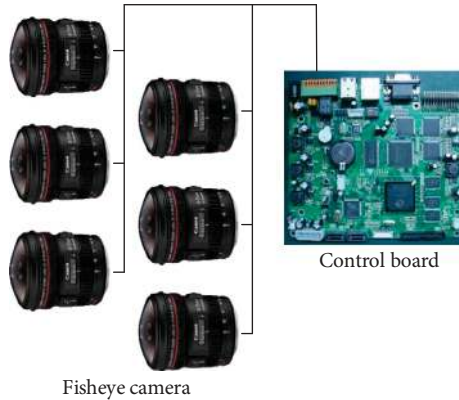


FIGURE 12: Structure of customized camera.

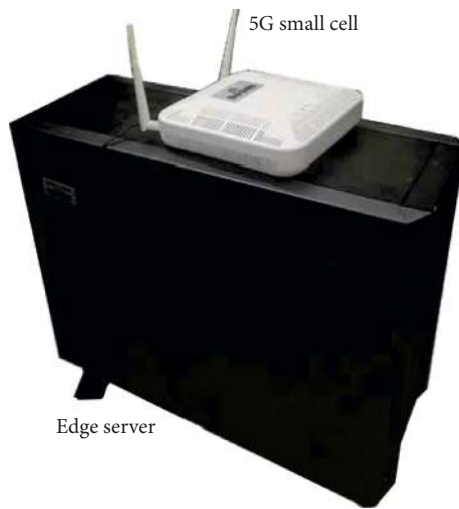


FIGURE 13: 5G edge server.

through the screen of the device. The upper right part of the figure shows the actual situation of the user watching the OV live streaming through the Android phone. The lower right part of the figure shows the working scene of the capture device of MELiveOV. As shown in the figure, we placed the omnidirectional camera on a handcart with power supply, and the camera communicates with the 5G small cell of the edge server over the wireless network.

**7.2. Experimental and Evaluation Results.** In this subsection, we tested the MELiveOV prototype system in different scenarios and analyzed the system performance. As shown in Figure 15, we conducted four experiments of OV live streaming in the *Playground*, *Road*, *Office*, and *Night* scenes. We collected data about the video quality and network bandwidth consumption of MELiveOV in four sets of experiments.

The overall resolution of OV in all four scenarios is around 4k (the resolution of the OV panoramic frame is not fixed due to the two-layer transmission scheme) and the frame rate is 25 fps. Besides, we used FFMPEG as our coding tool and H.264/AVC as our coding standard. The PSNR of

the OV picture is shown in Figure 11 during the live streaming. In Figure 16, we used PSNR (peak signal to noise ratio) to evaluate the picture quality during OV live streaming. The red column represents the quality of the video picture within the user's FOV, and the yellow column represents the quality of the non-FOV area. In the *Night*, the quality of the OV is relatively high because the picture content is relatively simple (mainly black) and the camera is fixed. In the *Road*, the camera is moving and there are too many objects (buildings and trees) in the scene, so the PSNR is the worst. The results of *Playground* and *Office* are more common. MELiveOV can guarantee that the PSNR of the user's FOV in OV live streaming is about 50 dB. At the same time, we can also ensure that the PSNR of non-FOV areas is maintained above 30 dB. When the user's viewpoint trajectory is predicted to be wrong, MELiveOV can still avoid image incompleteness in the user's field of view.

Figure 17 is analyzed with SSIM (structural similarity index) as a quality evaluation indicator. The results show that MELiveOV can also achieve better performance on SSIM, the quality of the FOV region is maintained above 0.98, and the non-FOV region is around 0.9.

We have verified the reliability of the picture quality of MELiveOV during live streaming. Next, we will show the network bandwidth situation of the MELiveOV. We set up a comparison system that puts the omnidirectional video generation task on the central cloud server (which is a cloud server leased on the public network). The comparison system does not include the streaming module of the last-mile edge server and the prediction module of the display terminal. It can only implement the most basic OV live streaming function. The results of the network bandwidth consumption experiment are shown in Table 1. We can see that in all scenarios, MELiveOV can save about 50% of the bandwidth demand, which can effectively reduce the transmission pressure of the network.

In terms of transmission delay, we also compared the two sets of schemes. The results are shown in Table 2. We can see that the service request during the OV live streaming can be responded in time due to the introduction of the MEC architecture. MELiveOV's average transmission delay can be reduced by 70% to 80%, which greatly enhances the real-time performance of OV live streaming. It can also be seen from the table that in the case of indoor scenes and fixed cameras, the transmission delay of the system is small. When the camera is outdoors and moving, the overall system latency rises significantly. We believe this is mainly due to the limited transmit power of the 5G small cell we used in the experiment. By the way, we noticed that the comparison system also achieved good latency performance in night scenes. This is mainly because of the fewer network users at night. And the network condition is better, so the transmission delay is significantly improved.

## 8. Conclusion and Future Work

In order to meet the needs of omnidirectional video (OV) live streaming services, this paper proposes a mobile edge assisted live streaming system for omnidirectional video

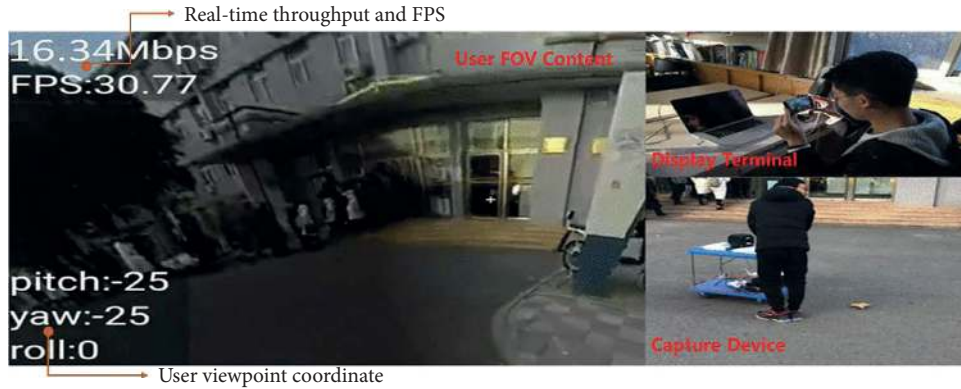


FIGURE 14: Presentation of MELiveOV prototype system.

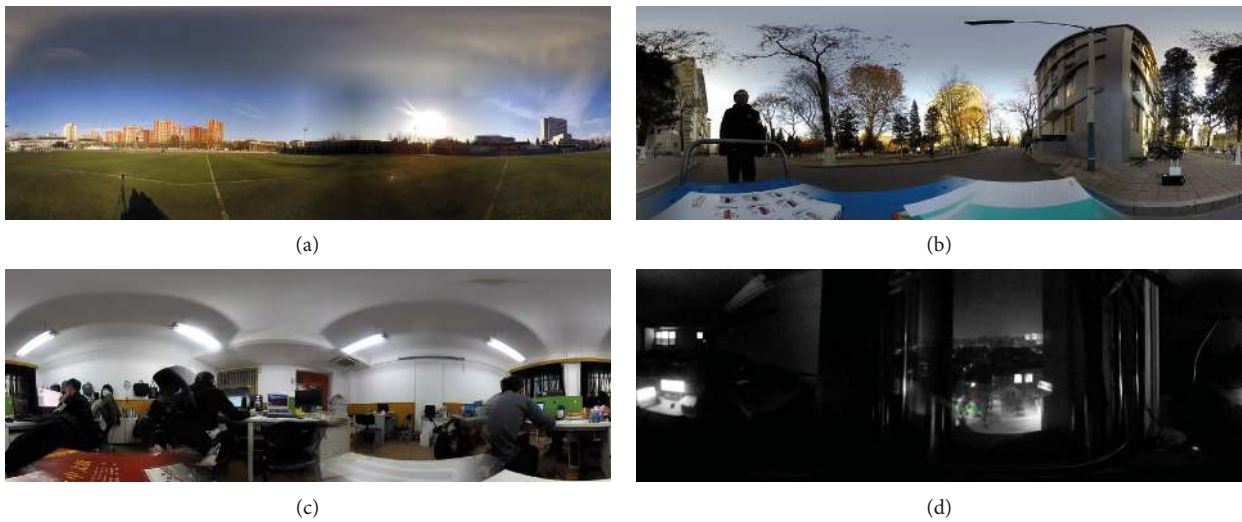


FIGURE 15: MELiveOV in different scenarios: (a) playground; (b) road; (c) office; (d) night.

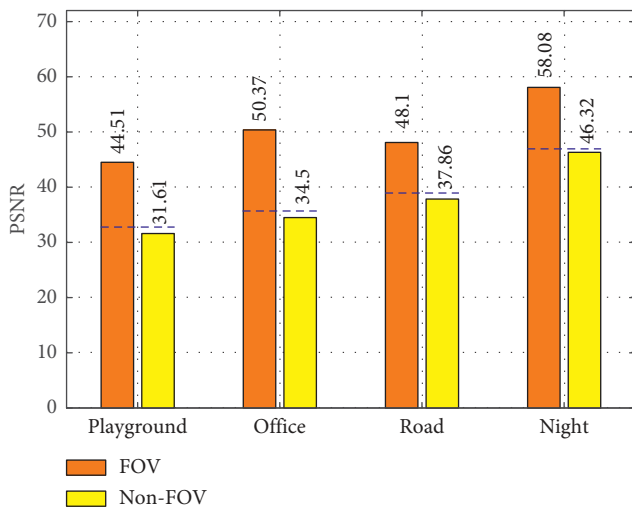


FIGURE 16: PSNR of OV during live streaming.

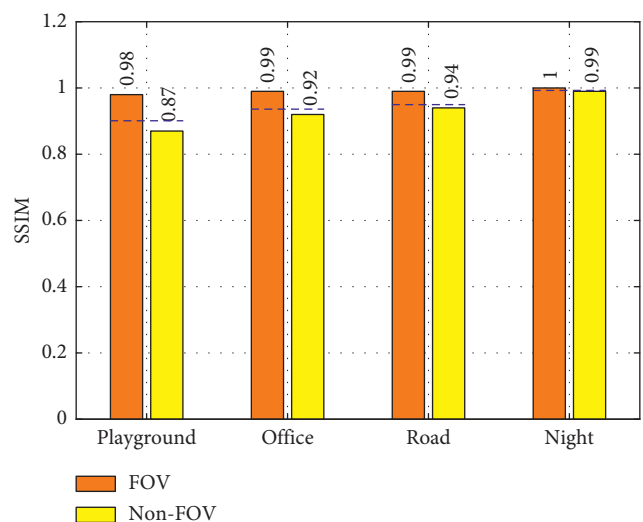


FIGURE 17: SSIM of OV during live streaming.

(MELiveOV). Enabled by the 5G edge servers with abundant computing resources, MELiveOV can offload the computational OV stitching tasks to the edge and introduce more complex prediction algorithms to optimize live streaming

performance. An end-to-end prototype system was built, and a complete service chain from capture to display for OV live streaming was implemented. The results of the

TABLE 1: Average bandwidth consumption of OV live streaming system.

Scenarios	Unoptimized streaming scheme (Mbps)	MELiveOV (Mbps)	Decrease ratio (%)
Playground	35.52	15.10	57.5
Road	38.88	16.64	57.2
Office	29.20	14.48	50.4
Night	22.48	11.52	48.8

TABLE 2: Average transmission delay of OV live streaming system.

Scenarios	Unoptimized streaming scheme (seconds)	MELiveOV (seconds)	Decrease ratio (%)
Playground	22.7	6.7	70.5
Road	26.4	5.2	80.3
Office	13.5	2.8	79.3
Night	8.7	2.4	72.4

evaluation experiment show that MELiveOV can reduce the network bandwidth requirement by about 50% and the transmission delay of more than 70% under the premise of ensuring the picture quality of viewers.

There are still many problems to be solved in the research of OV live streaming. For example, cameras may switch between multiple 5G base stations during long-distance movement. It is very important to design reliable mechanisms to ensure seamless migration of computational tasks between different edge servers. And how to achieve resource scheduling and data fusion in multiuser scenarios is also one of our future research directions. To conclude, 5G MEC is a promising solution and can well meet the needs of high-resolution OV live streaming services.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

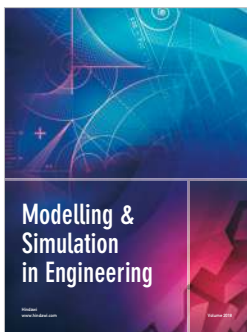
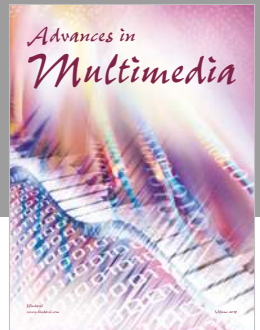
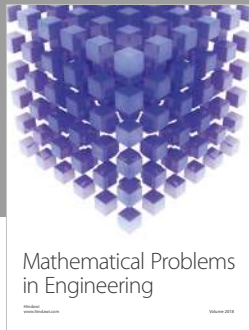
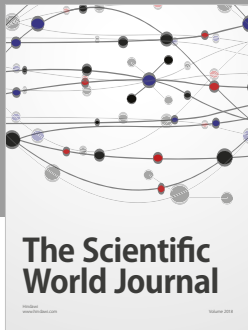
## Acknowledgments

This work was supported by the BUPT Excellent Ph.D. Students Foundation (CX2019102). This work was also supported by the project "Stereoscopic Coverage Communication Network Verification Platform for China Sea" (PCL2018KP002).

## References

- [1] Intel Ovum, *5G Economics of Entertainment Report*, Technical Report, 2018.
- [2] S. Oh and S. Hwang, "OMAF: generalized signaling of region-wise packing for omnidirectional video," in *Proceedings of the 118th MPEG Meeting of ISO/IEC JTC1/SC29/WG11, MPEG2017/m40423*, 2017.
- [3] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pp. 99–114, ACM, New Delhi, India, October 2018.
- [4] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "OpTile: toward optimal tiling in 360-degree video streaming," in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 708–716, ACM, Mountain View, CA, USA, October 2017.
- [5] C. Ozcinar, A. De Abreu, and Aljosa Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 2174–2178, IEEE, Beijing, China, September 2017.
- [6] R. Skupin, Y. Sanchez, Y.-K. Wang, M. M. Hannuksela, J. Boyce, and W. Mathias, "Standardization status of 360 degree video coding and delivery," in *Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, IEEE, St. Petersburg, FL, USA, December 2017.
- [7] D. V. Nguyen, H. T. T. Tran, A. T. Pham, and T. Cong Thang, "An optimal tile-based approach for viewport-adaptive 360-degree video streaming," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 29–42, 2019.
- [8] H. Ahmadi, O. Eltobgy, and H. Mohamed, "Adaptive multicast streaming of virtual reality content to mobile users," in *Proceedings of the on Thematic Workshops of ACM Multimedia*, pp. 170–178, ACM, Mountain View, CA, USA, October 2017.
- [9] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.
- [10] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using MPEG-DASH," in *Proceedings of the 7th International Conference on Multimedia Systems, MMSys '16*, pp. 41:1–41:3, New York, NY, USA, 2016.
- [11] Z. Liu, S. Ishihara, Y. Cui, Y. Ji, and Y. Tanaka, "Jet: joint source and channel coding for error resilient virtual reality video wireless transmission," *Signal Processing*, vol. 147, pp. 154–162, 2018.
- [12] C. Griwodz, M. Jeppsson, H. . Espeland et al., "Efficient live and on-demand tiled hevc 360 VR video streaming," in *Proceedings of the 2018 IEEE International Symposium on Multimedia (ISM)*, pp. 81–88, IEEE, Taichung, Taiwan, 2018.
- [13] X. Kelvin Zou, J. Erman, V. Gopalakrishnan et al., "Can accurate predictions improve video streaming in cellular networks," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 57–62, ACM, Santa Fe, NM, USA, February 2015.
- [14] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 965–984, 2018.
- [15] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2018.
- [16] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, "Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 1988–2001, 2015.
- [17] Y. Sanchez, G. Singh Bhullar, R. Skupin, C. Hellge, and T. Schierl, "Delay impact on mpeg OMAF's tile-based

- viewport-dependent 360° video streaming,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 18–28, 2019.
- [18] L. Sun, F. Duanmu, Y. Liu et al., “A two-tier system for on-demand streaming of 360 degree video over dynamic networks,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 43–57, 2019.
- [19] Z. Chen, Y. Li, and Y. Zhang, “Recent advances in omnidirectional video coding for virtual reality: projection and evaluation,” *Signal Processing*, vol. 146, pp. 66–78, 2018.
- [20] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications,” in *Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM)*, pp. 583–586, IEEE, San Jose, CA, USA, December 2016.
- [21] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, “Tile based HEVC video for head mounted displays,” in *Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM)*, pp. 399–400, IEEE, San Jose, CA, USA, December 2016.
- [22] M. Yu, H. Lakshman, and B. Girod, “A framework to evaluate omnidirectional video coding schemes,” in *Proceedings of the 2015 IEEE International Symposium on Mixed and Augmented Reality*, pp. 31–36, IEEE, Fukuoka, Japan, October 2015.
- [23] S.-H. Lee, S.-T. Kim, E. Yip, B.-D. Choi, J. Song, and S.-J. Ko, “Omnidirectional video coding using latitude adaptive down-sampling and pixel rearrangement,” *Electronics Letters*, vol. 53, no. 10, pp. 655–657, 2017.
- [24] R. Ghaznavi-Youvalari, A. Zare, H. Fang et al., “Comparison of HEVC coding schemes for tile-based viewport-adaptive streaming of omnidirectional video,” in *Proceedings of the 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSp)*, pp. 1–6, IEEE, London, UK, October 2017.
- [25] I. D. D. Curcio, H. Toukoma, and D. Naik, “Bandwidth reduction of omnidirectional viewport-dependent video streaming via subjective quality assessment,” in *Proceedings of the 2nd International Workshop on Multimedia Alternate Realities*, pp. 9–14, ACM, CA, USA, August 2017.
- [26] M. Yu, H. Lakshman, and B. Girod, “Content adaptive representations of omnidirectional videos for cinematic virtual reality,” in *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, pp. 1–6, ACM, Brisbane, Australia, October 2015.
- [27] M. Graf, C. Timmerer, and C. Mueller, “Towards bandwidth efficient adaptive streaming of omnidirectional video over http: design, implementation, and evaluation,” in *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 261–271, ACM, Taipei, Taiwan, June 2017.
- [28] J. Li, Z. Wen, S. Li, Y. Zhao, B. Guo, and J. Wen, “Novel tile segmentation scheme for omnidirectional video,” in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 370–374, IEEE, Phoenix, AZ, USA, 2016.
- [29] S. Gudumasu, H. Ahmed, Y. He, and Y. Ye, “A sub-picture-based omnidirectional video live streaming platform,” in *Proceedings of the Applications of Digital Image Processing XLI*, vol. 10752, p. 1075234, International Society for Optics and Photonics, San Diego, CA, USA, August 2018.
- [30] C. Ozcinar and A. Smolic, “Visual attention in omnidirectional video for virtual reality applications,” in *Proceedings of the 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, Berlin, Germany, September 2018.
- [31] A. Ninan and C. Atluru, “View direction based multilevel low bandwidth techniques to support individual user experiences of omnidirectional video,” US Patent App. 15/842,703, 2018.
- [32] R. Ghaznavi-Youvalari and A. Aminlou, “Geometry-based motion vector scaling for omnidirectional video coding,” in *Proceedings of the 2018 IEEE International Symposium on Multimedia (ISM)*, pp. 127–130, IEEE, Taichung, Taiwan, 2018.
- [33] J. Le Feuvre and C. Concolato, “Tiled-based adaptive streaming using MPEG-DASH,” in *Proceedings of the 7th International Conference on Multimedia Systems*, vol. 41, ACM, Wörthersee, Austria, May 2016.
- [34] Y. Sun, A. Lu, and Y. Lu, “Weighted-to-spherically-uniform quality evaluation for omnidirectional video,” *IEEE signal processing letters*, vol. 24, no. 9, pp. 1408–1412, 2017.
- [35] 3GPP, Virtual Reality (VR) media services over 3GPP (release 15), Technical Specification (TS) 26.918, 3rd Generation Partnership Project (3GPP), 06 2017, Version 1.0.0.
- [36] DVB, “DVB study mission on virtual reality (CM1706),” *Commercial Module, Digital Video Broadcasting*, DVB Project, 2016.
- [37] Y. Ye, E. Alshina, and J. Boyce, “JVET-E1003: Algorithm descriptions of projection format conversion and video quality metrics in 360 Lib,” *Joint Video Exploration Team ITU-T SG 16 WP3 ISO/IEC JTC 1/SC 29/WG 11 5th Meet*, Joint Video Exploration Team, Geneva, Switzerland, 2017.
- [38] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, “JVET common test conditions and evaluation procedures for 360 video,” in *Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-E1030*, pp. 1–6, 2017.
- [39] M. C. Potter, B. Wyble, C. E. Hagmann, and E. S. McCourt, “Detecting meaning in RSVP at 13 ms per picture,” *Attention, Perception, & Psychophysics*, vol. 76, no. 2, pp. 270–279, 2014.
- [40] G. He, J. Hu, H. Jiang, and Y. Li, “Scalable video coding based on user’s view for real-time virtual reality applications,” *IEEE Communications Letters*, vol. 22, no. 1, pp. 25–28, 2018.
- [41] F. Duanmu, E. Kurdoglu, Y. Liu, and Y. Wang, “View direction and bandwidth adaptive 360 degree video streaming using a two-tier system,” in *Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, Baltimore, MD, USA, May 2017.



Hindawi

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

