Hindawi
Complexity
Volume 2020, Article ID 8832715, 14 pages
https://doi.org/10.1155/2020/8832715

WILEY | Hindawi

*Research Article*

# Adaptive Panoramic Video Multicast Streaming with Limited FoV Feedback

**Jie Li**,[1] **Ling Han**,[1] **Cong Zhang**,[1] **Qiyue Li**,[2] **and Weitao Li**[2]

[1]*School of Computer and Information, Hefei University of Technology, Hefei, China*
[2]*School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China*

Correspondence should be addressed to Weitao Li; wtli@hfut.edu.cn

Received 21 August 2020; Revised 27 October 2020; Accepted 1 December 2020; Published 18 December 2020

Academic Editor: Zhile Yang



Virtual reality (VR) provides an immersive 360-degree viewing experience and has been widely used in many areas. However, the transmission of panoramic video usually places a large demand on bandwidth; thus, it is difficult to ensure a reliable quality of experience (QoE) under a limited bandwidth. In this paper, we propose a field-of-view (FoV) prediction methodology based on limited FoV feedback that can fuse the heat map and FoV information to generate a user view. The former is obtained through saliency detection, while the latter is extracted from some user perspectives randomly, and it contains the FoV information of all users. Then, we design a QoE-driven panoramic video streaming system with a client/server (C/S) architecture, in which the server performs rate adaptation based on the bandwidth and the predicted FoV. We then formulate it as a nonlinear integer programming (NLP) problem and propose an optimal algorithm that combines the Karush–Kuhn–Tucker (KKT) conditions with the branch-and-bound method to solve this problem. Finally, we evaluate our system in a simulation environment, and the results show that the system performs better than the baseline.

## 1. Introduction

Virtual reality (VR) technology has recently become increasingly important with the rise in demand for interactive applications. As one of the most important applications of VR, 360-degree video has become increasingly popular. With the help of a head-mounted display (HMD), users can freely adjust their heads to change the direction of the view, which provides an extraordinary immersive experience and a sense of depth in all directions. However, the motion-to-photon latency requirement for 360-degree video must be less than 20 ms [1]; otherwise, visually induced motion sickness (VIMS) will occur [2]. From the network perspective, the transmission of 360-degree video requires substantial bandwidth resources. There is a very large gap between the bandwidth capacity of traditional wireless technologies and the bandwidth demand of 360-degree video streaming. For example, the bandwidth demand is at least 4.2 Gbps when streaming 360-degree video with 4K resolution at 120 frames per second (FPS) [3], which greatly exceeds the current wireless (e.g., cellular) network bandwidth. Therefore, it is very difficult to transmit 360-degree video with full-quality streaming at 4K or higher resolution. This challenge is more serious for real-time 360-degree video streaming.

Due to the limitations of the display device, users can watch only approximately 12%−15% of the whole 360-degree video by wearing an HMD, which is commonly called the FoV [4]. Therefore, transmitting whole 360-degree content, as is the strategy of YouTube [5], results in a large waste of bandwidth and computing resources. One of the most popular methods for 360-degree video transmission is tile-based adaptive transmission, which divides 360-degree video frames into multiple tiles in space and encodes the tiles into multibitrate segments. Then, those tiles covering the users' FoV are transmitted with high quality, while those tiles outside the FoV are not transmitted or are transmitted but with lower quality. The two most important schemes in tile-based adaptive transmission are rate adaptation and viewport adaptation [6–8].

The rate adaptation methods focus on avoiding possible degradation of video quality or stalling under harsh network conditions for 360-degree video streaming. Considering the dynamic and uncertain effects of FoV and network conditions, it is important to make a decision between the video quality, bandwidth efficiency, and FoV switching latency. The goal of rate adaptation is to investigate adaptively controlling the video bitrate to achieve the optimal QoE for 360-degree video streams. For instance, in [6], the authors developed different QoE evaluation criteria and designed tile-based adaptive 360-degree video streaming to maximize the user QoE.

The viewport adaptation is used to adapt to a user's head movement by predicting the FoV. Since a user's FoV can only cover part of the full 360-degree scene at any given moment, it is possible to take a tile-based partial delivery approach by predicting the user's FoV in advance to transmit only the relevant tiles. However, due to the randomness and uncertainty of head movements, the accuracy of FoV prediction may be poor. Transmitting only the predicted FoV area may result in the viewer's real FoV not covering any content, which will greatly affect his or her QoE. Therefore, to improve the user QoE, non-FoV areas usually also need to be transmitted with lower quality. When the client requests video tiles, the server should select a suitable quality level of the video tiles for the next moment in real time. The quality level of the video tiles covering the entire 360-degree scene is randomly assigned and transmitted if the FoV prediction cannot be performed in real time, which may cause the video tile that the user wants to watch to be of low quality, thereby affecting the user's QoE.

For adaptive transmission, it is important to upload the real-time FoV information of users through the uplink channel, which can help the video server select the optimal representation transmission in a 360-degree video streaming system. However, in some video multicast applications such as VR theater, each user can choose the angle of interest to view a 360-degree video by wearing an HMD and request video tiles wirelessly. Since the channels of the current communication system are asymmetric and the bandwidth of the uplink channel is much smaller than that of the downlink channel, the uplink wireless network bandwidth cannot support the upload of FoV information for all users in real time, which could result in the feedback explosion problem [9]. At this time, only a small number of users can upload FoV information. Hence, it is a challenge to show how to use only a small amount of FoV information for optimal multicast video transmission.

The challenges mainly arise from two aspects: First, it is difficult to accurately predict a user's FoV. For instance, [10, 11] proposed utilizing the current FoV of a user to represent his or her future FoV, but their prediction horizon is very short. Moreover, the method requires uploading a large amount of user viewport information to ensure the accuracy of the prediction. Then, it is difficult to design a simple yet effective rate adaptation algorithm to tackle the variation of network bandwidth. Kan et al. [12] proposed a deep reinforcement learning (DRL)-based rate adaptation algorithm to maximize the user QoE. DRL offers a good performance, but its complexity is very high, and its calculation speed is slow.

Considering that humans share similar watching behaviors [13], each user will be attracted to the saliency area of a video, which represents the area that most users may watch. Therefore, performing saliency detection on a video can reveal the saliency areas, though this method has certain errors. We then use the FoV information of a small number of users to calibrate the saliency area to obtain more accurate viewport information of all users, which is then used as the basis for optimizing the transmission system. Thus, in this paper, a 360-degree video real-time adaptive transmission system based on limited FoV feedback is proposed. Specifically, the server extracts video saliency and combines it with limited ground-truth FoV feedback information to estimate the area most probably watched. We use a low computation fusion algorithm to perform a real-time prediction. Then, the estimation results and wireless network bandwidth are used to select the appropriate quality level transmission for each tile to maximize the QoE of all users. We define the transmission problem of the designed system as a nonlinear integer optimization problem. To solve this problem, we design an algorithm combining the KKT conditions and the branch-and-bound method.

In summary, our contributions are as follows:

(1) We propose a QoE-optimized 360-degree video streaming system with limited FoV feedback.

(2) We propose a multiuser viewport prediction method, which fuses the viewport of other users and saliency maps derived from the video sequence to obtain the FoV prediction results.

(3) We propose an optimal algorithm that combines the KKT conditions with the branch-and-bound method to solve the optimization problem.

(4) We propose a multiuser QoE-driven model to allocate the best bitrate for each tile. We simulate the QoE optimization model and conduct exhaustive experiments on a simulation testbed.

The remainder of this paper is organized as follows. Section 2 introduces the related work on 360-degree video and adaptive 360-degree video streaming. Section 3 describes the system models and formulates the optimization problem. In Section 4, we specifically describe the viewport prediction approach. Section 5 designs the KKT conditions with the branch-and-bound algorithm. A performance evaluation and comparison are presented in Section 6. Finally, the conclusions of this paper are given in Section 7.

## 2. Related Work

This section introduces the related schemes on the QoE model and 360-degree video streaming in the literature.

A 360-degree video can be recorded through a single omnidirectional camera (e.g., Samsung Gear 360), or multiple cameras can be used to record separate videos that are then combined into a 360-degree video by software aggregation and stitching. To facilitate the storage and

processing of 360-degree video from a spherical domain, 360-degree video from a spherical domain is typically projected onto a two-dimensional (2D) plane [14]. Since 360-degree video in 2D format can be reprojected onto a spherical plane by the HMD [15], the user can enjoy spherical video content by wearing the HMD. Compared with traditional video, transmitting a 360-degree video requires higher bandwidth. Hence, tile-based adaptive streaming is proposed to reduce the required transmission bandwidth and avoid possible degradation of video quality or stalling under poor network conditions.

Currently, the projection formats of 360-degree video include the cubemap projection (CMP) format, equirectangular projection (ERP) format, and truncated pyramid projection (TSP) format. Because the ERP and CMP formats outperform the TSP format, they are the most widely used in practice [16]. In this paper, the QoE-optimized transmission method proposed is based on the ERP format. After projecting the sphere onto a 2D plane in the ERP format, a 360-degree video undergoes tile-based adaptive streaming. In tile-based adaptive streaming, the server typically first temporally divides the 360-degree video with the ERP format into multiple groups of pictures (GoPs), each of which contains a certain number of video frames. The server then spatially divides each GoP into multiple tiles and encodes them at different bitrates. According to the network bandwidth and FoV, the server adaptively selects an optimal quality level for each tile and transmits it to the client. In the tile-based adaptive transmission, either viewport adaptation or rate adaptation can be carried out. The core of the former is to predict the user viewport, and the core of the latter is the allocation of resources by the server.

As the key role in viewport adaptive streaming, FoV prediction is of crucial importance. The purpose of predicting a user's future FoV is to deliver the appropriate part of the future video segment. The FoV prediction algorithms in the literature [10, 11, 17–24] can be categorized into two classes: trajectory-based and content-based. For instance, Feng et al. [11] proposed a naive prediction method that directly utilizes the user's current viewpoint to represent her/his future viewpoint, which can achieve an accuracy rate of more than 90% in a short time interval. However, when predicting the user's position in the next 2 seconds, the prediction accuracy rate quickly drops. Inaccurate FoV prediction may cause a mismatch between the content prefetched in the buffer and the content covered by the viewer's actual FoV. In [17], long short-term memory (LSTM) is used to encode the history of the FoV scan path and combines hidden state features with visual features to make predictions 1 second in advance. In [18], two deep reinforcement learning models were proposed: first an offline model is used to estimate the heat map of each frame of potential FoV based on visual features only, and then an online model is used to predict head motion based on past observed head positions and the heat maps of the offline model. A more recent work [19] treats the viewport prediction as a sequence learning problem and proposes predicting the target user's future viewport based on not only the user's own past viewport trajectory but also the future viewport locations of other users. In addition, the authors in [20, 24] proposed a cross-user learning-based approach to improve the FoV prediction accuracy by exploiting the users' similar region-of-interest when watching the same video.

To maximize the QoE under a bandwidth-constrained condition, the authors in [12, 25–30] studied adaptive 360-degree video streaming. Among them, [12, 26, 27] assumed that the user viewport is known and directly considered rate adaptation for tiles. For example, [12] proposed a DRL-based rate adaptation algorithm to maximize the user QoE by adapting the transmitted video quality to the time-varying network bandwidth. Li et al. [26] proposed a QoE-driven live system for 360-degree video, where the video server performs rate adaptation based on the uplink and downlink bandwidths and the real-time FoV information of each user. In addition, there are studies that combine viewport adaptation and rate adaptation. For instance, Xie et al. [29] proposed an adaptive-tile-based streaming media transmission system based on a probabilistic model, named 360ProbDASH, which combined viewport adaptation and rate adaptation to solve the QoE-driven optimization problem. In [15], the authors proposed an adaptive method combining viewpoint and rate, which uses Gaussian and Zipf models to predict the user's viewpoint and optimize the user's QoE using a two-stage optimization algorithm. In contrast to [15], our adaptive method combining rate and viewpoint is based on limited FoV feedback. We perform multiuser viewpoint prediction by fusing other users' viewports and saliency maps obtained from video sequences and optimize the multiuser QoE by combining the KKT conditions with the branch-and-bound method. Given the FoV and bandwidth estimation, the authors in [30] proposed server-side rate adaptation for 360-degree adaptive video streaming based on probability tile visibility.

## 3. 360-Degree Video Streaming System Overview

In this section, we introduce the application scenario, our 360-degree video real-time streaming system with limited FoV feedback, which predicts the FoV of all users and allocates downlink wireless resources based on the prediction results to maximize the overall QoE.

*3.1. System Model.* Considering the application scenario shown in Figure 1, everyone can choose the viewing angle of interest by wearing an HMD in a VR theater. The video server divides a complete 360-degree video into tiles that are encoded at different bitrates and selects the appropriate video content to transmit to the user according to the user's FoV. The video transmission is also affected by the downlink channel bandwidth. To provide the best QoE for all users, the existing streaming system assumes that all users upload FoV information in real time. However, because a large number of users are located in a small area, the uplink wireless network bandwidth cannot support the burden of all users uploading their FoVs at the same time, which greatly affects the results of transmission system optimization. To obtain

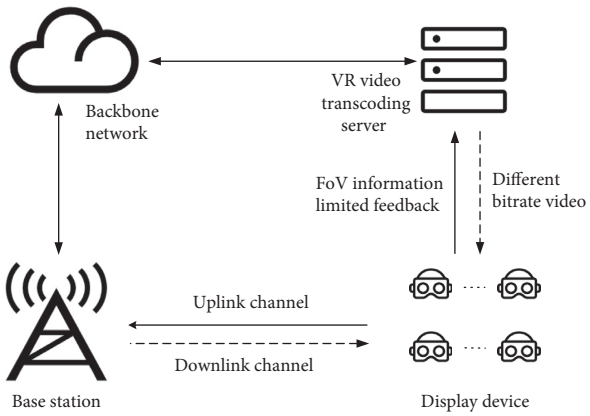


Figure 1: A typical scenario of a panoramic video streaming system.

the FoV information of all users, we first extract the saliency heat map of the video, which represents the area that most users may watch. Then, we use the ground-truth viewport of a small number of users as the calibration to improve the prediction accuracy.

In this paper, we design an adaptive 360-degree video streaming system based on limited FoV feedback. The system needs to make decisions in both the spatial domain (i.e., which tiles to acquire) and the quality domain (i.e., which qualities to acquire), which together constitute a very large search space. We first determine the tiles to be acquired based on the FoV prediction and then determine the quality level of the tiles through a rate adaptation algorithm. As shown in Figure 2, the system mainly consists of two parts: processing modules on both the server side and the client side.

The server performs a series of processes on the uploaded original 360-degree video. First, the original 360-degree video is transmitted as GoPs in time. The server spatially divides each GoP into many tiles, where each GoP consists of tiles of the same size, and then the server encodes each tile into a different quality level. The server's FoV estimation module predicts the user's viewport at each moment based on the heat map obtained by the 360-degree video saliency detection and the user's real-time FoV feedback. Finally, on the basis of the user FoV prediction, the server combines the feedback downlink wireless channel bandwidth information for rate-adaptive transmission, selects an optimal quality level for each tile, and transmits it to the client. Among them, the viewport prediction module is the most important module in the server.

The viewport prediction module extracts saliency heat maps of the 360-degree video and combines them with a limited number of user viewports via fusion to obtain more accurate information about the FoV of all users. We use a low computation fusion method, which calculates the similarity of pixel blocks through the regional covariance between the extracted heat map and the user's real-time viewport. Then, the similar pixel blocks between the heat map and the user's viewport are merged to obtain a fused image, which is the predicted FoV of all users. The computational complexity of this method is low; thus, the user's FoV can be predicted in real time.

At the client side, after a series of processes such as decoding, mapping, and rendering, the encoded tiles are finally synthesized into a complete 360-degree video and presented to the user through an HMD. In addition, the client transmits the real-time viewport information of a small number of users to the server through the uplink feedback channel, which helps in the prediction of the FoV of all users. As mentioned above, it is difficult for current asymmetric wireless communication networks to achieve real-time transmission of a large number of user viewports. To solve this problem, the client uploads only a certain amount of viewport information of each user to the server under the limitations of the bandwidth.

*3.2. Problem Formation.* Suppose that a 360-degree video is divided into $K$ GoPs (denoted by $k$) and is spatially divided into $T$ tiles (denoted by $t$). Each tile is encoded into $D$ representations (denoted by $d$) at different quality levels. When receiving a request to download tiles from the client, the server adaptively selects an optimal quality level for each tile based on the FoV prediction result and the network bandwidth. We denote the bitrates of tile $t$ with representation $d$ in GoP $k$ by $\theta_{k,t,d}$. Assume that the bandwidth when transmitting the $k$-th GoP is $B_k$.

We use the FoV prediction method described above to predict the viewports of all users in advance so that we can obtain a set of tiles covered by the predicted FoV. For user $n$, $T^n_{FoV}$ denotes the tiles covered by its FoV. To accurately model the experience of watching a 360-degree video, we use the QoE metric for user $n$ as follows, which is similar to [31]:

$$\begin{aligned}\mathrm{QoE}_n = &\sum_{k=1}^{K}\sum_{t=1}^{T^n_{FoV}}\sum_{d=1}^{D} q\left(\chi_{k,t,d}\times\theta_{k,t,d}\right) - \alpha\times\\ &\sum_{k=1}^{K}\sum_{t=1}^{T^n_{FoV}}\sum_{d=1}^{D}\chi\left(\chi_{k,t,d}\times\theta_{k,t,d} > B_k\right)\times T_k - \beta\times\\ &\sum_{k=1}^{K}\sum_{t=1}^{T^n_{FoV}}\sum_{d=1}^{D}\left(q\left(\chi_{k+1,t,d}\times\theta_{k+1,t,d}\right) - q\left(\chi_{k,t,d}\times\theta_{k,t,d}\right)\right)^2,\end{aligned} \tag{1}$$

where $\chi_{k,t,d}$ denotes the bitrate of tile $t$ with bitrate level $d$ in the $k$-th GoP. $\chi_{k,t,d}$ is a binary variable, which equals 1 if tile $t$ is transmitted with bitrate level $d$ in the $k$-th GoP and 0 otherwise. Function $q$ is a mapping function, which maps the bitrate of tile $t$ to the user's perceived quality. In this paper, we use the logarithm of the received bitrate as the form of function $q$, which conforms to the vision effect of the human eye.

The second term of (1) represents the number of stalls. Regardless of the effect of the buffers, we assume that stalling will occur when the bitrate is greater than the bandwidth $B_k$ of the $k$-th GoP and the stall time is approximately equal to the duration of the $k$-th GoP. $\chi(\chi_{k,t,d}\times\theta_{k,t,d} > B_k)$ is an indicator function, the value of which is 1 when $\chi_{k,t,d}\times\theta_{k,t,d} > B_k$ and 0 otherwise. This means that in one GoP, when

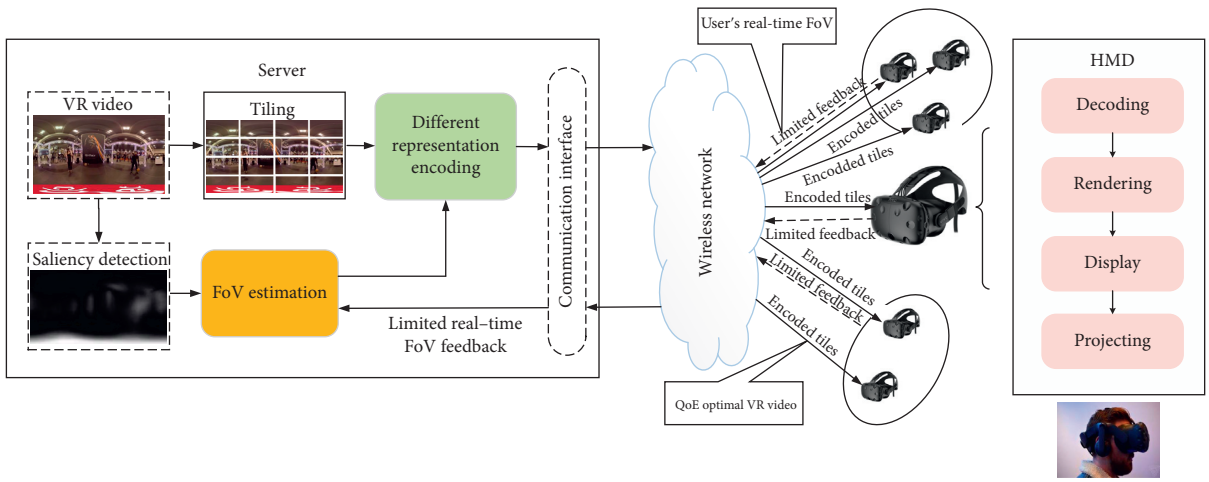


Figure 2: Panoramic video streaming system based on limited FoV feedback.

the bandwidth is less than the video bitrate, stalling will occur. $T_k$ is the duration of the k-th GoP.

The third term of (1) shows how the quality switches will impair the QoE. $(q(\chi_{k+1,t,d} \times \theta_{k+1,t,d}) - q(\chi_{k,t,d} \times \theta_{k,t,d}))^2$ considers the quality switches between the consequent GoPs. $\alpha$ and $\beta$ are nonnegative parameters used to balance the three factors.

To maximize the QoE of all users, our optimization problem is defined as follows:

$$\max \sum_{n \in N} \mathrm{QoE}_n, \tag{2}$$

s.t.

$$\sum_{d \in D} \chi_{k,t,d} = 1, \quad \forall k \in [1, K], \forall t \in [1, T^n_{FoV}], \tag{3}$$

$$\sum_{k \in K} \sum_{t \in T^n_{FoV}} \sum_{d \in D} \chi_{k,t,d} \times \theta_{k,t,d} \leq \sum_{k \in K} B_k. \tag{4}$$

Constraint (3) means that only one quality level can be selected for any video block. Constraint (4) indicates that the total bitrate of the transmitted video cannot exceed the bandwidth provided by the channel.

## 4. Viewport Estimation Based on Limited FoV Feedback

Ideally, if the head movements of all viewers during a 360-degree video session can be perfectly obtained, we can generate the best-tiled sequence that consumes the minimal bandwidth. However, due to the asymmetry of the wireless network, only a small number of users can upload viewport information in real time. Thus, we try to combine the real-time viewport information of a limited number of users with the saliency detection method of a 360-degree video to predict the most attractive area. First, we perform saliency detection on the video to obtain a video composed of saliency maps. Then, we use a small amount of user feedback FoV information to fuse the saliency maps derived from the video sequence to obtain more accurate information of all users' FoV. In addition, we combine the saliency map with the FoV feedback information via a low-computational-complexity fusion method due to its delay characteristics.

*4.1. 360-Degree Video Saliency Detection.* The human visual system (HVS) mechanism indicates that only a small region receives the most visual attention at high resolutions, while other peripheral regions receive negligible attention at low resolutions [32]. Additionally, a human's attention is more easily captured by the moving parts of objects. The goal of saliency detection is to analyze the video frames to find the areas to which users are most likely to be attracted.

To predict human attention, 360-degree video saliency detection has been extensively studied in recent years. Common deep learning (DL) methods can achieve good results in detecting the salient features of videos. In this paper, we use 360-degree video saliency detection similar to that in [33], which developed a deep neural network (DNN) architecture that combines an object-to-motion convolutional neural network (OM-CNN) and a saliency-structured convolutional long short-term memory (SS-ConvLSTM) network. A saliency heat map of each frame is generated from the 2D layers of SS-ConvLSTM, which uses the spatiotemporal features of the OM-CNN as its input. Specifically, in the OM-CNN, the objective subnet generates a cross-net mask on the features of the convolutional layers in the moving subnet. Then, the spatial features from the objective subnet and the temporal features from the motion subnet are concatenated by normalizing the proposed hierarchical features to generate the spatiotemporal features of the OM-CNN. Furthermore, taking the OM-CNN features as the input, SS-ConvLSTM leverages both the long-term and short-term correlations of the input features through the memory cells and hidden states of the 1st and 2nd LSTM layers at the last frame. Finally, the hidden states of the 2nd LSTM layer are fed into the 2D layers to generate a final saliency map. Afterward, the interframe saliency maps of a video can be generated, which take into account both the structured output with a center bias and the cross-frame transitions of human attention maps.

Because it is difficult to process a panoramic video in the spherical domain, a 360-degree video is usually projected onto a 2D plane, which causes a certain degree of distortion. The ERP projection directly uses the latitude and longitude on the sphere as the vertical and horizontal coordinates on the original frame, respectively, which results in greater distortion in the polar region of the sphere. The CMP projection maps a spherical video to an external cube. The upper and lower faces of the cube correspond to the polar regions, and the four faces in the middle correspond to the equatorial region. This solution has less distortion in the polar region of the sphere; thus, we first convert the panorama into 2D videos with the CMP projection and then perform saliency extraction with the OM-CNN and SS-ConvLSTM method. This process is illustrated in Figure 3.

*4.2. User Viewport.* To extract the FoV, the server needs to parse the viewer's head movement data collected from the HMD, usually expressed as $(X, Y, Z)$ in the Euler angle coordinate system, where $X$ is the roll angle, $Y$ is the pitch angle, and $Z$ is the yaw angle. First, we can obtain the longitude and latitude with

$$\begin{aligned}\phi &= \sin^{-1}\left(\frac{Y}{\left(X^2+Y^2+Z^2\right)^{1/2}}\right),\\ \psi &= \tan^{-1}\left(\frac{-Z}{X}\right),\end{aligned} \tag{5}$$

where $\phi$ and $\psi$ denote the longitude and latitude, respectively.

Then, the user viewport can be calculated by

$$\begin{aligned}w &= \left(\frac{\phi}{2\pi}+0.5\right)\times W,\\ h &= \left(\frac{\psi}{\pi}+0.5\right)\times H,\end{aligned} \tag{6}$$

where $W$ and $H$ are the width and height of the image, respectively.

The FoV is a rectangular area that is approximately 110 × 90 degrees (for the HTC VIVE HMD). To obtain a grayscale image, we assume that the intensity value of the viewport area follows a 2D Gaussian distribution, while the non-FoV region has an intensity value of 0.

*4.3. Viewport Estimation Based on Saliency and Limited FoV Feedback.* In this paper, we try to estimate the user viewport by fusing the saliency maps derived from the video sequence and the uploaded limited FoV information. The fusion module takes the saliency heat map and several viewport heat maps as its input. Then, the input images are divided into small regions, named pixel blocks, each containing several pixels. We can calculate the similarity of the corresponding pixel blocks based on the area covariance information and finally select similar pixel blocks to generate the estimation heat map. It is worth noting that the running time will increase as the complexity of the fusion method increases, which makes it difficult to perform real-time estimation and will impair the user experience. In this paper, we select the regional covariance features as a similarity measurement to construct a lightweight FoV estimation algorithm. The fusion process is shown in Figure 4.

First, the input image is divided into pixel blocks, each containing $m \times n$ pixels. Let $I(x, y)$ represent the intensity value of pixel $(x, y)$. Then, we select the following metrics as the features for the similarity calculation:

(1) Intensity value $(I(x, y))$ for pixel $(x, y)$. (2) Horizontal partial derivative value $(\partial I(x, y)/\partial x) = I(x+1, y) - I(x, y)$. (3) Vertical partial derivative value $(\partial I(x, y)/\partial y) = I(x, y+1) - I(x, y)$.

Then, we obtain the following statistical feature vector combinations:

$$\varnothing(I(x, y), x, y) = \left(x, y, I(x, y), \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y}\right). \tag{7}$$

For a single block $r$ with $m \times n$ pixels, we calculate the covariance matrix as follows:

$$\begin{aligned}C_r &= \frac{1}{mn-1}\sum_{x=1}^{x=m}\sum_{y=1}^{y=n}\left(\varnothing(I(x, y), x, y) - \mu\right)\\ &\quad\cdot\left(\varnothing(I(x, y), x, y) - \mu\right)^T,\\ \mu_r &= \frac{1}{mn}\sum_{x=1}^{x=m}\sum_{y=1}^{y=n}\varnothing(I(x, y), x, y),\end{aligned} \tag{8}$$

where $\mu$ represents the mean feature value of the pixel block of $\{\varnothing(I(x, y), x, y)\}_{x=1,2,\ldots,m,y=1,2,\ldots,n}$.

Based on the covariance matrix, we can calculate the Euclidean distance between two pixel blocks as their similarity metric and then use it as a basis to merge the saliency map and ground-truth feedback FoV. In this paper, we define the similarity between two pixel blocks $r$ and $s$ as

$$d(r, s) = \sqrt{\left(\mu_r - \mu_s\right)^T \times \left(C_r + C_s\right)^{-1} \times \left(\mu_r - \mu_s\right)}. \tag{9}$$

Obviously, the smaller the distance between the covariance matrices of two image pixel blocks, the more similar the two pixel blocks will be, and vice versa.

First, we cut the saliency map and FoV grayscale image into small pixel blocks, then calculate the Euclidean distance between blocks with the same position in any two heat maps, and normalize it to $[0, 1]$. If the distance is smaller than a certain threshold $T$, we consider the two pixel blocks as highly similar, meaning that this part of the video block is attractive, as measured by saliency detection software and real users. The heat maps should be merged, and we simply add together the two pixel blocks as the prediction result of the user's FoV.

## 5. QoE-Driven Rate Adaptation Algorithm

Rate adaptation of 360-degree video streaming involves two sequential procedures: viewport prediction and rate

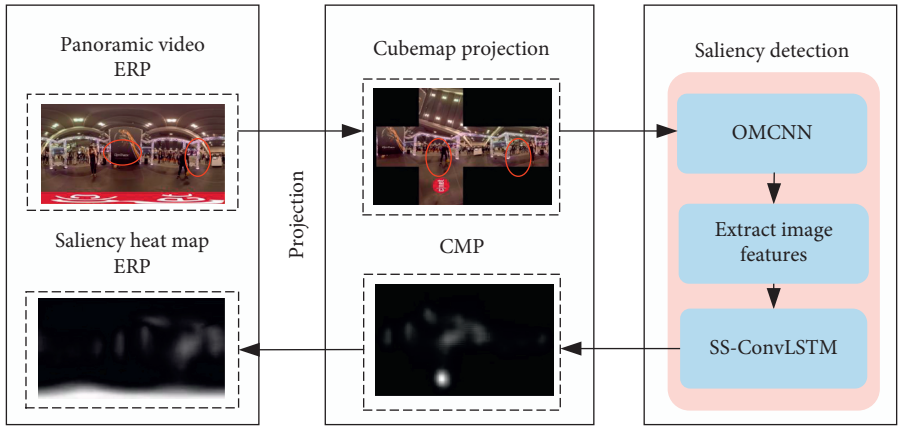


Figure 3: 360-degree video saliency detection.

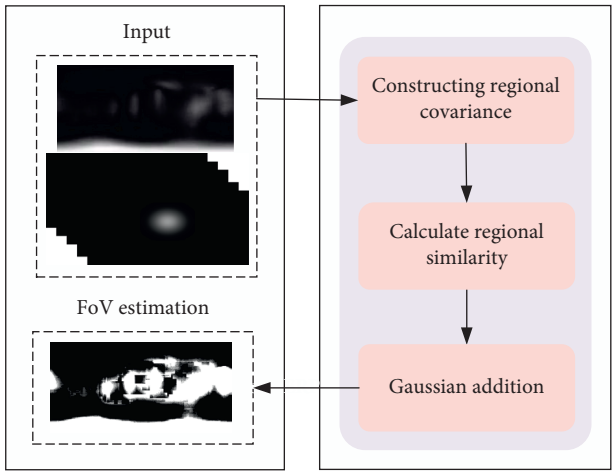


Figure 4: Viewport estimation based on saliency and limited FoV feedback.

allocation among the tiles. According to the result of FoV fusion, the area of interest of all users can be estimated. The remaining problem is how to determine the bitrate combination of tiles to improve the user QoE, which can be expressed as a QoE-driven multiuser resource allocation problem. By analyzing the QoE problem, we find that the problem is a nonlinear integer programming problem, which can be certified to be NP-hard. We approximate the indicator function in the QoE model with a logarithmic function so that the QoE function becomes a continuous function. This problem becomes a nonlinear programming problem. For general nonlinear programming problems, the KKT conditions are necessary for solving the optimal solution when the constraint satisfies the linear constraint, and the KKT conditions are also sufficient when the original problem is a convex problem, under which the optimal nonlinear problem solution can be obtained.

Since the constraint in problem (1) satisfies the linear constraint qualification, we can use the KKT conditions to solve the relaxation problem of problem (1). By using the logarithmic function to approximate the indicator function in the QoE model and relaxing the integer variables $\chi_{k,t,d}$ to continuous variables, we can solve the relaxed problem by applying the KKT conditions and the Lagrangian function. Then, we can obtain the optimal value of the original problem. Then, we use the branch-and-bound method to find the optimal solution, which meets the 0-1 constraint, to the obtained nonlinear problem.

*5.1. KKT Conditions.* First, we approximate the indicator function in the QoE model with a logarithmic function and relax $\chi_{(k,t,d)}$ to continuous variables. This problem becomes a nonlinear programming problem, which can be solved using the KKT conditions. The Lagrangian function of the problem is as follows (7):

$$L\left(\chi_{k,t,d},\lambda,\omega\right) = -\sum_{\forall n\in N} \mathrm{QoE}_n + \lambda h\left(\chi_{k,t,d}\right) + \omega g\left(\chi_{k,t,d}\right), \tag{10}$$

where

$$h\left(\chi_{k,t,d}\right) = \sum_{\forall d\in D} \chi_{k,t,d} - 1, \tag{11}$$

$$g\left(\chi_{k,t,d}\right) = \sum_{\forall k\in K} \sum_{\forall t\in T^n_{FoV}} \sum_{\forall d\in D} \chi_{k,t,d} \times \theta_{k,t,d} - \sum_{\forall k\in K} B_k. \tag{12}$$

Thus, we can obtain the relevant KKT conditions:

$$\frac{\partial L\left(\chi_{k,t,d},\lambda,\mu\right)}{\partial \chi_{k,t,d}} = -\sum_{\forall n\in N} \frac{\partial \mathrm{QoE}_n}{\partial \chi_{k,t,d}} + \lambda \frac{\partial h\left(\chi_{k,t,d}\right)}{\partial \chi_{k,t,d}} + \omega \frac{\partial g\left(\chi_{k,t,d}\right)}{\partial \chi_{k,t,d}} = 0, \tag{13}$$

$$\lambda \neq 0, \tag{14}$$

$$\omega >= 0, \tag{15}$$

$$h\left(\chi_{k,t,d}\right) = 0, \tag{16}$$

$$g\left(\chi_{k,t,d}\right) <= 0, \tag{17}$$

$$\omega g\left(\chi_{k,t,d}\right) = 0. \tag{18}$$

Among them, (12) is a necessary condition brought about when taking the extreme value of the Lagrange function, (13) is the Lagrange coefficient constraint (equation case), (14) is the inequality constraint case, (15) and (16) are the original constraint conditions, and (17) is the complementary relaxation condition. According to the KKT condition solving theorem, if the original problem is a convex problem, then the solutions that satisfy the KKT conditions are also the optimal solutions of the original problem. Thus, by combining (12)–(17) related to the KKT conditions, the optimal solution of the relaxed nonlinear problem can be obtained, which is denoted by $\chi_{\text{relax}}$, and then the branch-and-bound method can be used to find the 0-1 variable solution.

*5.2. Branch-and-Bound Method.* The branch-and-bound algorithm designed to solve problem P1 is shown in Algorithm 1. The initial inputs are $\chi_{\text{relax}}$ and $G_{\text{relax}}$, where $\chi_{\text{relax}}$ represents the solution to the corresponding relaxation problem solved using the KKT conditions. $G_{\text{relax}}$ represents the value of the corresponding optimal objective function. The output is the solution to the 0-1 variable $\chi_{0-1}$ and the corresponding optimal objective function value $G_{0-1}$.

## 6. Simulations and Experiments

To evaluate the performance of the proposed system, we perform exhaustive simulations and experiments.

*6.1. FoV Estimation.* We downloaded 4 360-degree videos from YouTube, each with at least 4K resolution (3840 pixels wide) and 24 frames per second. The duration of each video ranges from 20 to 40 seconds. Table 1 summarizes the content features of the four videos. We used an HTC VIVE as our HMD to play the 360-degree video clips to capture the head movement data of each viewer. 15 students were recruited to participate in the experiment. All participants reported normal or corrected-to-normal vision in the HMD setting. Each participant watched these four videos in a row. Video scenes were played sequentially with the starting point fixed (0° in latitude and 180° in longitude). The HTC VIVE was recalibrated before the next participant watched the video. During the experiment, the VIVE cinema open-source player collected the head orientation data (including the pitch, yaw, and roll) from built-in sensors at a high sampling rate (approximately 200 Hz). When watching the videos, each participant was free to explore the entire video scene.

To validate the effectiveness of our prediction method (denoted as VP), we estimated the users' viewports by fusing the saliency maps obtained from the video sequence and the uploaded limited FoV information. We first performed saliency detection on the original video to obtain a video composed of saliency maps, as shown in Figure 5. Then, we used a small amount of user FoV information at time $\tau$ to fuse the saliency map at time $\tau + \tau_0$ obtained from the video sequence to obtain the FoV of all users at time $\tau + \tau_0$. Among them, $\tau_0$ represents the interval of time from time $\tau$ used to predict the future. We chose four sets of interval time $\tau_0 = \{0.2\,\text{s}, 0.5\,\text{s}, 1\,\text{s}, 2\,\text{s}\}$ to perform FoV prediction on the entire video, similar to the procedure in [11]. To verify the performance of this prediction method, we compared it with the following two baseline methods: (1) an algorithm that does not consider the user viewport but performs only saliency detection on the video (denoted as Saliency Only) and (2) an algorithm that does not incorporate video saliency detection but only averages the user viewports (denoted as Viewport Only).

*6.1.1. Tile Overlap Ratio.* The heat-map-based FoV prediction method is intended for tile-based streaming systems where the client can request multiple tiles based on the predicted FoV heat map. Therefore, we used the tile overlap ratio as a performance indicator to evaluate the heat-map-based methods [19]. First, we determined the total number of tiles with nonzero values in the actual user FoV heat map. Next, we determined that the predicted heat map has a nonzero tile and determined the nonzero tile that predicted the overlap of the FoV heat map with the actual user FoV heat map. The ratio of the total number of nonzero overlapping tiles to the total number of nonzero tiles in the actual user FoV heat map is the tile overlap ratio.

The experiment first analyzed the performance of the prediction method under different prediction time intervals. Figure 6 shows the accuracy of the prediction method for 10 consecutive predictions of 0.2 s in the future. It can be seen from the figure that in the 10 predictions, the accuracy of our prediction method was higher than that of the two baseline methods.

In Figures 7(a) and 7(b) we compare the performance of the prediction methods under different time conditions. From the figure, we find that as the prediction time interval increases, the tile overlap ratio decreases. However, the tile overlap ratio of our proposed method is still higher than that of the two baseline methods. The higher the tile overlap ratio is, the more accurate the prediction result will be. In other words, our prediction method has the highest average accuracy compared with the baseline method. Use of the saliency heat maps and salient feature maps of other users can provide the best performance.

In addition, the experiment in this paper analyzes the influence of the number of feedback user viewports on the prediction method. Figures 8(a) and 8(b) show the effect of the number of feedback user viewports on the performance of the prediction method. Our method and Viewport Only include the user viewports to the prediction, and the tile overlap ratio increases as the number of feedback user viewports increases. Saliency Only does not consider the user viewport, and the change in the number of user perspectives does not affect the baseline method.

Figure 9 shows the relationship between our method and different time conditions and the number of feedback user viewports. The tile overlap ratio increases as the number of

**Input:** $\chi_{\text{relax}}$ Optimal objective function value for relaxation problem; $G_{\text{relax}}$ Optimal objective function value for relaxation problem; $\varepsilon$ Any value in the range (0, 1);
**Output:** $\chi_{0-1}$ The optimal solution to problem $p-1$ that meets the 0-1 constraint condition; $G_{0-1}$ The optimal objective function solution corresponding to the optimal 0-1 solution to problem $p-1$.

(1) Initialization: $k=0$, $L=0$, $U=G_{\text{relax}}$
(2) choose any solution $\chi_j$ that does not meet the 0-1 constraints from $\chi_{\text{relax}}$, that is, $\chi_j \in (0,1)$.
(3) **if** $0<\chi_j<\varepsilon$ **then**
(4) Add the constraint $\chi_j=0$ to problem $p-1$ to form subproblem I.
(5) **else**
(6) add the constraint $\chi_j=1$ to problem $p-1$ to form subproblem II.
(7) **end if**
(8) $k$++, continue to find the solutions to the relaxation problems in subproblems I and II (denoted as $\chi_k$), where the optimal objective function value is $G_k$.
(9) Find the maximum value of the optimal objective function as the new upper bound, that is, $U=\max\{G_k\}$, $\chi_k \in (0,1)$
(10) From the branches that meet the 0-1 condition, find the maximum value of the objective function as the new lower bound. $L=\max\{G_k\}$, $\chi_k \in (0,1)$
(11) **if** $G_k<L$ **then**
(12) cut this branch.
(13) **else if** $G_k>L$ and $\chi_k \in (0,1)$ **then**,
(14) return to step 2 and repeat.
(15) **else**
(16) the optimal solution to the then problem $p-1$ has been found, that is, $\chi_{0-1}=\chi_k$ and $G_{0-1}=G_k$
(17) **end if**

Algorithm 1: Branch-and-bound algorithm for solving the problem.

Table 1: Parameters of the four original videos used in the experiments.

| Sequence | Video name | Resolution | Frame rate | Duration (s) |
|---|---|---|---|---|
| 1 | Movie | 3840 × 1920 | 25 fps | 21 |
| 2 | Game | 3840 × 2160 | 24 fps | 30 |
| 3 | Basketball | 3840 × 1920 | 25 fps | 22 |
| 4 | Concert | 3840 × 1920 | 30 fps | 23 |

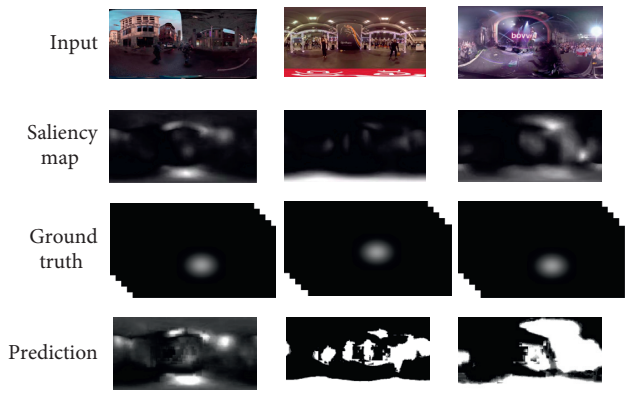


Figure 5: The saliency maps of only one frame are shown for each selected video. The prediction map is derived by combining the saliency map with the ground-truth user viewport.

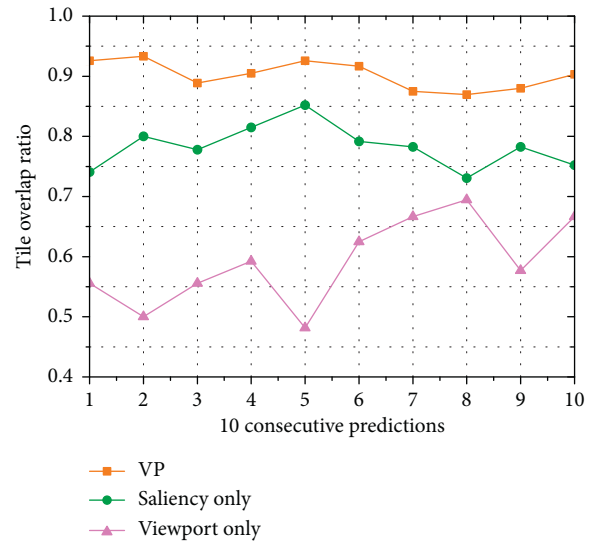


Figure 6: Example result of 10 consecutive predictions, with an interval time $t_0=0.2$ s.

feedback user viewports increases and decreases as the prediction time interval increases.

*6.2. Tile Rate Allocation.* To solve the optimization problem, we proposed a rate-adaptive algorithm based on the combined KKT conditions and branch-and-bound method (denoted as Algorithm 1). During the simulation, we first divided the 360-degree video into 10 GoPs, with the duration of each being 1 second, taking into account the FoV prediction result, and then divided each GoP into 4 × 4 tiles of the same size. We encoded 5 bitrate levels for

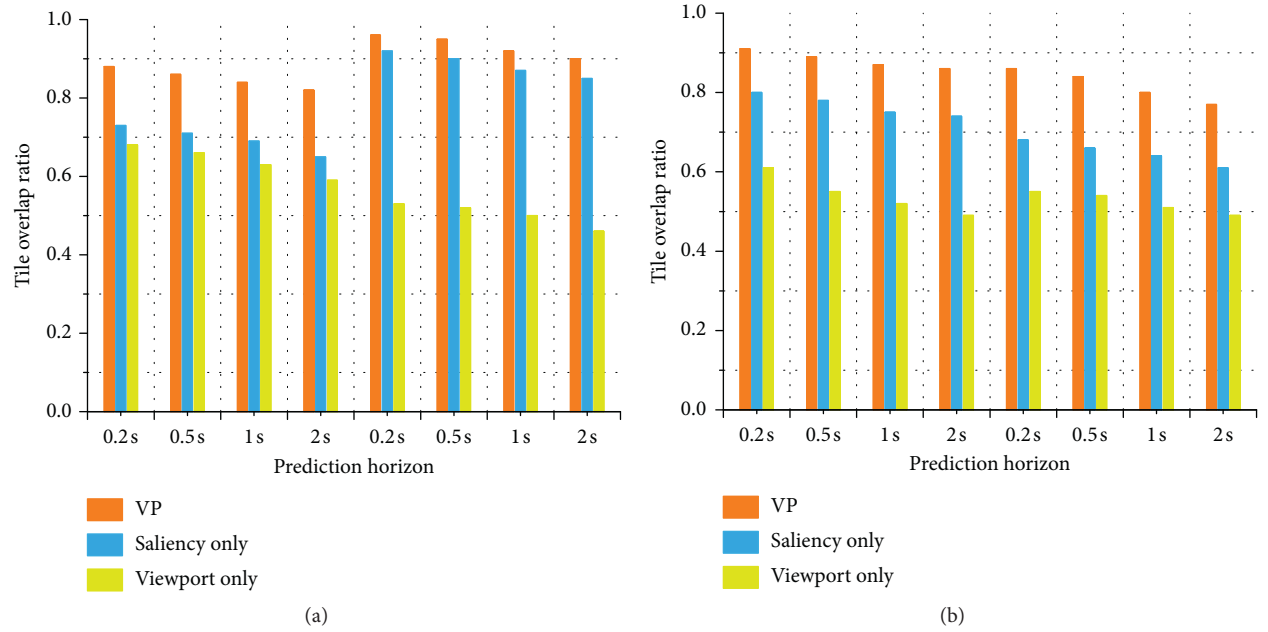


Figure 7: Prediction of the average tile overlap rate of the entire video at different time intervals. (a) Videos 1 and 2. (b) Videos 3 and 4.

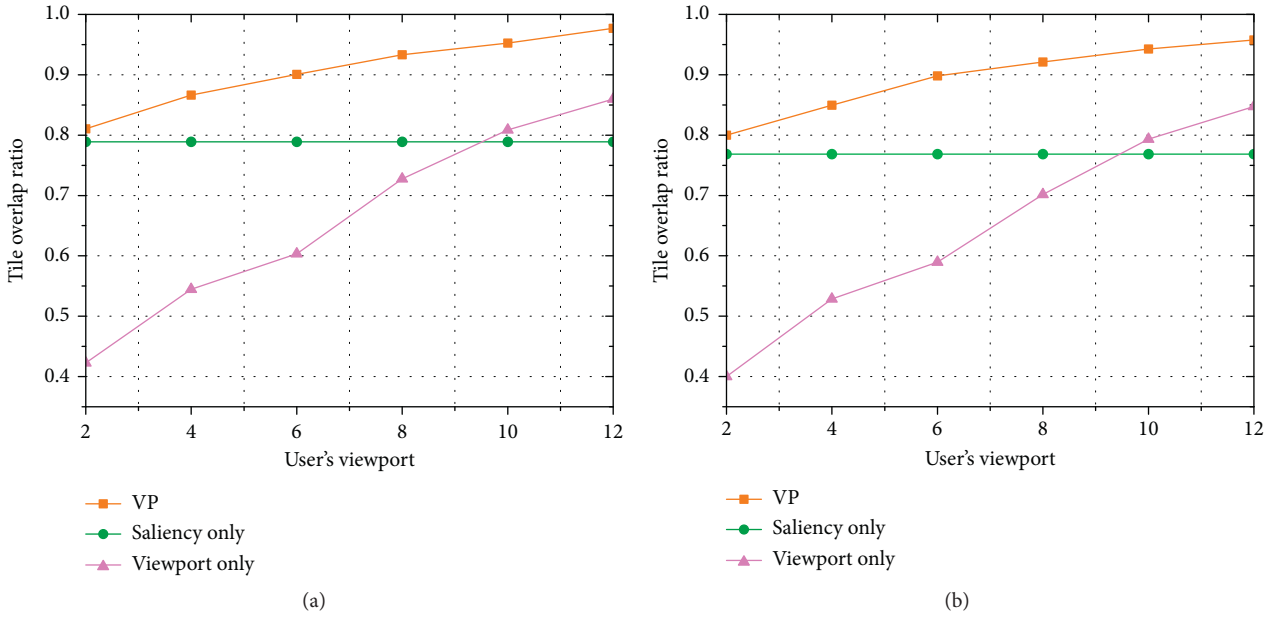


Figure 8: Tile overlapping ratio for different feedback user viewports. (a) Interval time $\tau_0 = 0.2$s. (b) Interval time $\tau_0 = 2$s.

each tile and set the bitrate levels to $\{0.2, 0.4, 0.6, 0.8, 1\}$ Mbps.

To verify the performance of this algorithm, we compared it with two other tile-based rate adaptation schemes: greedy [11] (denoted as Algorithm 2), where tiles in the predicted FoV region of users are allocated with high quality, while the remaining tiles are allocated with the lowest quality, and baseline (denoted as Algorithm 3), where all the tiles are transmitted to the users with the same quality and the FoV prediction is not taken into account. This is a simple algorithm and, thus, is used widely in practice [6].

The experiments in this paper evaluated the impact of the bandwidth in the downlink on the overall QoE of the system. In Figure 10, we can observe that the proposed algorithm can achieve higher QoE values under different bandwidth conditions and that the value of the QoE increases as the bandwidth increases. In addition, the QoE value of our proposed algorithm is significantly higher than that of the baseline method.

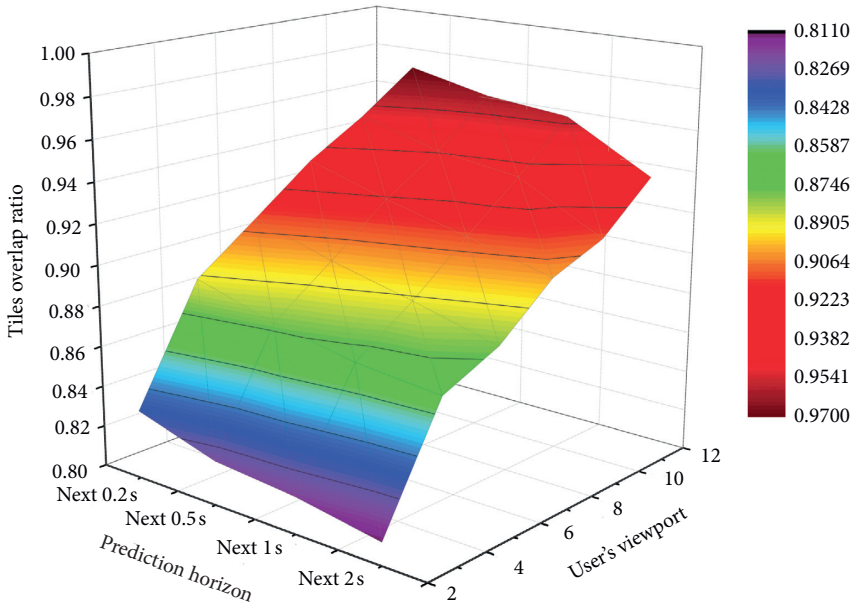


Figure 9: Tile overlapping ratio under different time conditions and different numbers of feedback user viewports.

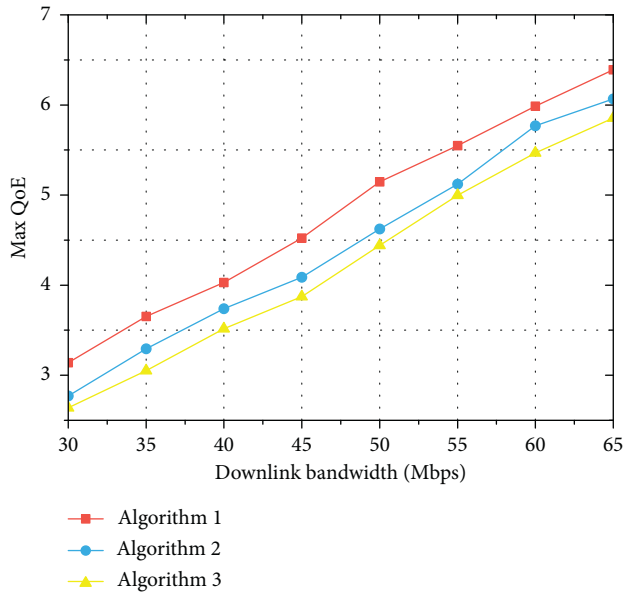


Figure 10: Relationship between downlink bandwidth and system QoE value.

To further verify the performance of the proposed scheme and make the simulation more realistic, we conducted simulations using real-world network traces [34] with perfect knowledge of future network conditions. The simulation results are shown in Figure 11(a). From the simulation results, we can observe that the proposed algorithm can achieve a higher QoE value with all four LTE traces because we have effectively allocated the resources. We can also observe that a larger bandwidth leads to a larger QoE value by comparing the performances achieved for different traces.

*6.2.1. Perceived Quality Level.* Figure 11(b) plots the average quality level of tiles for each streaming scheme. The average quality level of tiles produced by our scheme is the highest.

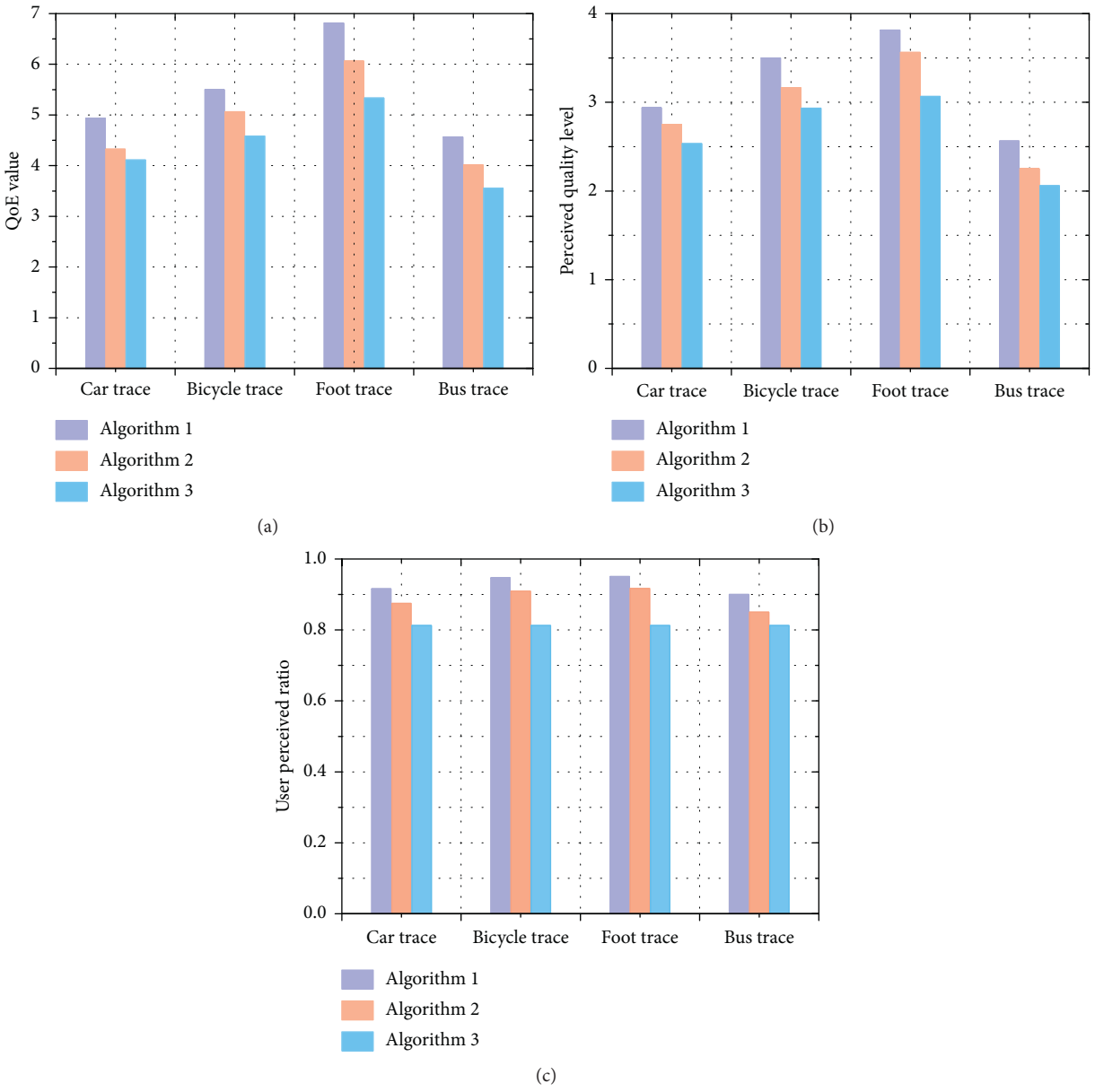


Figure 11: Comparison of 3 streaming schemes over the LTE network. (a) QoE value. (b) Perceived quality level. (c) User perception ration.

*6.2.2. User Perception Ratio.* This is the ratio between the actual bitrate consumed by the user and the predicted overall video bitrate. Figure 11(c) shows the ratio between the actual consumed and the overall video bitrate. The results show that the user perception ratios for our method significantly outperform those of Algorithm 2 and Algorithm 3.

Moreover, we also analyzed the impact of the number of feedback user viewports on the QoE value of all users. In our method and Algorithm 2, the FoV prediction is considered, and the accuracy of the FoV prediction is affected by the number of user viewports, thereby affecting the user QoE value. It can be found from Figure 12 that our method and Algorithm 2 increase as the number of feedback user viewports increases, as does the user QoE. However, Algorithm 3 does not consider the user FoV prediction; hence, the change in the number of viewports does not affect the user QoE.

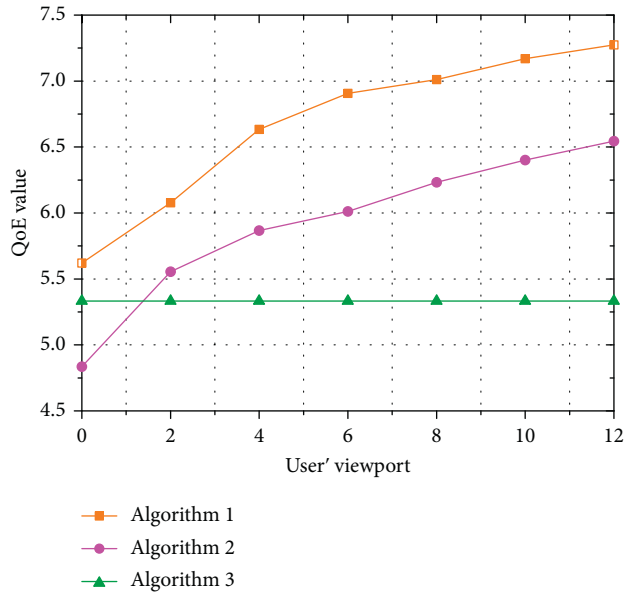


Figure 12: The relationship between the number of feedback user viewports and the QoE value of all users.

## 7. Conclusions

In this paper, we propose an FoV prediction method based on limited FoV feedback, which fuses and corrects the heat map obtained by saliency detection and FoV information randomly extracted from a small number of users to generate a user view to obtain a more accurate FoV prediction result for all users. We use this as a basis to optimize the transmission system. A multiuser QoE-driven 360-degree video streaming system is designed for adaptive 360-degree video streaming. According to the predicted FoV, under the constraint of the total transmission bitrate, by combining the KKT conditions and the branch-and-bound method, our algorithm can choose the optimal quality level for each video block, maximizing the QoE of all users. The simulation results show that the performance of our algorithm is superior to that of other schemes.

## Data Availability

The data used to support this research are available from the author at hanling@mail.hfut.edu.cn.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported in part by the Fundamental Research Funds for the Central Universities, under Grant nos. JZ2019HGTB0089 and PA2020GDJQ0027.

## References

[1] R. Yao, T. Heath, A. Davies, T. Forsyth, N. Mitchell, and P. Hoberman, "Oculus vr best practices guide," *Oculus VR*, vol. 4, 2014.

[2] B. Keshavarz, H. Hecht, and L. Zschutschke, "Intra-visual conflict in visually induced motion sickness," *Displays*, vol. 32, no. 4, pp. 181–188, 2011.

[3] S. Yang, Y. He, and X. Zheng, "Fovr: attention-based vr streaming through bandwidth-limited wireless networks," in *Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, London, UK, 2019.

[4] R. Monnier, R. van Brandenburg, and R. Koenen, "Streaming uhd-quality vr at realistic bitrates: mission impossible," in *Proceedings of the 2017 NAB Broadcast Engineering and Information Technology Conference (BEITC)*, Berlin, Germany, 2017.

[5] Z. Watson, "Vr for news: the new reality?" 2017.

[6] A. Ghosh, V. Aggarwal, and F. Qian, "A rate adaptation algorithm for tile-based 360-degree video streaming," 2017.

[7] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: divide and conquer," in *Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM)*, pp. 107–110, Berlin, Germany, 2016.

[8] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "Mpeg dash srd: spatial relationship description," in *Proceedings of the 7th International Conference on Multimedia Systems*, pp. 1–8, Berlin, Germany, 2016.

[9] LTE Introduction, 2013.

[10] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: motion-prediction-based transmission for 360-degree videos," in *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data)*, pp. 1161–1170, Berlin, Germany, 2016.

[11] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pp. 1–6, Berlin, Germany, 2016.

[12] N. Kan, J. Zou, K. Tang, C. Li, N. Liu, and H. Xiong, "Deep reinforcement learning-based rate adaptation for adaptive 360-degree video streaming," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4030–4034, London, UK, 2019.

[13] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in vr spherical video streaming," 2017.

[14] 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, New York, NY, USA, 2015.

[15] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. Kwong, "Spatial and temporal consistency-aware dynamic adaptive streaming for 360-degree videos," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 177–193, 2020.

[16] 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, London, UK, 2016.

[17] Y. Xu, Y. Dong, J. Wu et al., "Gaze prediction in dynamic 360° immersive videos," in *Proceedings of the 2018 IEEE/CVF*

*Conference on Computer Vision and Pattern Recognition*, pp. 5333–5342, London, UK, 2018.

[18] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: a deep reinforcement learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2693–2708, 2019.

[19] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very long term field of view prediction for 360-degree video streaming," in *Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 297–302, London, UK, 2019.

[20] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "Cub360: exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming," in *Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, London, UK, 2019.

[21] C. Guo, Y. Cui, and Z. Liu, "Optimal multicast of tiled 360 vr video," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 145–148, 2019.

[22] K. Long, Y. Cui, C. Ye, and Z. Liu, "Optimal wireless streaming of multi-quality 360 vr video by exploiting natural, relative smoothness-enabled and transcoding-enabled multicast opportunities," *IEEE Transactions on Multimedia*, vol. 8, 2020.

[23] 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.

[24] L. Xie, X. Zhang, and Z. Guo, "A cross-user learning based system for improving qoe in 360-degree video adaptive streaming," pp. 564–572, 2018.

[25] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," 2016.

[26] J. Li, R. Feng, W. Sun, Z. Liu, and Q. Li, "Qoe-driven coupled uplink and downlink rate adaptation for 360-degree video live streaming," *IEEE Communications Letters*, vol. 24, no. 4, pp. 863–867, 2020.

[27] S. Rossi and L. Toni, "Navigation-aware adaptive streaming strategies for omnidirectional video," 2017.

[28] D. V. Nguyen, H. T. T. Tran, A. T. Pham, and T. C. 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.

[29] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360probdash: improving qoe of 360 video streaming using tile-based http adaptive streaming," 2017.

[30] J. Zou, C. Li, C. Liu, Q. Yang, H. Xiong, and E. Steinbach, "Probabilistic tile visibility-based server-side rate adaptation for adaptive 360-degree video streaming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 161–176, 2020.

[31] R. I. T. D. C. Filho, M. C. Luizelli, M. T. Vega et al., "Predicting the performance of virtual reality video streaming in mobile networks," 2018.

[32] W. Lin and C.-C. Jay Kuo, "Perceptual visual quality metrics: a survey," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 297–312, 2011.

[33] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "Deepvs: a deep learning based video saliency prediction approach," 2018.

[34] J. van der Hooft, S. Petrangeli, T. Wauters et al., "HTTP/2-Based adaptive streaming of HEVC video over 4G/LTE networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.