Ebublio: Edge Assisted Multi-User 360-Degree Video Streaming

Yili Jin, Junhua Liu, Fangxin Wang, *Member, IEEE*, and Shuguang Cui, *Fellow, IEEE*

***Abstract*—As one of the most important manifestations of virtual reality (VR), 360° panoramic videos in recent years have experienced booming development due to the desire for immersive and interactive experiences. Compared to traditional videos, 360° videos are featured with uncertain user field of view (FoV), more sensitive delay tolerance, and much higher bandwidth requirement, bringing unprecedented challenges to 360° video streaming. Meanwhile, the development of 5G and mobile edge computing starts to pave the way for high-bandwidth low-latency video streaming. Some preliminary works focus on either individual FoV prediction or multi-user QoE oriented cache strategy design, while how to design a holistic solution toward optimizing the overall user QoE with considerations over fairness and long-term system cost remains a non-trivial problem.**

In this paper, we propose `Ebublio`, a novel intelligent edge caching framework to address the aforementioned challenges in 360° video streaming. `Ebublio` consists of a collaborative FoV prediction (CFP) module and a long-term tile caching optimization (LTO) module to jointly optimize the long-term user QoE and system cost. The former module integrates the features of video content, user trajectory, and other users' records for combined prediction. The latter one employs the Lyapunov framework and a subgradient optimization approach towards the optimal caching replacement policy. Our trace-driven evaluation demonstrates the superiority of our framework, with about 42% improvement in FoV prediction, and 36% improvement in QoE at similar traffic consumption.

***Index Terms*—360° video streaming, Edge computing and caching, Lyapunov optimization.**

(Corresponding author: Fangxin Wang.)

Yili Jin and Junhua Liu are with the Future Network of Intelligence Institute (FNii) and School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen. (e-mail: {yilijin,junhualiu}@link.cuhk.edu.cn)

Fangxin Wang is with the School of Science and Engineering and the Future Network of Intelligence Institute (FNii), the Chinese University of Hong Kong, Shenzhen; with the Guangdong Provincial Key Laboratory of Future Networks of Intelligence; and also with Peng Cheng Laboratory. (e-mail: wangfangxin@cuhk.edu.cn)

Shuguang Cui is with the School of Science and Engineering and the Future Network of Intelligence Institute (FNii), the Chinese University of Hong Kong, Shenzhen; with the Guangdong Provincial Key Laboratory of Future Networks of Intelligence; with Peng Cheng Laboratory; and also with Shenzhen Research Institute of Big Data. (e-mail: shuguangcui@cuhk.edu.cn).

The work is supported in part by the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, by National Natural Science Foundation of China (Grant No. 62102342), by Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515012668), by Shenzhen Science and Technology Program (Grant No. RCBS20221008093120047), by Shenzhen Outstanding Talents Training Fund 202002, by Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), by Young Elite Scientists Sponsorship Program by CAST (Grant No. 2022QNRC001) and by The Major Key Project of PCL Department of Broadband Communication.

I. Introduction

RECENT years have witnessed the booming development of virtual reality (VR), which provides users an immersive and interactive experience to enjoy the virtual world. As one of the most important manifestations, 360° videos bring a new watching experience that has been widely adopted in many multimedia applications such as gaming, education, tourism, and sports. It is also a key technology that supports the development of the new paradigm *Metaverse*. Cisco Mobile Visual Networking Index (VNI) Forecast [1] indicates that 360° videos mobile data traffic will grow nearly 12-fold from 2017 to 2022. According to a recent market research report published on Globe Newswire [2], the global market size was USD 4.42 billion in 2020, exhibiting a significant growth of 42.2% compared to the average year-on-year growth during 2017-2019. They also predict that the market is projected to grow to USD 84.09 billion in 2028 at a compound annual growth rate (CAGR) of 44.8% in the 2021-2028 period.

In 360° videos, users usually need to wear a professional VR headset like HTC VIVE or simple VR-compatible mobile devices such as Google Daydream. How to stream 360° videos have recently captured great attention in academia [3]–[5]. Given the spherical features of 360° videos, a user is allowed to freely move his/her head and eye gaze to watch the most attractive portion of the video. And such a portion is called Field of View (FoV). Given the huge bandwidth overhead of transmitting the whole spherical angles of a 360° video, tile-based HTTP live streaming is widely used to balance the video transmission and user QoE [6]–[8]. With this standard, the video segment can be divided into many tiles with different qualities so that we can assign high-quality content in the FoV while low quality (or even blank content) for the rest of the video. However, the user's FoV can be quite dynamic and affected by many factors such as personalized watching preferences and video content, making it difficult to achieve accurate FoV prediction.

Besides, providing high-bandwidth and low-latency streaming to support multi-user viewing is also critical to 360° video watching. The development of 5G and mobile edge computing (MEC) provides a promising opportunity. The geo-distributed edge servers, e.g., base stations, can be used to cache the frequently requested tiles, which can significantly decrease network latency and bandwidth consumption, thus both improving the users' QoEs and saving the service cost [9]–[12]. Designing such a caching system, however, is non-trivial. Given an edge node serves multiple users within the proximity, how to satisfy the users' overall QoE with fairness into account

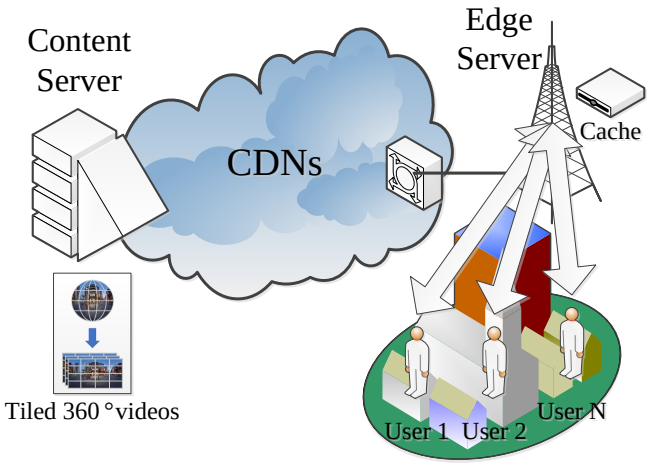


Figure 1. Multi-user 360° Video Streaming on the Edge

and maintain a long-term cost-effective strategy to avoid back and forth replacement is still a challenging problem.

Existing solutions to these challenges are inadequate because they do not adequately address the problem from a holistic perspective. Most works [13]–[16] in this field focus on a single client and use a non-cooperative approach to video requesting, which leads to unnecessary waste of bandwidth and computation resources. Other efforts [9], [10], [17], [18] aim to improve the cache hit rate for multiple users by using information from FoV data, but these approaches are often short-sighted because they only consider the current state without considering long-term planning. Additionally, these works often prioritize user QoE at the expense of users whose FoVs are significantly different from the majority of users. This can lead to suboptimal performance for these maverick users. Furthermore, these works do not consider the potential benefits of collaborative FoV prediction for multiple users, which could improve performance for all users.

To this end, we propose `Ebublio`[1], a novel intelligent edge caching framework that aims to optimize user QoE and system cost by combining user FoV prediction and 360° video tile caching. `Ebublio` consists of two modules: a collaborative FoV prediction (CFP) module and a long-term tile caching optimization (LTO) module. We first conduct a comprehensive spatial and temporal analysis of the user FoV trajectory and find it is highly correlated with the video content, the target user's historical trajectory, and the FoV of other users who have watched the video. Thus, we design an intelligent CFP module that integrates these three features toward collaborative FoV prediction. This collaborative approach allows for more accurate FoV prediction and improved QoE for users. Once the FoV prediction has been made, the LTO module determines a caching policy that maximizes overall user QoE and minimizes system cost. It uses the Lyapunov framework, dual composition, and subgradient descent to solve the long-term optimization problem and determine the optimal caching policy.

[1]Ebublio is an incantation in *Harry Potter* to entrap targets(video tiles) in a bubble(edge cache).

We have conducted extensive evaluations and the real-trace driven experiments show that compared to state-of-the-art solutions `Ebublio` can achieve 36% improvement in QoE under similar traffic consumptions and outperforms in FoV prediction.

The contributions and novelty can be summarized as follows.

- We propose a novel framework that combines user field of view (FoV) prediction and video tile caching with edge computing to improve quality of experience (QoE) and economic efficiency for 360° video streaming for multi-users.
- We propose a collaborative FoV prediction architecture that integrates video content, user trajectory, and other users' records for combined prediction, resulting in accurate and robust results.
- We formulate a video tile caching problem that considers both QoE maximization and cost minimization, and solve it effectively using Lyapunov optimization and subgradient descent. This allows for the optimization of both QoE and economic efficiency in 360° video streaming.

The rest of this paper is organized as follows. Related works are presented in Section II. Section III formulates the whole system. Section IV and Section V give detailed descriptions for collaborative FoV prediction and long-term tile caching optimization respectively. Section VI evaluates the experimental results and Section VII concludes the paper.

II. Related Work

The main research related to our work can be divided into three areas. The first part is about 360° video streaming. Then we show previous efforts on FoV prediction. At last, we present works about edge computing and caching for video streaming.

A. 360° Video Streaming

360° video is a new kind of video representation that is recorded by omnidirectional cameras and can provide an immersive and interactive watching experience to viewers. The video content forms as a sphere rather than a plane, where the viewer can freely adjust his/her head angle to watch a portion of content in the sphere. But in the stage of coding, the sphere is actually projected to a 2D plane using such projection methods as equirectangular or cube-map projections because existing encoders work on 2D rectangles.

Given such spherical features, streaming 360° videos requires much higher network bandwidth than traditional videos. To accommodate the high resource consumption, tile-based video encoding/streaming together with FoV-adaptive tile selection is proposed. Each frame of a video is split into different tiles and only tiles inside the user's FoV are streamed at high quality [6]–[8], [19], [20]. Petrangeli et al. [21] show an HTTP/2-based adaptive streaming framework to achieve higher performance. Nasrabadi et al. [22] propose a Scalable Video Coding encoding method so that the number of video rebuffering events can be reduced. Zare et al. [23] use the motion-constrained tile set feature of High Efficiency Video

Coding standard to tackle the problem of multiple decoders at the user side to decode each tile. Guan et al. [24] leverage the 360° video-specific factors and Dasari et al. [25] use super Super-Resolution to save the bandwidth.

Hou et al. [16] propose a predictive adaptive streaming approach for mobile 360° and VR experiences. And Fei et al. [26] focus on the evaluation of QoE for 360° video transmission, including online, offline and mixed scenarios that can meet the requirement of real applications.

B. FoV Prediction

Accurate FoV prediction for 360° video is the key to tile-based adaptive streaming where many pioneer efforts have been made toward this goal. Many works [27], [28] use regression-based methodologies to predict the future FoV according to the historical trajectory, but they are not quite capable of capturing the inherent correlations. Xie et al. [29] cluster users periodically based on the head movement trajectory and assign new users to the existing clusters to do the prediction. Sun et al. [11] propose a flocking-based methodology for a live 360° video streaming where a large number of users are available concurrently. DRL360 [14] and SR360 [30] introduce deep reinforcement learning frameworks to predict FoV.

The above works are just based on historical trajectory and other works take video content into consideration. Most of those works analyze the video contents through a saliency map that shows the properties of an image at the pixel level. Fan et al. [31] use LSTM based model that learns the sensor-related features and image saliency map to predict viewer fixation. Nguyen et al. [32] propose PanoSalNet to learn the saliency map from user FoV data using DCNN and uses the LSTM network to predict the FoV. Park et al. [33] use the same inputs and find a tile probability map by a CNN + LSTM network. Besides the saliency map, PARIMA [34] uses YOLOv3 [35] to detect the objects and obtain their bounding box coordinates, then predict the FoV based on the track of objects.

C. Edge Computing and Caching

The emergence of edge computing provides a new compute paradigm for multimedia streaming that the videos can be cached or instantly processed at the distributed edge servers for better services. In this way, the end-to-end latency, bandwidth consumption, and energy consumption can be reduced for high-quality video streaming. Hou et al. [36] shift extensive rendering to edge servers to address the challenge from bitrate and latency requirements. Mangiante et al. [37] take advantage of mobile edge computing (MEC) to process and render FoV in order to optimize bandwidth consumption and battery utilization. Chakareski et al. [38] study the delivery of 360°-navigable videos to 5G VR/AR wireless clients in future cooperative multi-cellular systems.

Cheng et al. [15] took a holistic approach to video coding, proactive caching, computation offloading, and data transmission. Maniotis et al. [13] focused on the live-streaming scenario. But their works are only for a single user.

Teng et al. [18] looked at a massive MIMO system with multiple users in a single-cell theater, with a focus on wireless communication. Maniotis et al. [17] used the Deep Q-Network (DQN) algorithm for cache policy. But their work did not consider the multi-user problem in terms of FoV prediction.

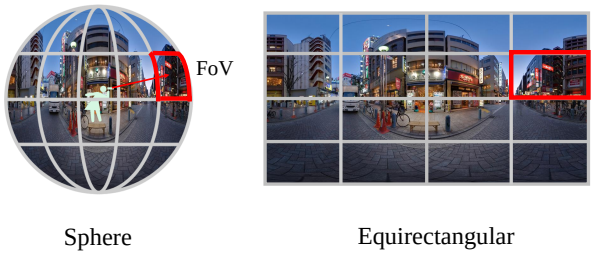


Figure 2. Equirectangle Projection

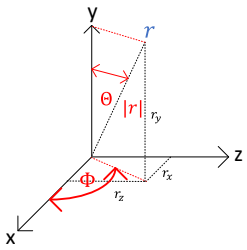


Figure 3. Sketch of ϕ and θ

The challenge in existing works is how to effectively and fairly serve multiple users within proximity at an edge node while maintaining a cost-effective strategy to avoid frequent replacement with an accurate multi-user-based FoV prediction.

III. System Design

In this section, we first clarify the model of the 360° video and then give the formal formulation of the problem. Finally, our proposed design `Ebublio` is presented.

A. Model of 360° Video

We consider a video-on-demand (VoD) scenario that which all the requested videos are already pre-recorded. Each frame is an equirectangular representation of the 360° spherical view as shown in Fig. 2, and is divided into tiles. For clarity of presentation, we suppose tiles are non-overlapping and each tile is encoded, cached, and transmitted separately.

The following shows how to transform spherical data (qx, qy, qz, qw), which are unit quaternions representing the rotation, into equirectangular data (x, y), which are two-dimensional points.

The unit vector $\mathbf{r} = (r_x, r_y, r_z)$ is calculated by unit quaternion from following equations:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} 2 \times qx \times qz + 2 \times qy \times qw \\ 2 \times qy \times qz - 2 \times qx \times qw \\ 1 - 2 \times qx^2 - 2 \times qy^2 \end{bmatrix} \quad (1)$$

By unit vector, we can calculate the angle between x-axis, y-axis shown in Figure 3 by

$$\varphi = \arctan\left(\frac{r_y}{r_x}\right), \theta = \arccos\left(\frac{r_y}{|r|}\right) \quad (2)$$

Then we can calculate (x, y) by

$$y = \begin{cases} height - height \times \left(\frac{\theta}{2\pi}\right), \text{if } r_y \geq 0 \\ height \times \left(\frac{\theta}{2\pi}\right), \text{if } r_y < 0 \end{cases} \tag{3}$$

$$x = \begin{cases} \frac{3width}{4} + sgn(r_x)\frac{\varphi}{2\pi} \times width, \text{ if } r_z > 0 \\ \frac{width}{2} - sgn(r_x)\frac{\varphi}{2\pi} \times width, \text{ if } r_z < 0 \end{cases} \tag{4}$$

where $height$ and $width$ are the height and width of the video respectively, and $sgn()$ is the sign function.

B. Problem Formulation

Suppose that the caching edge will make a cache decision for every time duration T_0. And at timeslot t, the set of video chunks cached in the edge is denoted as $\mathcal{S}_t$.

The quality of a user's Quality of Experience (QoE) is mainly determined by two factors: the quality of the video and the delay. In the scenario discussed in this paper, a video is made up of multiple tiles that can either be cached or not. When a tile is cached, it will have a higher quality and a lower delay. If a tile has not been cached, it must be fetched from a remote server, which can result in lower video quality and longer delays.

Under the same edge, if the tile has been cached (or not), it leads to a similar impact toward QoE. The main difference is whether the tile has been cached, which can lead to a significant difference in QoE. For this reason, we can simplify the impact of each tile (including its quality and delay) on the QoE model by categorizing them into two cases: cached or not.

Since the goal of this paper is to design a cache policy that determines which tiles should be cached based on their predicted future FoV, this simplification allows us to focus on the cache policy itself. And note that this simplified QoE model can be easily extended to consider the quality and delay of each tile if needed.

We first define $a_{t,n}$ and $b_{t,n}$ as the number of tiles in FoV at timeslot t which have been cached and haven't been cached, respectively. Thus we can define the QoE of user n as

$$U_{t,n} = \frac{a_{t,n} + \alpha b_{t,n}}{p} \tag{5}$$

where p is the number of tiles in the FoV. The QoE has a proportional relationship to how many tiles in FoV are cached, i.e., the QoE is 1 if all tiles in FoV are cached and a smaller value if some tiles are missed because fetching from a remote content server usually leads to a lower bitrate and larger delay. The parameter α is the discount factor to adjust the cache missing penalty.

The traffic used between edge and content server comes from two parts. One is that the active cache at each step needs to download the extra video tiles that are different from the previous time step. The other is the extra traffic cost when a user is requesting video tiles that are not cached in the edge. Thus, the traffic used can be defined as:

$$B_t = H_1(\mathcal{S}_{t+1} - \mathcal{S}_t) + H_2(b_{t,n}) \tag{6}$$

where $H(.)$ denotes the traffic cost. Specifically, $H_1(.)$ denotes the traffic cost brought from the update of cache policy, and $H_2(.)$ denotes the traffic cost for tiles needed but not cached.

The objective of the cache manager is to optimize the system performance, i.e., maximize the QoE and minimize the traffic cost, by finding the best cache policy $\mathcal{S}$. Thus this problem can be formulated as

$$objective: \min_{\mathcal{S}_t} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left[\left(-\sum_n U_{t,n} \right) + \beta B_t \right] \tag{7}$$

$$s.t. 0 \leq \sum \mathcal{S}_t \leq C \tag{8}$$

$$B_t \leq B_{max,t} \tag{9}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} B_t \leq \eta_1 \tag{10}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} U_{t,n} \geq \eta_2, \forall n \tag{11}$$

where the objective function shown in Equation 7 is long-term expected QoE for all users with the traffic consumption. Equation 8 is the cache size constraint; Equation 9 is the bandwidth constraint in each timeslot; Equation 10 is the individual long term expected traffic consumption constraint; Equation 11 is the individual long term expected QoE for a single user, to avoid one user suffer an extremely bad QoE.

C. QoE for Different Users

The QoE of different users served by the same edge server has correlations due to the limited resources at the edge server, specifically the cache size.

For example, consider two extremes: if all users under the same edge server are watching the same video with the same trajectory at the same time, caching the tiles that will be viewed will benefit all users. However, if all users are watching different parts of the video, they will compete for the limited resources at the edge server, and the benefits to one user may come at the expense of another user's QoE.

In real-world situations, most users under the same edge server tend to behave similarly. However, there may be some "outliers" whose behavior is significantly different from that of other users. If we only focus on maximizing the overall quality of experience, these outliers may end up having the worst viewing experience, which is unfair. Our model takes into account the fairness for these minority users by implementing a minimum QoE mechanism in Equation 11.

D. System Architecture

We propose `Ebublio` which consists of two main modules: a collaborative FoV prediction (CFP) module and a long-term tile caching optimization (LTO) module. The user FoV trajectory is highly correlated with the video content, the target

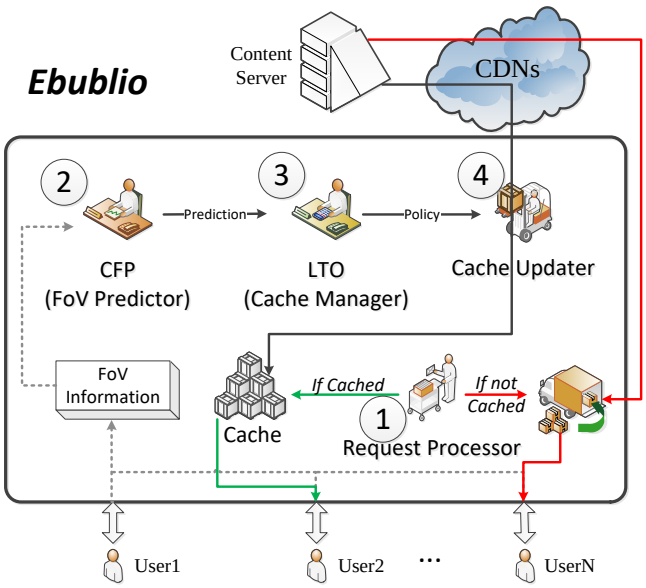


Figure 4. The Structure of Ebublio

user's historical trajectory, and the FoV of other users who have watched the video. Thus, we design an intelligent CFP module that integrates these three features toward collaborative FoV prediction. After the FoV prediction, the LTO module then seeks a caching policy that both maximized the overall user QoE and minimizes the system cost. It solves the long-term optimization problem with both the Lyapunov framework and dual composition as well as subgradient descent. The detail of CFP and LTO will be described in section IV and section V separately.

Fig. 4 shows the structure of Ebublio and the procedure shows as follows

1) At timeslot t, users request the next chunk, if the requested tile is in the cache, return it. If not, the edge downloads it from the content server at a lower bitrate.
2) CFP predicts future FoVs of each user and sends the results to LTO.
3) Based on the prediction, LTO calculates a cache policy.
4) Update the cache according to the policy.

IV. CFP: Collaborative FoV Prediction

Fig. 5 shows the structure of CFP. It predicts FoV based on multiple perspectives: historical trajectory, video content, and shared FoVs. Each component first works individually and then calculates its weighted average, where the weights are generated by the Passive-Aggressive Regression model.

A. Historical Trajectory Based Prediction

If we only consider historical trajectory, it's a time series prediction problem. So we leverage the Long Short-Term Memory (LSTM) [39] model to predict. The prediction process can be formulated as below

$$\begin{cases} X_t^{LSTM} = LSTM(X_0, X_1, \cdots, X_{t-1}; \phi_X) \\ Y_t^{LSTM} = LSTM(Y_0, Y_1, \cdots, Y_{t-1}; \phi_Y) \end{cases} \tag{12}$$

where ϕ_X and ϕ_Y are the LSTM network's parameters.

360° video is commonly in the form of equirectangular projection (ERP). This projecting method is straightforward to comprehend. However, the distribution of the pixels is highly unbalanced. The pixels around the polar are dense, while the pixels around the equator are sparse. But through our analysis, we found that while watching 360° videos, users mainly watch the center of the videos. And more importantly, two polar of the videos are hardly ever watched. So, the distortion of ERP doesn't matter here.

B. Video Content based Prediction

Hypothesis 1. *Users' FoVs are related to the video content.*

To do the prediction based on video content, we use YOLOv3 [35] to detect objects at each frame. Each object has a unique ID but new objects will appear and old objects may disappear permanently or temporarily. And the objects may be discontinuous because of the limited viewport. To solve the issue, we use a spherical projection algorithm to index objects robustly. The algorithm will compute the centroids of objects and convert objects into the spherical projection, which will orient and index the objects exactly. Then we predict the FoV based on the location of those objects.

C. Shared FoVs based Prediction

When predicting the FoV in timeslot t for user i, the system looks for users who have watched and uses a weighted average to do to prediction. The prediction process can be formulated as below

$$\left(X_t^{Share}, Y_t^{Share}\right) = \sum_{n=1}^{W} \rho_n \left(X_{t,n}, Y_{t,n}\right) \tag{13}$$

where W is users who have watched this video or is watching afterward of the video, $(X_{t,n}, Y_{t,n})$ is user n's centroid coordinate of FoV, and ρ_n is the weight. To determine the weights, we first give an hypothesis.

Hypothesis 2. *Users with a more similar trajectory in the past are more likely to have similar FoV in the future.*

We use the distance to calculate similarity, the similarity between user i and user n can be denoted as below

$$s(i,n) = \sum_{f=0}^{F} \frac{d_f(i,n)}{\|F\|} \tag{14}$$

where T is a set of frames that i and n both have watched in the past. $d_f(i,n)$ is the distance of centroid coordinates of FoVs. To avoid the sudden change in the boundary, we use great circle sphere distance.

The reversed sigmoidal function defined below is our choice to calculate the unnormalized weight from similarity.

$$\omega_n = \mathcal{H}(s(i,n)) = \frac{e^{-\gamma(s(i,n)-\phi)}}{1+e^{-\gamma(s(i,n)-\phi)}} \tag{15}$$

where γ and ϕ are parameters needed to be tuned.

The weights are normalized as:

$$\rho_n = \frac{\omega_n}{\sum_{n=1}^{W} \omega_n} \tag{16}$$

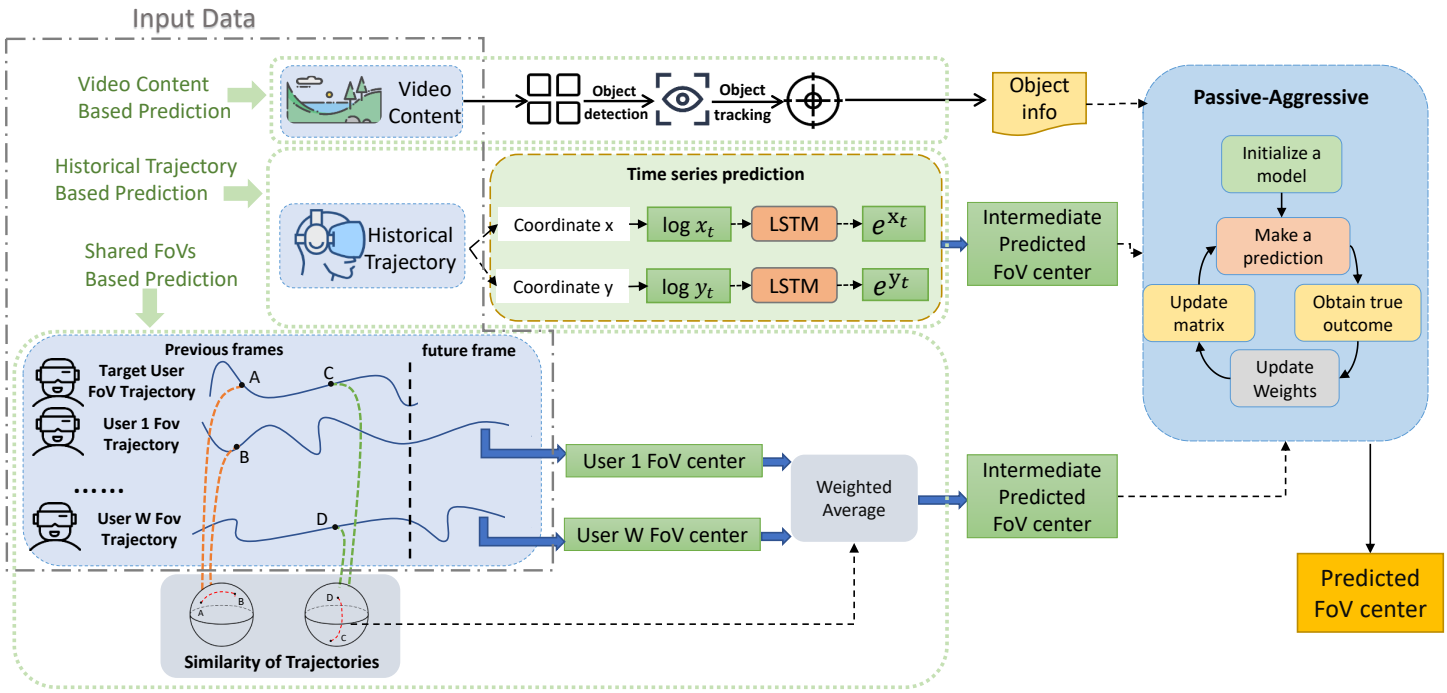


Figure 5. The Structure of CFP

D. Weights Assignment

We use Passive-Aggressive Regression model [40], which is an efficient online learning regression algorithm, to assign weights for the above intermediate results. The algorithm computes the mapping $f : \mathbb{R}^n \to \mathbb{R}, f(\boldsymbol{x}; \theta) = \theta^T \boldsymbol{x}$ where, paramters θ, predictors $\boldsymbol{x} \in \mathbb{R}^n$. The algorithm uses the Hinge Loss Function:

$$L(\theta, \epsilon) = max(0, |y - f(\boldsymbol{x}_t; \theta)| - \epsilon) \tag{17}$$

where y is the actual value of the response varible. ϵ is a tolerance value for prediction errors. The Passive-Aggressive Regression model updates θ according to the following equation:

$$\theta^{t+1} = \theta^t + \zeta \frac{L(\theta, \epsilon)}{\|\boldsymbol{x}_t\|^2 + \frac{1}{2C}} sign(y - \theta^T \boldsymbol{x}_t) \boldsymbol{x}_t \tag{18}$$

We run two Passive-Aggressive Regression models to predict x and y coordinates respectively. The equations for the prediction for next timeslot are given by:

$$\begin{cases} X_t = \theta_{0x} + \theta_{1x} X_t^{LSTM} + \theta_{2x} X_t^{Share} + \sum_{i=1}^{N_{obj}} \boldsymbol{\theta}_{3X} O_{Xit} \\ Y_t = \theta_{0y} + \theta_{1y} Y_t^{LSTM} + \theta_{2y} Y_t^{Share} + \sum_{i=1}^{N_{obj}} \boldsymbol{\theta}_{3Y} O_{Yit} \end{cases} \tag{19}$$

where (X_t, Y_t) is the predicted centroid coordinate of FoV. (X_t^{LSTM}, Y_t^{LSTM}) and $(X_t^{Share}, Y_t^{Share})$ are the intermediate results obtained from section IV-A and section IV-C respectively. (O_{Xit}, O_{Yit}) is the coordinates for the i^{th} object according to section IV-B.

V. LTO: Long-term Tile Caching Optimization

This section shows the detail of LTO. We first leverage Lyapunov optimization to transform the long-term optimization problem into a problem in one timeslot, then solve this problem by dual decomposition and subgradient descent.

A. Lyapunov Optimization

The optimization problem involves the long-term expected terms, so we seek the technique of Lyapunov optimization.

Firstly, we introduce two virtual queues

$$F_{t+1} = max\{F_t + B_t - \eta_1, 0\} \tag{20}$$

and

$$G_{t+1,n} = max\{G_{t,n} - U_{t,n} + \eta_2, 0\}, \forall n \tag{21}$$

Lemma 1. *If the virtual queue F_t and $G_{t,n}$ are rate stable, i.e.,*

$$\lim_{T\to\infty} \frac{F_T}{T} \le 0, \lim_{T\to\infty} \frac{G_{T,n}}{T} \le 0, \tag{22}$$

then we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} B_t \le \eta_1, \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} U_{t,n} \ge \eta_2. \tag{23}$$

Proof. We first rewrite F_{t+1} as below

$$F_{t+1} = \begin{cases} F_t + B_t - \eta_1, & \text{if } F_t \le -B_t + \eta_1 \\ 0, & \text{if } F_t < -B_t + \eta_1 \end{cases} \tag{24}$$

Then, we have

$$\begin{aligned} F_{t+1} - F_t &= \begin{cases} B_t - \eta_1, & \text{if } F_t \leq -B_t + \eta_1 \\ -F_t, & \text{if } F_t < -B_t + \eta_1 \end{cases} \\ &= max\{B_t - \eta_1, -F_t\} \\ &\geq B_t - \eta_1 \end{aligned} \tag{25}$$

So we have

$$\begin{aligned} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} F_{t+1} - F_t &\geq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} B_t - \eta_1 \\ \lim_{T\to\infty} \frac{F_T}{T} &\geq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} B_t - \eta_1 \end{aligned} \tag{26}$$

Because the virtual queue is rate stable, then we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} B_t \leq \eta_1 \tag{27}$$

Similarly, we can show that

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} U_{t,n} \geq \eta_2 \tag{28}$$

if $G_{t,n}$ is rate stable. □

By assuming the virtual queues F_t and $G_{t,n}$ are rate stable, the origin problem can be rewritten as:

$$objective: \min_{\mathcal{S}_t} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left[\left(-\sum_n U_{t,n} \right) + \beta B_t \right] \tag{29}$$

$$s.t. 0 \leq \sum \mathcal{S}_t \leq C \tag{30}$$

$$B_t \leq B_{max,t} \tag{31}$$

$$\lim_{T\to\infty} \frac{F_t}{T} \leq 0 \tag{32}$$

$$\lim_{T\to\infty} \frac{G_{t,n}}{T} \leq 0, \forall n \tag{33}$$

Then we can utilize the penalty drift in Lyapunov optimization to solve Equation 29. We first introduce a concatenated vector of the virtual queues as

$$\Theta_t \triangleq [F_t, G_{t,1}, \cdots, G_{t,N}] \tag{34}$$

The corresponding Lyapunov function can be defined as

$$L(\Theta_t) \triangleq \frac{F_t^2}{2} + \frac{G_{t,1}^2}{2} + \cdots + \frac{G_{t,N}^2}{2} \tag{35}$$

Then, the Lyapunov penalty drift $\Delta(\Theta_t)$ can be obtained as follows

$$\Delta(\Theta_t) = E\left[L(\Theta_{t+1}) - L(\Theta_t) | \Theta_t \right] \tag{36}$$

According to the Lyapunov optimization theory, enforcing the virtual queue constraint is equivalent to minimize the drift penalty, and minimizing the objective function with the virtual queue constraints is equivalent to minimize the "drift-plus-penalty" defined as follows

$$\Delta(\Theta_t) + V \times E\left[\left(-\sum_n U_{t,n} \right) + \beta B_t \middle| \Theta_t \right] \tag{37}$$

Algorithm 1 Cache Management Based on Dual Decomposition

Input: the number of time slots T, total cache size C, the number of tiles E, step size δ, threshold ϵ
Output: Cache Policy

```
Initialize F'_1, G'_1
for t = 1 to T do
    Initialize λ
    while |Δλ| > ε do
        for e = 1 to E do
            if e is in the predicted FoV then
                Solve Equation 53 to obtain γ*_t(e)
            end if
        end for
        λ = [λ − δ(C − Σ_{e=1}^{E} γ*_t(e))]^+
        Δλ = −δ(C − Σ_{e=1}^{E} γ*_t(e))
    end while
    Update F'_{t+1}, G'_{t+1}
end for
```

where $V \geq 0$ is the penalty weight, which represents the importance of the objective function compared to the virtual queue constraints.

Thus, the optimization problem can be rewritten as

$$objective: \min_{\mathcal{S}_t} \left\{ \Delta(\Theta_t) + V \times E\left[\left(-\sum_n U_{t,n} \right) + \beta B_t \middle| \Theta_t \right] \right\} \tag{38}$$

$$s.t. 0 \leq \sum \mathcal{S}_t \leq C \tag{39}$$

$$B_t \leq B_{max,t} \tag{40}$$

B. Dual Decomposition

A 360° video can be divided into E tiles both spatially and temporally. For convenience, we define two indicator variables

$$\gamma_t(e) = \begin{cases} 1, & \text{if } e \text{ in } \mathcal{S}_t \\ 0, & \text{if } e \text{ not in } \mathcal{S}_t \end{cases} \tag{41}$$

$$FoV_{t,n}(e) = \begin{cases} 1, & \text{if } e \text{ in } n\text{'s FoV} \\ 0, & \text{if } e \text{ not in } n\text{'s FoV} \end{cases} \tag{42}$$

Then we can rewrite $U_{t,n}$ and B_t as

$$U'_{t,n} = \frac{FoV_{t,n}(e)\left[\gamma_t(e) + \alpha'(1 - \gamma_t(e))\right]}{\|FoV_{t,n}\|} \tag{43}$$

and

$$B'_t = \gamma_{t+1}(e)\left[1 - \gamma_t(e)\right] \tag{44}$$

where for convience, we combine the term denoted the traffic consumption caused by uncached but requested tiles in $U'_{t,n}$ by changing α'.

So the Equation 38 can be written as

$$objective: \max_{\gamma_t(e)} \sum_{e}^{E} \left\{ \Delta(\Theta'_t) + V \times E\left[\left(-\sum_{n} U'_{t,n} \right) + \beta B'_t \middle| \Theta_t \right] \right\} \tag{45}$$

$$s.t. 0 \leq \sum_{e}^{E} \gamma_t(e) \leq C \tag{46}$$

$$B_t \leq B_{max,t} \tag{47}$$

This optimization problem is clearly non-convex, to overcome this obstacle, we initiate the use of specific functions as described below

$$\begin{cases} c_0(\gamma_t(e)) = \Delta(\Theta'_t) + V \times E\left[\left(-\sum_n U'_{t,n} \right) + \beta B'_t \middle| \Theta_t \right] \\ c_1(\gamma_t(e)) = B_t - B_{max,t} \\ c_2(\gamma_t(e)) = \sum_e^E \gamma_t(e) - C \end{cases} \tag{48}$$

By utilizing Equation 48, the optimization problem presented in Equation 45 can be re-presented in a succinct manner as demonstrated below

$$objective: \max_{\gamma_t(e)} \sum_{e=1}^{E} c_0(\gamma_t(e)), \tag{49}$$

$$s.t.\ c_1(\gamma_t(e)) \leq 0, \tag{50}$$

$$c_2(\gamma_t(e)) \leq 0 \tag{51}$$

Next, we employ the technique of dual decomposition to address the complex coupling optimization problem. This approach transforms the original problem into a centralized master problem and several distrubuted subproblems. To begin with, we establish the definition of the Lagrange multiplier λ, which plays a crucial role in this process, as follows:

$$\min_{\lambda} \sum_{e=1}^{E} \psi_e(\lambda) - \lambda C,\ s.t. \lambda \geq 0 \tag{52}$$

where the subproblem for each tile is shown as follows

$$\psi_e(\lambda) = \sup_{\gamma_t(e)} \left\{ c_0(\gamma_t(e)) + \lambda \gamma_t(e) \middle| c_1(\gamma_t(e)) \leq 0 \right\} \tag{53}$$

For any given λ announced by the master problem, the edge can choose which tiles to cache by finding the near-optimal value $\gamma_t^*(e)$ by solving Equation 53. Then the Lagrange multiplier λ can be updated by the subgradient of the master problem Equation 52 shown as follows

$$\lambda_{K+1} = \left[\lambda_K - \delta(C - \sum_{e=1}^{E} \gamma_t^*(e)) \right]^+ \tag{54}$$

where δ is a positive stepsize, K is the iteration index, and $[.]^+$ denotes the projection onto the non-negative orthant.

Iterating all video items and tiles will use up a lot of computing power, making it difficult to deploy the system on an edge server. However, not all tiles need to be calculated. Only the tiles that appear in the predicted FoV are necessary, which is a small fraction compared to the total number of video items and tiles. Therefore, we only need to iterate over those effective tiles.

Algorithm 1 shows the procedure.

VI. Evaluation

This section discusses the experiments that we have performed to demonstrate the superiority of `Ebublio`. We first evaluate the whole system, then we analyze the performance of each module.

A. Settings

Here we give a brief description of the settings.

We use the dataset collected by Wu et al. [41], which is a popular dataset that has nine videos watched by 48 users with an average view duration of 164 seconds. Each trace of the head tracking logs for the dataset consists of unit quaternions along with the timestamp. We split the videos into 8×8, with a total of 64 tiles, in a generally used setting.

Different chunk sizes towards trade-offs between the streaming time and the prediction accuracy. A smaller chunk size facilitates better prediction accuracy. On the other hand, streaming time would have to be low for the chunks to ensure smooth streaming of video without buffering. The network latency of a 3G network is about 150ms [42]. To apply our model even in a poor network situation, we set the chunk size as 1 second.

And we set the hyper parameters of the Passive-Aggressive Regression model as $C = 0.01, \epsilon = 0.001$, which is achieved by hyper parameter tuning.

B. Performance of the Whole System

To evaluate the performance of `Ebublio`, we compare it with baselines as described below:

- **QoE-Greedy:** For this method, we let `Ebublio` greedily optimize QoE, ignoring traffic consumption.
- **Most Frequently Used (MFU):** Similar to the concept Least Frequently Used in passive cache paradigm, this method actively prefetch tiles used most frequently to replace those fewer ones.
- **CooPEC [9], [10]:** CooPEC prefetch one tile based on FoV prediction to replace another based on previous FoV information. Instead of the Naïve-Bayes algorithm, which is originally used in CooPEC to do FoV prediction, we replace it with the CFP module in `Ebublio`, which is more accurate. CooPEC also works on reducing core network bandwidth consumption and enhancing QoE for users on the edge. As far as we know, it's a state-of-the-art framework.

We set the cache size as 20% of the size of the whole 360° video in a chunk of time.

Fig. 6 shows the comparisons of traffic consumption for different methods. In `Ebublio`, by changing β in Equation 7, the importance of QoE and traffic cost can be balanced. Here we set β to let the traffic consumption of `Ebublio` to be similar to CooPEC and MFU to compare the QoE.

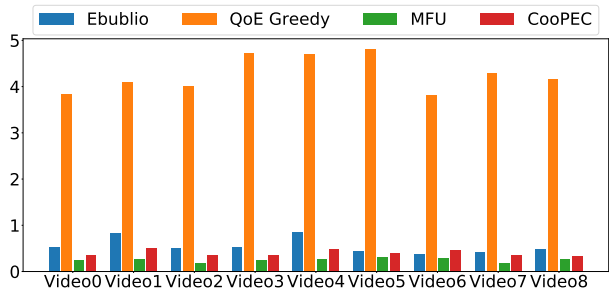


Figure 6. Traffic Consumption Comparisons of Ebublio and Baselines

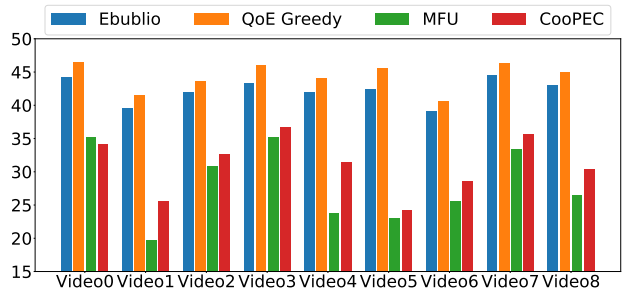


Figure 7. QoE Comparisons of Ebublio and Baselines

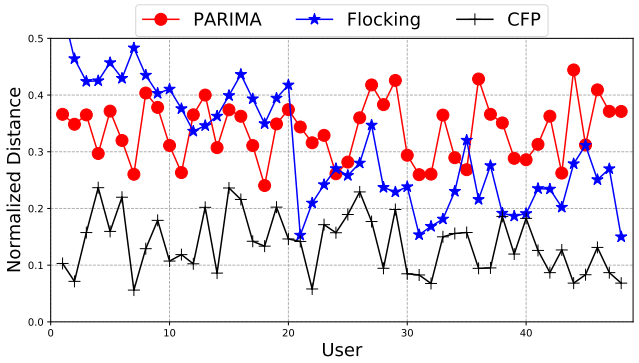


Figure 8. Comparasions of CFP, PARIMA, and Flocking

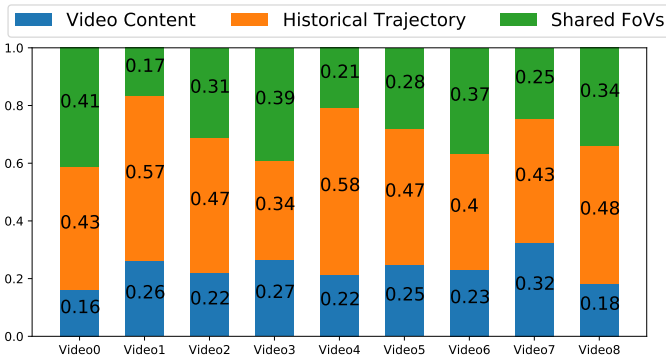


Figure 9. Contributions of Components in CFP

Fig. 7 shows the comparisons of QoE for different methods. From the figure, we can find that the QoE of QoE-Greedy is slightly higher than Ebublio, but the traffic consumption is nearly 10 times larger. It's because QoE-Greedy updates the cache frequently and severely to achieve better QoE. Both these two methods are much better than MFU and CooPEC from the perspective of QoE.

In particular, the QoE of Ebublio outperforms CooPEC 36%. The reason mainly comes from two aspects: (1) CooPEC is designed for a single user, ignoring the multi-user situation and (2) CooPEC hasn't considered the long-term optimization problem. And in most videos, CooPEC works better than MFU, where the reason is that CooPEC predicts the future FoV.

C. Performance of CFP

To evaluate the performance of CFP, we compare it with baselines as described below:

- **Flocking [11]:** This method utilizes the actual FoV information of users in the front of the flock to predict users behind them.
- **PARIMA [34]:** PARIMA is an augmentation of ARIMA which combines video content to predict.

We use the normalized distance between the predicated FoV and the actual FoV to evaluate the algorithms. With the smaller distance, the prediction is better. Fig. 8 shows the results. The horizontal axis is the user number, and for each user, he/she can use the actual information of users whose number is smaller than him/her.

From the figure, we can find that when the user number is small,i.e., the number of actual information that can be used is limited, PARIMA performs better than Flocking. And with the number becoming bigger, Flocking outperforms PARIMA gradually. In fact, PARIMA oscillates besides a certain value because it doesn't use the information of other users. And CFP outperforms greatly compared with Flocking and PARIMA.

D. Analysis of CFP

In section IV, we claim two hypotheses: (1) Users' FoVs are related to the video content. And (2) Users with a more similar trajectory in the past are more likely to have similar FoV in the future. To verify the above hypotheses, we evaluate the contributions of *Video Content* and *Shared FoVs* in predicting the FoV, which determines how significant they are in predicting the FoV of the user.

We extract the contributions from Equation 19 by calculating the proportion of $\theta_1, \theta_2, \boldsymbol{\theta}_3$ in the sum of them. Fig. 9 shows the results for different videos.

We first analyze the results in total. *Video Content*, *Historical Trajectory*, and *Shared FoVs* contribute about 23%,46%, and 30% to the output, separately. We can find that *Historical Trajectory* contributes nearly half and *Shared FoVs* contributes more than *Video Content*.

Specifically, the contribution of *Video Content* varies from 16% to 32%. And the contribution of *Shared FoVs* varies from 17% to 41%. *Video Content* works worst in Video 0 while *Shared FoVs* works best in this video. Video 0 is a stage performance video where several people act at the center of the stage. The reason why *Shared FoVs* works well may be that

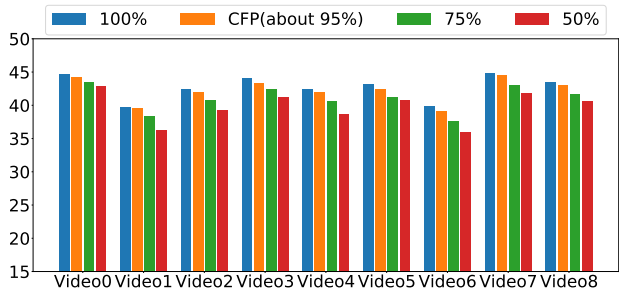


Figure 10. QoE of LTO under Different Prediction Accuracy

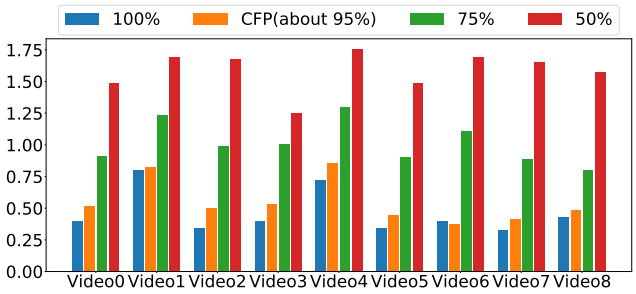


Figure 11. Traffic Consumption of LTO under Different Prediction Accuracy

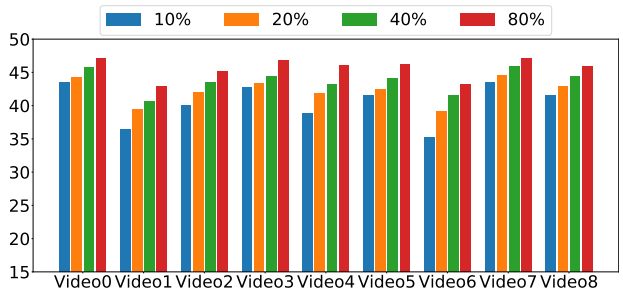


Figure 12. QoE of LTO under Different Cache Size

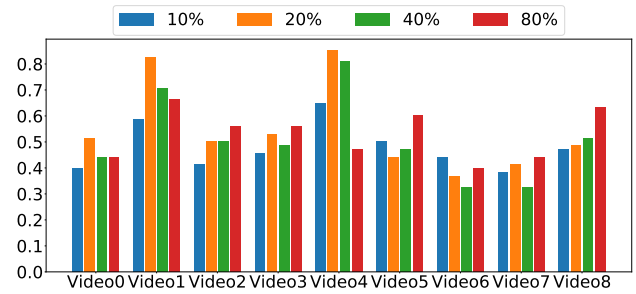


Figure 13. Traffic Consumption of LTO under Different Cache Size

all users focus on the actors. And other objects that appeared in the video, such as the audience, are not important, which is why *Video Content* works worse.

By analyzing the videos and the contribution of *Shared FoVs*, we find that it is highly related to the sparsity of the Region of Interest (ROI). If the ROI of the video is intensive, users' would focus on this ROI, which makes their behaviors more similar, so that *Shared FoVs* works well. On the contrary, in videos whose ROI is sparse, such as Video 1 (freestyle skiing) and Video 4 (Surfing), *Shared FoVs* works worse.

From the results, it is evident that *Video Content*, *Historical Trajectory*, and *Shared FoVs* play important roles together in predicting FoV. Depending on the properties of the videos themselves, the contribution varies.

E. Different Accuracy of Prediction

Instead of CFP, we can also use other prediction algorithms as the input of LTO. In order to thoroughly analyze the performance of LTO at different levels of accuracy, we utilized true FoV data and generated additional FoV data with accuracy levels of 75% and 50% to serve as inputs for our study. The results of this analysis are shown in Fig. 10 and Fig. 11, which illustrate LTO's QoE and traffic consumption under different prediction accuracy levels. Our findings demonstrate that LTO performs well even in situations where the accuracy of its predictions is relatively low.

F. Different Cache Size

Then we change the size of the cache to analyze the performance of LTO. We set the cache size as 10%, 20%, 40%, and 80% of the size of the whole 360° video in a chunk time. Fig. 12 and Fig. 13 show the QoE and traffic consumption of LTO under different cache size.

The QoE is linearly related to the cache size, but interestingly, the traffic consumption is not. In some videos, the consumption of middle-size cache is larger, while in others, it's the contrary. After analyzing the videos, we find that it's related to the video content. The videos which let users watch the same portion such as skiing is the former situation and other videos that users are more likely to watch any place they like are the latter situation. For videos with more similar users' trajectories, the small cache will choose to keep some tiles that would be requested frequently and the large cache can cache most of the frequently requested tiles, so their traffic consumptions are lower. For videos where users' trajectories are different, the small cache cannot find tiles worth being cached for a long time, and for the large cache, after caching the most frequently requested tiles, it still has free cache memory, which will be updated severely.

VII. Conclusion

In this paper, we propose a novel intelligent edge caching framework, named `Ebublio` to optimize overall user QoE with fairness and long-term system cost in 360° video streaming. `Ebublio` consists of a collaborative FoV prediction (CFP) module and a long-term tile caching optimization (LTO) module. CFP integrates the video content, the target user's historical trajectory, and the FoV of other users who have watched the video toward collaborative FoV prediction. After the FoV prediction, LTO then seeks a caching policy that both maximized the overall user QoE and minimizes the system cost. It solves the long-term optimization problem with both the Lyapunov framework and dual composition as well

as subgradient descent. Our evaluation further demonstrates the superiority of our framework, with an average of 36% improvement in QoE compared with a state-of-the-art solution. And our FoV prediction also outperforms the state-of-the-art solution 42%.

REFERENCES

[1] Cisco, "Cisco visual networking index: Forecast and trends," 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] F. B. Insights, "Virtual reality market size, share and covid-19 impact analysis, 2021-2028," 2021. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378

[3] C. Fan, W. Lo, Y. Pai, and C. Hsu, "A survey on 360° video streaming: Acquisition, transmission, and display," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 71:1–71:36, 2019.

[4] Y. Jin, J. Liu, F. Wang, and S. Cui, "Where are you looking?: A large-scale dataset of head and gaze behavior for 360-degree videos and a pilot study," in *Proceedings of the 30th ACM International Conference on Multimedia (MM), Lisboa, Portugal, October 10 - 14, 2022*.

[5] J. Liu, B. Zhu, F. Wang, Y. Jin, W. Zhang, Z. Xu, and S. Cui, "Cav3: Cache-assisted viewport adaptive volumetric video streaming," in *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Shanghai, China, March 25 - 29, 2023*.

[6] M. Graf, C. Timmerer, and C. Müller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: design, implementation, and evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys), Taipei, Taiwan, June 20-23, 2017*.

[7] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proceedings of the IEEE International Symposium on Multimedia (ISM), San Jose, CA, USA, December 11-13, 2016*.

[8] P. Rondao-Alface, J. Macq, and N. Verzijp, "Interactive omnidirectional video delivery: A bandwidth-effective approach," *Bell Labs Tech. J.*, vol. 16, no. 4, pp. 135–147, 2012.

[9] A. Mahzari, A. T. Nasrabadi, A. Samiei, and R. Prakash, "Fov-aware edge caching for adaptive 360° video streaming," in *Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM), Seoul, Republic of Korea, October 22-26, 2018*.

[10] A. Mahzari, A. Samiei, and R. Prakash, "Coopec: Cooperative prefetching and edge caching for adaptive 360° video streaming," in *Proceedings of the IEEE International Symposium on Multimedia (ISM), Naples, Italy, December 2-4, 2020*.

[11] L. Sun, Y. Mao, T. Zong, Y. Liu, and Y. Wang, "Flocking-based live streaming of 360-degree video," in *Proceedings of the 11th ACM Multimedia Systems Conference (MMSys), Istanbul, Turkey, June 8-11, 2020*.

[12] Z. Guo, J. Wang, S. Liu, J. Ren, Y. Xu, and Y. Wang, "Spongetraining: Achieving high efficiency and accuracy for wireless edge-assisted online distributed learning," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.

[13] P. Maniotis and N. Thomos, "Tile-based edge caching for 360° live video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4938–4950, 2021.

[14] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "DRL360: 360-degree video streaming with deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Paris, France, April 29 - May 2, 2019*.

[15] Q. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang, and T. Q. S. Quek, "Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming," *IEEE Trans. Multim.*, vol. 24, pp. 1529–1544, 2022.

[16] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and VR experiences," *IEEE Trans. Multim.*, vol. 23, pp. 716–731, 2021.

[17] P. Maniotis and N. Thomos, "Viewport-aware deep reinforcement learning approach for 360° video caching," *IEEE Trans. Multim.*, vol. 24, pp. 386–399, 2022.

[18] L. Teng, G. Zhai, Y. Wu, X. Min, W. Zhang, Z. Ding, and C. Xiao, "Qoe driven VR 360° video massive MIMO transmission," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 1, pp. 18–33, 2022.

[19] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, May 21-25, 2017*.

[20] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Trans. Multim.*, vol. 18, no. 9, pp. 1819–1831, 2016.

[21] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. D. Turck, "An http/2-based adaptive streaming framework for 360° virtual reality videos," in *Proceedings of the ACM on Multimedia Conference (MM), Mountain View, CA, USA, October 23-27, 2017*.

[22] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proceedings of the ACM on Multimedia Conference (MM), Mountain View, CA, USA, October 23-27, 2017*.

[23] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the ACM Conference on Multimedia Conference (MM), Amsterdam, The Netherlands, October 15-19, 2016*.

[24] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: optimizing 360° video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM), Beijing, China, August 19-23, 2019*.

[25] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proceedings of the 39th IEEE Conference on Computer Communications (INFOCOM), Toronto, ON, Canada, July 6-9, 2020*.

[26] Z. Fei, F. Wang, J. Wang, and X. Xie, "Qoe evaluation methods for 360-degree VR video transmission," *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 1, pp. 78–88, 2020.

[27] 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 IEEE International Conference on Big Data (BigData), Washington DC, USA, December 5-8, 2016*.

[28] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom), New Delhi, India, October 29 - November 02, 2018*.

[29] L. Xie, X. Zhang, and Z. Guo, "CLS: A cross-user learning based system for improving qoe in 360-degree video adaptive streaming," in *Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM), Seoul, Republic of Korea, October 22-26, 2018*.

[30] J. Chen, M. Hu, Z. Luo, Z. Wang, and D. Wu, "SR360: boosting 360-degree video streaming with super-resolution," in *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), Istanbul, Turkey, June 10-11, 2020*.

[31] C. Fan, S. Yen, C. Huang, and C. Hsu, "Optimizing fixation prediction using recurrent neural networks for 360° video streaming in head-mounted virtual reality," *IEEE Trans. Multim.*, vol. 22, no. 3, pp. 744–759, 2020.

[32] A. Nguyen, Z. Yan, and K. Nahrstedt, "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction," in *Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM), Seoul, Republic of Korea, October 22-26, 2018*.

[33] S. K. Park, A. Bhattacharya, Z. Yang, S. R. Das, and D. Samaras, "Mosaic: Advancing user quality of experience in 360-degree video streaming with machine learning," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 1, pp. 1000–1015, 2021.

[34] L. Chopra, S. Chakraborty, A. Mondal, and S. Chakraborty, "PARIMA: viewport adaptive 360-degree video streaming," in *Proceedings of the Web Conference 2021 (WWW), Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*.

[35] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[36] X. Hou, Y. Lu, and S. Dey, "Wireless VR/AR with edge/cloud computing," in *Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, July 31 - Aug. 3, 2017*.

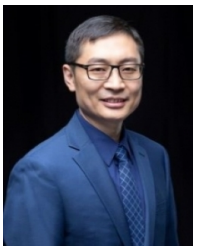
[37] S. Mangiante, G. Klas, A. Navon, G. Zhuang, R. Ju, and M. D. Silva, "VR is on the edge: How to deliver 360° videos in mobile networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, VR/AR Network@SIGCOMM, Los Angeles, CA, USA, August 25, 2017*.

[38] J. Chakareski, "VR/AR immersive communication: Caching, edge computing, and transmission trade-offs," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, VR/AR Network@SIGCOMM, Los Angeles, CA, USA, August 25, 2017*.
[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[40] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, 2006.
[41] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys), Taipei, Taiwan, June 20-23, 2017*, pp. 193–198.
[42] W. Dong, Z. Ge, and S. Lee, "3g meets the internet: Understanding the performance of hierarchical routing in 3g networks," in *Proceedings of the 23rd International Teletraffic Congress (ITC), San Francisco, CA, USA, September 6-9, 2011*.

Yili Jin (S'22) received his B.E. degree from Sun Yat-sen University, China, in 2021. He is currently pursuing an M.Phil. degree at The Chinese University of Hong Kong, Shenzhen, China. His research interests include Multimedia Streaming, VR Video, and Network Verification.

Junhua Liu is currently pursuing an B.S. degree at The Chinese University of Hong Kong, Shenzhen, China. His research interests include Multimedia Streaming, VR Video, and Computer Vision.

Fangxin Wang (S'15-M'20) is an assistant professor at The Chinese University of Hong Kong, Shenzhen (CUHKSZ). He received the Ph.D., M.Eng., and B.Eng. degree all in Computer Science and Technology from Simon Fraser University, Tsinghua University, and Beijing University of Posts and Telecommunications, respectively. Before joining CUHKSZ, he was a postdoctoral fellow at the University of British Columbia. Dr. Wang's research interests include Multimedia Systems and Applications, Cloud and Edge Computing, Deep Learning and Big Data Analytics, Distributed Networking and System. He lead the intelligent networking and multimedia lab at CUHKSZ. He has published more than 30 papers at top journal and conference papers, including INFOCOM, Multimedia, ToN, TMC, IOTJ, etc. He served as the publication chair of IEEE/ACM IWQoS, TPC member of IEEE ICC, and reviewer of many top conference and journals, including INFOCOM, Multimedia, ToN, TMC, IOTJ.



Shuguang Cui (S'99-M'05-SM'12-F'14) received his Ph.D in Electrical Engineering from Stanford University, California, USA, in 2005. Afterwards, he has been working as assistant, associate, full, Chair Professor in Electrical and Computer Engineering at the Univ. of Arizona, Texas A&M University, UC Davis, and CUHK at Shenzhen, respectively. He has also served as the Executive Dean for the School of Science and Engineering at CUHK, Shenzhen, and the Executive Vice Director at Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large data set analysis, IoT system design, energy harvesting based communication system design, and cognitive network optimization. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds Most Influential Scientific Minds by ScienceWatch in 2014. He was the recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and TPC co-chairs for many IEEE conferences. He has also been serving as the area editor for IEEE Signal Processing Magazine, and associate editors for IEEE Transactions on Big Data, IEEE Transactions on Signal Processing, IEEE JSAC Series on Green Communications and Net working, and IEEE Transactions on Wireless Communications. He has been the elected member for IEEE Signal Processing Society SPCOM Technical Committee (2009-2014) and the elected Chair for IEEE ComSoc Wireless Technical Committee (2017-2018). He is a member of the Steering Committee for IEEE Transactions on Big Data and the Chair of the Steering Committee for IEEE Transactions on Cognitive Communications and Networking. He was also a member of the IEEE ComSoc Emerging Technology Committee. He was elected as an IEEE Fellow in 2013, an IEEE ComSoc Distinguished Lecturer in 2014, and IEEE VT Society Distinguished Lecturer in 2019. He has won the IEEE ICC best paper award, ICIP best paper finalist, and the IEEE Globecom best paper award all in 2020.