MEC-Assisted FoV-Aware and QoE-Driven Adaptive 360° Video Streaming for Virtual Reality

Chih-Ho Hsu
Wireless and Mobile Networking Lab., Dept. of Electrical Engineering
National Taiwan University, Taipei, Taiwan
Email: b05611040@ntu.edu.tw

***Abstract*—Virtual reality (VR) has been envisioned as the killer-application in the 5G mobile networks. Among numerous VR services, 360° video streaming is the most promising one. Nevertheless, its wide adoption is hindered by large latency incurred in cloud-based video delivery and insufficient bandwidth resource in Radio Access Network (RAN). Fortunately, the emergence of Multi-access Edge Computing (MEC) become an enabler to fulfill the potential of VR by providing caching and computing resources at network edges. Also, since a user can view only a part of the entire 360° video frame due to the limitation of eye vision, user's Quality of Experience (QoE) can be further enhanced if we can predict his Field of View (FoV). In this paper, we propose a novel MEC-assisted FoV-aware and QoE-driven Adaptive Streaming (MFQAS) scheme for 360° videos. Specifically, we first provide a comprehensive QoE model for 360° video streaming. Second, we adopt AutoRegression Moving Average (ARMA) model in FoV prediction. Finally, we propose a heuristic algorithm to optimize the caching and computing decision at MEC server based on predicted FoV so that user's QoE can be enhanced. The simulation results show that our proposed method can provide much better prediction accuracy and QoE compared with baseline algorithms.**

***Index Terms*—adaptive 360° video streaming, viewport prediction, quality of experience, virtual reality, multi-access edge computing**

I. Introduction

Virtual reality (VR), aiming to provide interactive and immersive experiences to the users, has been envisioned as the killer-application in the 5G mobile networks. Also, the data traffic generated by VR devices is expected to increase 650% from 2017 to 2021 [1]. Among numerous VR services, 360° video streaming, where users can dynamically alter their Field of View (FoV) during video sessions, is the most promising one [2]. Nevertheless, its wide adoption is hindered by large latency incurred in current cloud-based network architecture, insufficient bandwidth resources in Radio Access Network (RAN) and limited battery and computation capacity of VR devices. Typically, 360° video streaming requires the perceived latency being less than 20ms to avoid dizzying [3] and consume more than 200 Mbps bandwidth to support playback with 8K resolution [4], which is 5 times higher than that of traditional videos services. Needless to say, 360° video requires transforming each 2D video frame, obtained by equirectangular projection (ERP), into 3D stereoscopic video in real-time to be displayed on the VR devices [4-7]. To address these challenging issues, a lot of efforts from both the academia and the industry have been made recently.

To begin with, proposed by ETSI [8], Multi-access Edge Computing (MEC) has emerged as a potential enabler to meet the ultra-low latency requirements of 360° video delivery. By providing caching and computing resources at network edges, the MEC system can shorten the distance between network services and users and thus the response time is significantly reduced. Specifically, by caching the required contents in advance [3-7],[9-13], users can enjoy the uninterrupted 360° video streaming services with less latency while alleviating data traffic of the backhaul network. Also, the computing resource of edge nodes can be further utilized [4-7], [9-11] to offload the computation-intensive preprocessing tasks from VR devices. In this way, not only the delay experienced by the users but also the energy consumption of VR devices can be reduced.

However, the existing edge-based solutions mainly focus on designing the decision policies of the MEC system, neglecting the intrinsic characteristics of 360° video streaming. Firstly, in state-of-art tiling-based streaming, a 360° video frame is spatially divided into multiple tiles, and each tile can be encoded with different resolution levels. Under such scheme, the MEC server can adaptively select appropriate video resolution for users given their wireless channel condition so that users' QoE can be further enhanced. Secondly, another unique attribute of 360° video streaming is that due to the limitation of the human eye vision, a user can only view 90 vertically and 110 horizontally of the 360° video at the same time. Thus, it will cause considerable bandwidth waste during transmitting the unviewed part of the 360° video.

Based on these features, there have been several works investigating FoV-aware adaptive 360° video streaming [13-24]. Among these schemes, FoV prediction serves as both the enabler and the most critical issue in optimizing users' QoE. Generally, FoV prediction approaches can be classified into two types, including content-agnostic and content-aware [2]. The content-aware solutions aim to predict the future FoV by analyzing historical data of the cross-users watching behavior. Though they can achieve high prediction accuracy, the historical viewing behavior of certain contents may be unavailable in many cases, making them be less implementable in the real world. On the other hand, the content-agnostic approaches take into account the historical value of the FoV in a video session to predict the future FoV of the user. Nonetheless, how to predict FoV in content-agnostic schemes with high accuracy and less complexity is still an open but critical issue.

Finally, another challenge is that users' perceptual QoE toward streaming services is more sensitive in the immersive environment compared with traditional 2D videos. Nevertheless, since adaptive streaming for 360° video is still a nascent research area, few existing studies have comprehensively modeled the QoE in their optimization problem.

Motivated by the above work as well as the challenges, in this paper, we aim to optimize users' QoE toward 360° video streaming by predicting FoV and utilizing caching and computing resources of edge nodes. Specifically, our main contributions are summarized as follows:

1) We investigate the potential benefits of providing adaptive 360° video streaming service for VR devices based on the edge computing paradigm. Specifically, by utilizing caching and computing resources at edge nodes to serve and process video requests locally, the user's perceived latency is reduced and thus QoE can be further enhanced.

2) We mathematically model the users' QoE toward 360° video streaming in terms of video resolution level, quality variation rate, and video stalling time. In this way, we can provide an effective guide to the QoE optimization of 360° video streaming service.

3) We are the first to adopt the ARMA model in FoV prediction. To comprehensively evaluate the proposed scheme, we adopt the real-world dataset to conduct the experiment, whose results show that the proposed method can achieve high accuracy with less complexity, resulting in better QoE and less bandwidth consumption compared with existing methods.

The remainder of this paper is organized as follows. Section II first discusses the related literature of our work. Secondly, our system model and problem formulation are organized in Section III. Afterward, Section IV presents the design of the proposed method. Eventually, Section V elaborates the simulation results to evaluate our proposed method and Section VI closes this work with conclusions.

II. Related Work

This section first briefly outlines existing efforts regarding the related topic of this paper, including edge-based 360° video streaming and FoV-aware 360° video streaming. Afterward, we will present a comparison table to validate our contribution.

A. 360° Video Streaming based on Edge Computing

To begin with, a view synthesis-based 360° VR caching system over C-RAN is designed in [3], where a specific view is cached in the Base Band Units (BBU) pool or Remote Radio Heads (RRH), or can be synthesized with the aid of the cached adjacent views. On the other hand, to optimize resource allocation at both Fog Access Points (F-APs) and VR devices, a joint radio communication, caching and computing decision framework framework is proposed in [4-5] with an aim to maximize the average tolerant delay. Similarly, the author of [6-7] presents a novel MEC-based VR delivery framework to jointly determine which FoVs to cache and which FoVs to compute at the mobile device under cache size, average power consumption as well as latency constraints. Furthermore, the author of [9] integrates scalable layered 360° video tiling, viewport-adaptive rate-distortion optimal resource allocation, and edge caching to propose a system that enables high-quality VR streaming. A mixed strategy is proposed in [10] to jointly consider the dynamic caching replacement and the deterministic offloading of the MEC system with an aim to minimize energy consumption and the latency. Then, the paper [11] leverages transcoding-enabled edge caching and makes joint caching, transcoding and delivery decisions among multiple edge servers for adaptive tile-based 360° video streaming. Finally, a VR video delivery system over named data network is discussed in [12]. They propose the cache policy integrating hotspot-based and popularity-based method.

B. Adaptive 360° Video Streaming and FoV Prediction

The FoV of users is assumed to be known when designing their adaptive streaming scheme in [14-16]. In [14], based on the assumption that FoV is known exactly beforehand, they fetch the invisible portion of a video at the lowest resolution and determine the video resolution for the visible portion based on bandwidth prediction. Similarly, by dividing each video frame into three zones with different priorities, the method proposed in [15] will adaptatively decide the video resolution in each prioritized zone. Also, by assigning different weights to regions inside and outside of the current viewport, a heuristic method is proposed in [16] to allocate video resolution of each video frame.

On the other hand, there are some literatures considering the content-aware FoV prediction scheme. Specifically, in [10], they first model the time-varying viewpoint popularity as a Markov chain and build a Long Short Term Memory (LSTM) network to predict the future FoV. Further, in [13,18] they adopt probability models to predict the visibility of a tile. In [18] they study the server-side rate adaptation strategy for multiple users based on the historical viewport data. Finally, in [12,17] the authors predict future FoV of an incoming request by calculating the average popularity of each viewpoint. Specifically, a novel viewport-based bitrate allocation algorithm is proposed in [17].

Finally, the adaptive 360° video streaming scheme based on content-agnostic FoV prediction is discussed in the following. The linear regression method is adopted in [19-20] for viewport estimations. Specifically, the approach proposed in [19] can efficiently decide the resolution of tiles according to user head movements and network conditions. In [20] they designed a 360° video streaming scheme to maximize QoE by predicting user's viewport and prefetch video segments into the buffer and replacing some unbefitting segments. On the other hand, the references [21-22] predict the future FoV by extending the current trajectory on the unit sphere. In [21] they design a heuristic rate adaptation for tile-based video, which prioritizes tiles according to the distance between the predicted viewport and the tile's center. Also, based on the bandwidth estimation, viewport prediction and user's buffer

TABLE I
A Comparison of Related Work in the Literature

#	Metric	Cache	Comput.	FoV	Prediction
3	Latency	O	X	Known	N/A
4	Latency	O	O	Known	N/A
5	Latency	O	O	Known	N/A
6	Latency	O	O	Uniform dis.	N/A
7	Latency	O	O	Uniform dis.	N/A
9	Quality	O	O	Known	N/A
10	Latency,EC	O	O	Adaptive	LSTM
11	Cost	O	X	Adaptive	Popularity
12	Latency	O	X	Adaptive	Popularity
13	Quality	O	X	Adaptive	MLE
14	Quality	X	X	Known	N/A
15	QoE	X	X	Known	N/A
16	Quality	X	X	Known	N/A
17	PSNR	X	X	Adaptive	Popularity
18	Quality	X	X	Normal dis.	Popularity
19	Quality	X	X	Adaptive	LR
20	Quality	X	X	Adaptive	LR
21	Quality	X	X	Adaptive	Trajectory
22	Quality	X	X	Adaptive	Trajectory
23	Bandwidth	X	X	Adaptive	NN
24	Bandwidth	X	X	Adaptive	NN
We	QoE	O	O	Adaptive	ARMA

Hints: **EC**=Energy Consumption, **PSNR**=Peak signal to Noise Ratio, **Comput**=Computing, **Dis**=Distribution, **MLE**=Maximum Likelihood Estimation, **ARMA**=Auto Regressive Moving Average

status, a heuristic video adaptation strategy is given in [22]. Finally, the neural network (NN) is trained to predict future FoV of users in [23-24]. A novel transmission mechanism is proposed in [23] to minimizes the overall bandwidth consumption of users based on the viewer's motion prediction. Then, in [24], they present a promising scheme to optimize the network bandwidth using motion prediction based multicast to serve concurrent viewers. However, their prediction model may require numerous training data and high computational complexity, which is less applicable in real-time scenario.

After classifying the literature by the resources utilized in MEC, how dynamics of users' FoV is considered and the way they predict future FoV, a comparison of related work is listed in TABLE I. We can first observe that most of the existing works merely focus on optimizing the network performance such as latency and quality while few studies had discussed the comprehensive QoE model for 360° video streaming service. In addition, few studies had jointly considered the integration of edge computing and FoV prediction for adaptive 360° video streaming. Therefore, we start to draw a design and implement these features.

III. System Model and Problem Description

In this paper, we investigate the MEC-assisted adaptive 360° video streaming system for VR with an aim to promote users' QoE. As shown in Fig. 1, our system composed of three main components: *1)* a small base station (SBS) equipped with a MEC server, *2)* Remote content server and *3)* one mobile VR device. In this section, we will focus on the mathematical description of the mentioned components. The main notations involved in our system are summarized in Table II.

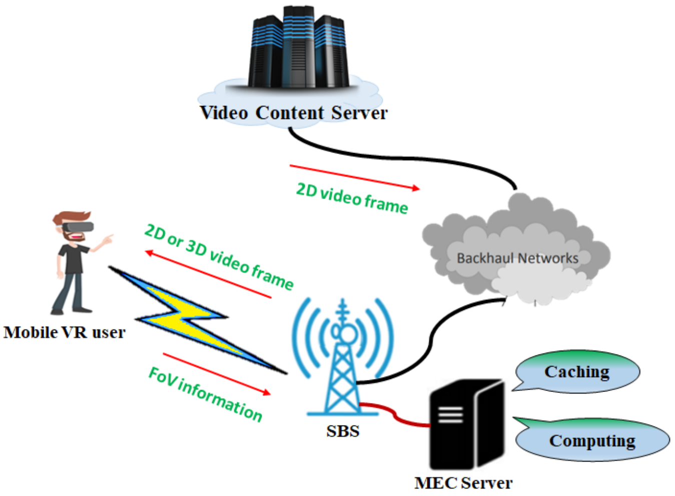


Fig. 1. Architecture of the MEC-assited 360° video delivery system.

A. *MEC-Assited* 360° *Video Delivery Framework*

To begin with, we consider that there is a libary of 360° video F stored in the video contents server and each video is divided into n consecutive chunks with fixed time duration Δt. Furthermore, each chunk of the video is partitioned into m spatial tiles and each tile can be independently encoded and streamed to the user. To present such idea, we define $\chi = \{\chi_1, \chi_2, ..., \chi_L\}$ as the set of supported resolution version, where χ_k is the encoding bitrate of k^{th} resolution level. Hence, we can express the size of a tile with resolution χ_k by $v(\chi_k)$. Finally, by expressing $r_{i,j} \in \chi$ as the allocated resolution at j^{th} tile in i^{th} chunk, the size of the requested chunk i will be $\sum_{j=0}^{m} v(r_{i,j})$ [14].

Based on the above elaboration, we present the delivery process for VR devices in our system as follows. As shown in Fig. 1. Initially, each frame in a video chunk (i.e a 2D equirectangular representation of the 360° spherical views) is stored in the video content server. The mobile VR user can dynamically request for the video based on his FoV (i.e. viewing area consisting a sub-sets of tiles among all tiles), and then the requested content will be transmitted through backhaul network and SBS. Finally, the obtained 2D video frame will be projected on the user's display in 3D format. Meanwhile, the MEC server is further utilized to facilitate the delivery process. Specifically, the MEC server will determine the resolution of each tile and then prefetch the video chunk into its cache based on the FoV prediction. In this way, the user can download the video directly from the MEC server with less latency. Also, the MEC server can help process the workload required for projection so as to relieve the processing delay of the VR device.

B. *Caching, Communication and Computing Model*

We first consider the cache placement at the MEC server. In our designed system, MEC server will update its cache list in first in first out (FIFO) manner. That is, every duration Δt, the oldest content in the cache, which had been delivered to

TABLE II
LIST OF NOTATIONS

Symbol	Definition
Δt	Fixed time duration of a video chunk
n	The number of chunks in a video
m	The number of tile in a video chunk
L	The number of supported video resolution
χ	The set of resolution level
$v(\chi_k)$	The size of a tile with resolution χ_k
$r_{i,j}$	The allocated resolution at j^{th} tile in i^{th} chunk
$c_{i,j,k}$	Cache decision at j^{th} tile with k^{th} resolution level
$x_{i,j}$	Computing decision at j^{th} tile of chunk i
C/X	Vector denoting the caching/computing decision
b_t	Downlink capacity of the VR device at instant t
T_i^D	Latency for downloading the video chunk i
T_i^C	The completion time for projecting video chunk i
$\omega_{i,j}$	Indicator denoting if the j^{th} tile of i^{th} chunk is in FoV
D_{MEC}	Caching capacity of the MEC server
f_{VR}	Computing capacity of the VR device
f_{MEC}	Computing capacity of the MEC server
q	The order of ARMA model
Y^t/P^t	The recorded yaw and pitch angle at time t
$\hat{Y}/\hat{P}^t$	The predicted yaw and pitch angle
ρ_j	The priority of j^{th} tile
h_j	Distance between predicted FoV and j^{th} tile

the user, will be removed while the MEC server will continue to prefetch the video chunk that may be requested in the near future. As a result, equipped with caching capacity D_{MEC}, the aim of the MEC server is to determine which resolution version of the tiles in chunk should be cached based on the FoV prediction. Mathematically, we denote the $c_{i,j,k} \in \{0,1\}$ as the caching decision of MEC server. That is, $c_{i,j,k} = 1$ if the j^{th} tile with k^{th} resolution version of the i^{th} chunk is decided to be cached and $c_{i,j,k} = 0$ otherwise. This can be regarded as a complement to the notation $r_{i,j}$ becasue the cached tile will be deliver to user in the future. For example, if $c_{i,j,k} = 1$, the user will perceive the tile j with k^{th} resolution level (i.e. $r_{i,j} = \chi_k$) in the future.

On the other hand, by adopting large-scale pathloss and shadowing in modeling the wireless connection between the user and SBS, the achievable downlink data rate of the VR device at instant t can be obtained by Shannon capacity, formulated as:

$$b_t = BW \log_2 \left(1 + \frac{gPW}{\sigma^2}\right) \tag{1}$$

where BW is the dedicated frequency band, PW signifies the transmission power of the SBS and g is the channel gain between the user and SBS. Additionally, σ^2 is the noise variance of the additive white Gaussian noise (AWGN). Afterward, we can calculate the expected latency T_i^D for downloading the video chunk i from the MEC server as follow.

$$T_i^D(C_i) = \frac{\sum_{j=0}^{m} v(r_{i,j})}{b_t} \tag{2}$$

where $C_i = [c_{i,1,1}, c_{i,1,2}, ..., c_{i,m,k}]$ is a vector denoting caching decision of chunk i.

We turn to consider the computing decision for the projection task of each tile. Specifically, the projection task of tile j in i^{th} chunk is described by a tuple $(b_{i,j}^I, b_{i,j}^O, a_{i,j})$, where $b_{i,j}^I$ and $b_{i,j}^O$ are the data sizes of the 2D FV and 3D FoV, respectively, and $a_{i,j}$ is the number of required CPU cycles of the projection task of tile j. In general, $\frac{b_{i,j}^O}{b_{i,j}^I}$ must greater than 2 in order to create a stereoscopic vision. Then, we denote $x_{i,j} \in \{0,1\}$ the computing decision of the tile j in chunk i, where $x_{i,j} = 1$ indicates that the projection is processed at MEC server and $x_{i,j} = 0$ if the projection is executed at mobile VR device. Finally, by denoting the computing capacity of mobile VR device and MEC server as f_{VR} and f_{MEC} respectively, the completion time T_i^C for projecting all of the tiles in i^{th} chunk can be written as.

$$T_i^C(X_i) = max\left(\frac{\sum_{j=0}^{m} x_{i,j} a_{i,j}}{f_{VR}}, \frac{\sum_{j=0}^{m} (1 - x_{i,j}) a_{i,j}}{f_{MEC}}\right) \tag{3}$$

Similarly, $X_i = [x_{i,1}, x_{i,2}, ..., x_{i,m}]$ is a vector denoting the computing decision of chunk i.

C. QoE Model

As pointed out by several research works in Dynamic Adaptive Streaming over HTTP (DASH) [2], the main factors that affect the QoE in video streaming are 1) video resolution, 2) quality variation rate, and 3) video stalling time. We will mathematically describe these factors in the following.

First, since the effect of video resolution toward QoE would follow logarithmic law, as noted in [25], we consider video resolution in our QoE model as follow:

$$VQ = \frac{1}{n} \sum_{i=0}^{n} \sum_{j=0}^{m} \omega_{i,j} \cdot \log(r_{i,j}) \tag{4}$$

where $\omega_{i,j} \in \{0,1\}$ signify whether the tile fall in the current FoV. That is, $\omega_{i,j} = 1$ if the j^{th} tile in i^{th} video chunk is covered by the user's FoV and $\omega_{i,j} = 0$ otherwise.

Another metric toward the user's QoE is the quality variation rate, which can be obtained by the difference of current resolution and resolution in the previous moment. Here, we take the absolute of the resulting value to keep it positive. Then, the average quality variation rate experienced by the user is defined as:

$$SW = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=0}^{m} \omega_{i,j} \omega_{i-1,j} \cdot |r_{i,j} - r_{i-1,j}| \tag{5}$$

Note that in the product $\omega_{i,j}\omega_{i-1,j}$ in the equation state that the user will perceive quality variation of a tile only if the tile fall in the FoV when watching current and previous video chunk. The last QoE metric is video stalling, which happens when the playback buffer gets empty. Without loss of generality, we formulate the stalling time of video perceived by user as:

$$ST = \frac{1}{n} \sum_{i=0}^{n} max\Big(T_i^D(C_i) + T_i^C(X_i) - \Delta t, 0\Big) \tag{6}$$

In equation (6). The term $T_i^D(C_i) + T_i^C(X_i) - \Delta t$ means that the download time and computing time of the required resolution should be less than the duration of a chunk so that the next video chunk can be played on time. Otherwise, the stalling event will happen and the value of ST will increase.

Finally, we define the QoE function of mobile VR user as a weighted sum of three factors mentioned above, expressed as follow:

$$QoE = \alpha VQ - \beta SW - \gamma ST, \left\{ \begin{array}{l} \alpha + \beta + \gamma = 1 \\ 0 \le \alpha, \beta, \gamma \le 1 \end{array} \right. \quad (7)$$

where the α, β, γ is the weight factor representing the importance of these three different metrics respectively.

Eventually, the problem of this work is defined as to maximize the user's QoE of the entire streaming session by determining caching and computing decision, which can be formulated as follow:

$$\begin{aligned} & \max_{C,X} QoE \\ \text{s.t.} \quad & C1 : \sum_{j=0}^{m} \sum_{k=0}^{L} c_{i,j,k} \cdot v(\chi_k) \le D_{MEC}, \forall i \\ & C2 : r_{i,j} \in \chi, \forall i, j \end{aligned} \quad (8)$$

In (8), constraint C1 describes that the total size of all video chunks to cache should not exceed the cache capacity of the MEC server. Then, constraint C6 represents the encoding bitrate of the tile must fall in the set of the supported resolution level.

IV. MEC-assisted FoV-aware QoE-driven Adaptive 360° Video Streaming Scheme

In this section, we will first illustrate the designed FoV prediction model in our system. Then, we will propose a heuristic method called MEC-assisted FoV-aware QoE-driven Adaptive Streaming (MFQAS) to optimize users' QoE.

A. FoV Prediction

Generally, once the value of yaw and pitch angle in VR devices is given, the FoV of a user is determined. Thus, predicting FoV can be regarded as predicting the yaw angle and pitch angle of a user's gaze. In this work, we treat the FoV prediction as a time-series regression problem and predict yaw and pitch independently [23]. To achieve high prediction accuracy with low complexity, we adopt the ARMA model, a statistics-based and widely used prediction method, in our FoV prediction. In the ARMA model, the future value of a variable is a linear combination of past values and past errors [26]. Thus, given the historical value of pitch and yaw, the predicted value by ARMA with order q can be formulated as:

$$\begin{aligned} \hat{Y} &= \sum_{n=1}^{q} \varphi_n Y^{t-n} + \sum_{n=1}^{q} \psi_n \zeta^{t-n} \\ \hat{P} &= \sum_{n=1}^{q} \varphi_n P^{t-n} + \sum_{n=1}^{q} \psi_n \zeta^{t-n} \end{aligned} \quad (9)$$

where t present the present time, Y^{t-n} and P^{t-n} are the recorded yaw and pitch at time $(t-n)$ respectively and $\hat{Y}^t$, $\hat{P}^t$ are the predicted angle. Also, in this equation, ζ^{t-n} is the prediction error of $\hat{P}$ or $\hat{Y}$ at time $(t-n)$. Finally, φ and ψ are the weight coefficients of the ARMA model. In this way, we can predict the future FoV based on the user's motion data collected in a sliding window of size q.

B. MEC-assisted FoV-aware QoE-driven Adaptive Streaming

1) Resolution selection: In the proposed MFQAS, an appropriate resolution will be selected for each tile based on the predicted FoV. To begin with, since only the tiles within the actual FoV will be watched by the mobile VR user, we should provide these tiles with higher resolution to improve user's perceived QoE. On the other hand, since the tile outside the region of predicted FoV is less like to be watched by the user, we will gradually reduce the resolution of these tiles based on the distance between the center of predicted FoV and center of each tile. In this way, as long as resolution decisions is optimized, the bandwidth consumption can be significantly reduced while achieving decent QoE.

Based on the presented concept, we define the priority for each tile in each video chunk to quantify the impact of a tile on the QoE [12], which can be expressed as follow.

$$\rho_j = \left\{ \begin{array}{ll} 1, & \text{If tile } j \text{ inside the predicted FoV} \\ \frac{1}{\mu h_j}, & \text{Otherwise.} \end{array} \right. \quad (10)$$

where h_j is the distance between the center of predicted FoV and center of tile j and μ is an adjustable coefficient. The value of ρ_j will be 1 if tile j fall in the region of predicted FoV, implying that we treat equally to the tiles in predicted FoV. Then, we will adopt the concept of priority in designing our heuristic method. An example of the resulting resolution distribution of the proposed MFQAS is shown in Fig. 2.

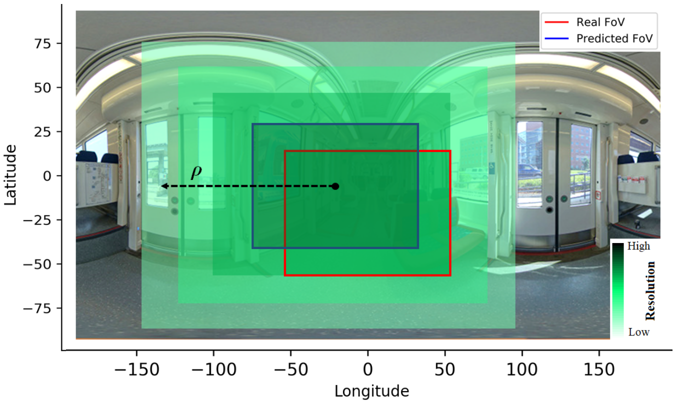


Fig. 2. Example of predicted FoV and resolution decision in the our system.

2) Optimization metric for the heuristic: To begin with, since the defined QoE model (7) can only be measured at the end of playing, we need a more reasonable optimization metric to measure the effect of caching, computing decisions in real-time. Thus, we define a new objective ϖ in the below.

$$\begin{aligned} \varpi = & \max_{C,X} \sum_{j=0}^{m} \rho_j \left(\alpha \log \left(r_{i,j} \right) - \beta \omega_{i-1,j} \left| r_{i,j} - r_{i-1,j} \right| \right) \\ & \text{s.t.} \quad \mathrm{C1, C2, C3: T_i^D(C_i) + T_i^C(X_i) \leq \Delta t} \end{aligned} \tag{11}$$

which can be interpreted as to maximize the expected instant video resolution and instant quality variation in the near future by determining caching and computing decisions. Similar to (5), the term $\omega_{i-1,j}$ means that the user will perceive quality variation of tile j only if the tile j fall in the FoV in the previous chunk. The constraint C1-2 and α, β are set as same in (8) while the additional constraint C3 is added to prevent the potential stall events causing by inappropriate caching and computing decisions. In this way, the algorithm will not select the resolution that can maximize the value of ϖ but negatively affect the overall QoE. Also, note that since the designed objective (11) only serves as the optimization metric for the algorithm, it didn't equivalent to the original problem (8). Instead, we intend to optimize QoE of the entire video session by adaptively optimizing (11) in every duration of a video chunk (i.e. Δt).

3) Workflow of MFQAS: In the designed solving process, the algorithm will first predict the future FoV by the ARMA model described in Sec. IV A. Next, we will initialize all of the computation tasks to be executed at the MEC server for the purpose of saving energy consumption of VR device, and set all of the tiles to the lowest resolution (i.e. χ_1). After that, we will sort each tile in descending order based on the obtained benefit if we upgrade its resolution, where the benefit for upgrading a tile is calculated by the difference of ϖ (i.e. $\varpi' - \varpi$). In the beginning, the algorithm tends to increase the resolution of the tiles within the predicted FoV because they have the highest priority (i.e. 1) and thus contribute significantly to the value of ϖ. However, since the resolution toward QoE is a logarithmic function, at some point, the benefit for increasing the resolution of tiles within the predicted FoV may be less than the benefit for increasing the resolution of the other tiles. In that case, the algorithm will turn to increase the resolution of the other tiles. We repeat this process until every tile is in its highest resolution or the constraint C3 is about to be violated. For the latter case, the algorithm will first sort the computation task by their required CPU cycle (i.e. a_j). Afterward, the algorithm will change the computing decision from executing at MEC server to VR device one by one in ascending order until the constraint C3 is satisfied. Ideally, the final resolution decision will be similar to that in Fig. 2. Finally, we will assign the caching decision of the MEC server based on the optimized resolution (e.g. set $c_{i,j,3}$ to 1 if $r_{i,j} = \chi_3$) and the MEC server will prefetch the corresponding chunk into its local cache. Next, the MEC server will execute the assigned projection task in real-time. The overall procedure is illustrated in Alg. 1.

Algorithm 1: The Proposed MFQAS Scheme

```
Input: The historical value of Yaw and Pitch ($Y\&P$)
Output: Caching and computing decision of chunk $i$ ($C\&X$)
Predict FoV of user ($\hat{Y}\&\hat{P}$) by (9)
Set $x_{i,j} = 1, \forall j$ & Set $c_{i,j,1} = 1, \forall j$ // initilization
while $T_i^D(C_i) + T_i^C(X_i) \leq \Delta t$ do
    Sort all tiles descendingly by their benefit ($\varpi' - \varpi$)
    $j \leftarrow$ the tile with maximum benefit increment.
    Set $r_{i,j} = \chi_{i,j,k+1}$, $k$ is the current resolution level
    if $r_{i,j} = \chi_L, \forall j$ then
        break
    end
    foreach $0 < j \leq m$ do
        Set $c_{i,j,k} = 1$ if $r_{i,j} = \chi_k$ //update caching decision
    end
    if $T_i^D(C_i) + T_i^C(X_i) > \Delta t$ then
        Sort each computation task $j$ by the value of $a_{i,j}$
        foreach $0 < j \leq m$ do
            Set $x_j = 0$ //update computing decision
            if $T_i^D(C_i) + T_i^C(X_i) \leq \Delta t$ then
                break
            end
        end
    end
end
```

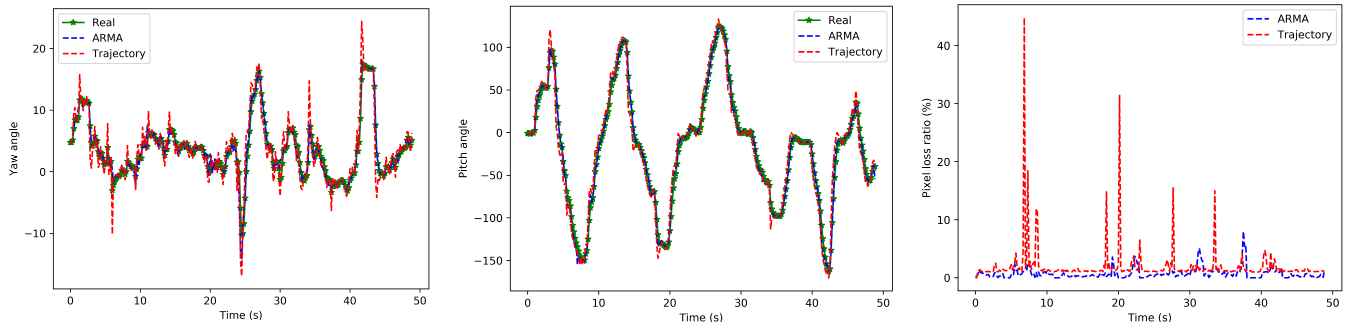
V. Simulation Result

In this section, we will conduct a series of experiments to evaluate the proposed MFQAS. Specifically, a real-world dataset [27] is adopted to simulate users' dynamic FoV. The dataset contains 28 360° videos and the recorded FoV traces from 60 users watching these video. The detailed simulation setting is given in the following.

To begin with, we consider each chunk consist of 32 tiles [14], which maps the overall 360° degree video in 2D video frame. Then, each tile is encoded into 8, 16, 24, 32 Mbps and the duration of each chunk is 1 second. Also, the data sizes of 2D FoV (i.e. $b_{i,j}^{I}$) is set to be $[15, 25]$Mbits and the size of 3D FoV (i.e. $b_{i,j}^{O}$) is set to be 2.5 times of b_j^I. On the other hand, the cache size and computing capacity of the MEC server are set to be 50GB and 100Mbps respectively. As for wireless video delivery, the bandwidth of 10MHz (i.e. BW) is considered. Besides, we set the the parameters for QoE model (7) with (α, β, γ) = (0.5, 0.1, 0.4) and the order of ARMA model *q* to be 5. Eventually, we compare the proposed MFQAS with several baselines, including *1)* the method proposed in [22], which predict FoV by extending the current trajectory and allocate video resolution by dividing the video frame into higher priority, medium priority and lower priority area, and *2)* Full transmission scheme, which will delivery entire 360° videos with the highest resolution to the mobile VR device.

A. Performance of FoV Prediction

Since FoV prediction plays a critical role in MFQAS, it is important to analyze the effectiveness of our prediction model. To begin with, Fig. 3 (a-b) shows the real-time dynamics of user's actual FoV and predicted FoV in the video trace



(a) Yaw (b) Pitch (c) Pixel Loss Ratio

Fig. 3. Performance of FoV prediction.

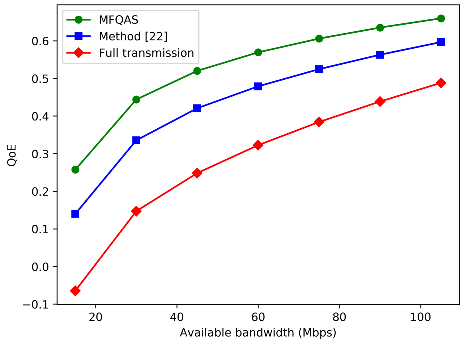


Fig. 4. The QoE Performance verse the avaliable bandwidth.

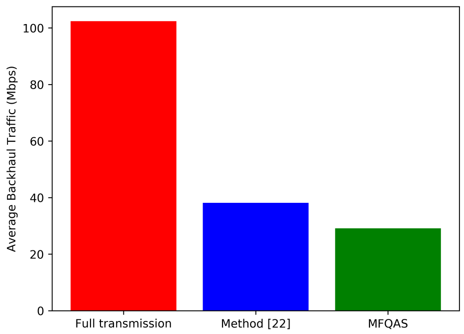


Fig. 5. The average bandwidth in different video delivery schemes.

$'0Z4VWJ1'$. We can observe that, in general, the performance of the ARMA model used in this work is better than the trajectory based prediction adopted in [22], especially at the turning point of the curve. This is because prediction in trajectory based method is made by assuming the head movement of users to be linear, which may be less accurate when the sudden motion of the user occurs. In addition, we can see from Fig. 3 (a-b) that the pitch angle in FoV is rather steady while the yaw angle tends to fluctuate with time. As a result, both the trajectory based prediction and ARMA model achieve higher accuracy in predicting pitch angle than predicting yaw angle.

We then adopt another metric named pixel loss rate [24] to systematically present the performance of FoV prediction in Fig. 3(c), which measures the percentage of actual FoV that did not fall in the region of predicted FoV. We can observe that, as expected, the performance of ARMA on average is better than the trajectory based prediction. Also, the pixel loss rate of trajectory based prediction will be much higher in the turning point of the FoV curve, which can be caused by the sudden motion of the user.

B. Analysis of QoE Performance

Fig. 5 illustrates the QoE performance under different available bandwidth of the backhaul network. Generally, the QoE performance in all of schemes will increase with available bandwidth logarithmically. It is reasonable because the video resolution in the QoE model (7) is considered in logarithmic form, and the increased bandwidth will directly contribute to the received video resolution. Also, we can observe that the full transmission scheme receives the lowest QoE among the comparison. This mainly because the stalling time incurred by insufficient bandwidth in full transmission will be much greater than the other schemes, who will adaptively determine video resolution based on the network condition. On the other hand, the result shows that MFQAS achieves better QoE than the method [22]. This could result from 3 perspectives: *1)* the FoV prediction made in the ARMA model is more accurate than the method [22], as shown in Fig. 3, and thus we can provide more precise resolution decision *2)* the method [22] only consider resolution selection in 3 different regions with corresponding priority while the proposed MFQAS will gradually reduce the resolution decision depended on the distance between predicted FoV and each tile so that QoE the can be further enhanced, and *3)* with the aid of MEC server, the processing delay and transmission delay can be further reduced in MFQAS so that the probability of video stalling will also be decreased accordingly, resulting in better QoE.

C. Analysis of Network Performance

Finally, Fig. 5 presents the average backhaul traffic of different video delivery scheme. First, we can observe that full transmission has the highest backhaul traffic among all of the compared schemes. In contrast, the backhaul traffic in MFQAS and the method [22] is significantly reduced since instead of transmitting the entire video frame with the highest resolution, we selectively adapt the resolution of each tile based on user's FoV. Then, it can be shown that MFQAS has fewer bandwidth consumption compared with the method [22]. This is because in [22], though they allocate different resolutions to the corresponding region, all of the tiles will be downloaded and then be delivered to the VR device. In contrast, the resolution decision in MFQAS scheme will be made depended on the result of optimization. Thus, in certain cases, some tiles may even not be downloaded. In this way, we can further reduce bandwidth consumption while maintaining high QoE performance.

VI. Conclusions

In this paper, we proposed a novel MEC-assisted FoV-aware and QoE-driven adaptive 360° video streaming scheme called MFQAS. Specifically, we investigate the potential benefit of MEC toward 360° videos streaming for VR by utilizing caching and computing resources at the network edge. In this way, we can reduce both transmission delay and processing delay at VR device and enhance user' QoE by offloading projection task from VR device and prefetching video chunks with corresponding resolution in each tile based on the FoV prediction. The simulation results show that the adopted ARMA model can achieve high prediction accuracy with less complexity. Based on predicted FoV, the proposed MFQAS can provide much better QoE compared with baseline algorithms while saving bandwidth consumption of the backhaul network.

References

[1] V. N. I. Cisco, "Forecast'cisco visual networking index: Global mobile data traffic forecast update, 2016–2021'," 2017.

[2] A. Yaqoob, T. Bi and G. Muntean, "A Survey on Adaptive 360 Video Streaming: Solutions, Challenges and Opportunities," in IEEE Communications Surveys & Tutorials, to appear.

[3] J. Dai, Z. Zhang, S. Mao and D. Liu, "A View Synthesis-based 360°VR Caching System over MEC-enabled C-RAN," in IEEE Transactions on Circuits and Systems for Video Technology, to appear.

[4] T. Dang and M. Peng, "Joint Radio Communication, Caching, and Computing Design for Mobile Virtual Reality Delivery in Fog Radio Access Networks," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 7, pp. 1594-1607, July 2019.

[5] T. Dang, M. Peng, Y. Liu and C. Liu, "Joint Bandwidth, Caching, and Computing Resource Allocation for Mobile VR Delivery in F-RANs," in IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, pp. 1-6, 2019.

[6] Y. Sun, Z. Chen, M. Tao and H. Liu, "Communication, Computing and Caching for Mobile VR Delivery: Modeling and Trade-Off," in IEEE International Conference on Communications (ICC), Kansas City, MO, pp. 1-6, 2018.

[7] Y. Sun, Z. Chen, M. Tao and H. Liu, "Communications, Caching, and Computing for Mobile Virtual Reality: Modeling and Tradeoff," in IEEE Transactions on Communications, vol. 67, no. 11, pp. 7573-7586, Nov. 2019.

[8] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal et al., "Mobile-edge computing introductory technical white paper," White Paper, Mobile-edge Computing (MEC) industry initiative, 2014.

[9] J. Chakareski, "Viewport-Adaptive Scalable Multi-User Virtual Reality Mobile-Edge Streaming," in IEEE Transactions on Image Processing, vol. 29, pp. 6330-6342, 2020.

[10] C. Zheng, S. Liu, Y. Huang and L. Yang, "MEC-Enabled Wireless VR Video Service: A Learning-Based Mixed Strategy for Energy-Latency Tradeoff," in IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea (South), pp. 1-6, 2020.

[11] Q. Lu, C. Li, J. Zou, K. Tang, Q. Wang and H. Xiong, "Transcoding-Enabled Edge Caching and Delivery for Tile-Based Adaptive 360-Degree Video Streaming," in IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, pp. 1-4, 2019.

[12] Y. Zhang, X. Jiang, Y. Wang and K. Lei, "Cache and delivery of VR video over named data networking," in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, pp. 280-285, 2018.

[13] A. Mahzari, A. T. Nasrabadi, A. Samiei and R. Prakash. "FoV-Aware Edge Caching for Adaptive 360° Video Streaming." in Proceedings of the 26th ACM international conference on Multimedia, 2018.

[14] A. Ghosh, A. Vaneet and Q. Feng, "A Rate Adaptation Algorithm for Tile-based 360-degree Video Streaming," 2017. [Online].Available: https://arxiv.org/abs/1704.08215.

[15] Y. Han, Y. Ma, Y. Liao and G. Muntean, "QoE Oriented Adaptive Streaming Method for 360° Virtual Reality Videos," in IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, Leicester, United Kingdom, pp. 1655-1659, 2019.

[16] C. Ozcinar, A. De Abreu and A. Smolic, "Viewport-aware Adaptive 360° Video Streaming using tiles for Virtual Reality," in IEEE International Conference on Image Processing (ICIP), Beijing, pp. 2174-2178, 2017.

[17] C. Ozcinar, J. Cabrera and A. Smolic, "Omnidirectional Video Streaming Using Visual Attention-Driven Dynamic Tiling for VR," in IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, pp. 1-4, 2018.

[18] C. Liu, N. Kan, J. Zou, Q. Yang and H. Xiong, "Server-Side Rate Adaptation for Multi-User 360-Degree Video Streaming," in 25th IEEE International Conference on Image Processing (ICIP), Athens, pp. 3264-3268, 2018.

[19] 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," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, pp. 29-42, March 2019.

[20] H. Hu, Z. Xu, X. Zhang and Z. Guo, "Optimal Viewport-Adaptive 360-Degree Video Streaming Against Random Head Movement," inIEEE International Conference on Communications (ICC), Shanghai, China, pp. 1-6, 2019.

[21] J. v. der Hooft, M. Torres Vega, S. Petrangeli, T. Wauters and F. D. Turck, "Optimizing Adaptive Tile-Based Virtual Reality Video Streaming," in IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, pp. 381-387, 2019.

[22] F. Yang, J. Luo, J. Yang and Z. Xu, "Region Priority Based Adaptive 360-Degree Video Streaming Using DASH," in International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, pp. 398-405, 2018.

[23] 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 IEEE International Conference on Big Data (Big Data), Washington, DC, , pp. 1161-1170, 2016.

[24] Y. Bao, T. Zhang, A. Pande, H. Wu and X. Liu, "Motion-Prediction-Based Multicast for 360-Degree Video Transmissions," in 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), San Diego, CA, pp. 1-9, 2017.

[25] W. Zhang, Y. Wen, Z. Chen and A. Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks," in IEEE Transactions on Multimedia, vol. 15, no. 6, pp. 1431-1445, Oct. 2013.

[26] G. E. P. Box, G. M. Jenkins and G. C. Reinsel, "Time series analysis: Forecasting and control," Prentice-Hall, Englewood Cliffs, NJ, 1994.

[27] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. McMahan, R. Prakash, M. C. Farias, and M. M. Carvalho, "A taxonomy and dataset for 360 videos," in Proceedings of the 10th ACM Multimedia Systems Conference, pp. 273–278, 2019.