Communications, Caching, and Computing for Mobile Virtual Reality: Modeling and Tradeoff

Yaping Sun, Zhiyong Chen, Meixia Tao, *Fellow, IEEE*, and Hui Liu, *Fellow, IEEE*

***Abstract*— Virtual reality (VR) over wireless is emerging as an important use case of 5G networks. Fully-immersive VR experience requires the wireless delivery of huge data at ultra-low latency, thus leading to ultra-high transmission rate requirement for wireless communications. This challenge can be largely addressed by the recent network architecture known as mobile edge computing (MEC) network, which enables caching and computing capabilities at the edge of wireless networks. This paper presents a novel MEC-based mobile VR delivery framework that is able to cache parts of the field of views (FOVs) in advance and compute certain post-processing procedures on demand at the mobile VR device. To minimize the average required transmission rate, we formulate the joint caching and computing optimization problem to determine which FOVs to cache, whether to cache them in 2D or 3D as well as which FOVs to compute at the mobile device under cache size, average power consumption as well as latency constraints. When FOVs are homogeneous, we obtain a closed-form expression for the optimal joint policy which reveals interesting communications-caching-computing tradeoffs. When FOVs are heterogeneous, we obtain a local optima of the problem by transforming it into a linearly constrained indefinite quadratic problem and then applying concave convex procedure. Numerical results demonstrate the proposed mobile VR delivery framework can significantly reduce communication bandwidth while meeting low latency requirement.**

***Index Terms*— Virtual reality, mobile edge computing, wireless caching, low latency, transmission rate.**

I. Introduction

A. Motivation

VIRTUAL reality (VR) over wireless, namely mobile VR delivery, is emerging as an important use case of 5G and beyond networks, due to its ability to generate an immersive experience at the full fidelity of human perception [1]–[3]. A recent market report forecasts that the data consumption from mobile VR devices (smartphone-based or standalone) will grow by over 650% in the next 4 years (2017-2021) [4]. Immersive VR experience requires the delivery of massive amount of data (in the order of *Gigabyte*) at ultra-low latency (less than *20 ms*), thus demanding ultra-high transmission rate and leading to the wireless bandwidth bottleneck problem [3].

The recent network architecture concept known as mobile edge computing (MEC) network is envisioned as one of the key enablers for mobile VR delivery via enabling caching and computing capabilities at the edge of wireless networks (e.g., at the celluar base stations or the mobile devices) [5]. Firstly, it is observed that field of views (FOVs) for different users follow similar pattern when the users watch the same 360° video and a probabilistic model for the popularity distribution of FOVs can be learnt [6], [16]–[19]. Based on the popularity distribution of FOVs, the caching capabilities at both the MEC server and the mobile VR device can be leveraged to proactively store some FOVs for future requests. Secondly, the computation complexity of the post processing procedures for VR video production is relatively low, while the computing capability of the mobile VR device is increasing. Therefore, the computing resource of the mobile VR device can be utilized to operate the post-processing procedure to reduce the response time [8]. Thus, in this paper, we aim to investigate the mobile VR delivery using MEC network architecture and find out how to make the best use of the caching and computing resources of the mobile VR device to minimize the bandwidth requirement for mobile VR delivery while satisfying the stringent latency constraint.

B. Our Contributions

To illustrate the problem at hand, we first analyze a typical 360° VR video production framework [10], as shown in Fig. 1: i) *Stitching*, which obtains a spherical video by stitching the videos captured by a multi-camera array; ii) *Equirectangular projection*, which obtains 2-dimensional (2D) video by unfolding the obtained spherical video; iii) *Extraction*, which extracts the 2D FOV of the viewpoint captured by the *tracker* at the mobile VR device from the 2D video; iv) *Projection*, which projects 2D FOV into 3D FOV; v) *Rendering*, which renders the obtained 3D FOV onto the display of the mobile VR device.

In this paper, we propose the following realization method of the aforementioned video production framework within the

Manuscript received December 28, 2018; revised April 14, 2019 and May 24, 2019; accepted May 27, 2019. Date of publication June 3, 2019; date of current version November 19, 2019. This work is supported by the National Natural Science Foundation of China under grant 61671291 and 61571299, and the Science and Technology Commission of Shanghai Municipality under grant 18DZ2270700. This article was presented in part at the IEEE ICC 2018 [1]. The associate editor coordinating the review of this article and approving it for publication was T. He. *(Corresponding author: Zhiyong Chen.)*

Y. Sun and M. Tao are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yapingsun@sjtu.edu.cn; mxtao@sjtu.edu.cn).

Z. Chen is with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai 200240, China (e-mail: zhiyongchen@sjtu.edu.cn).

H. Liu was with Shanghai Jiao Tong University, Shanghai 200240, China. He is now with Silkwave Holdings, Hong Kong, and also with the University of Washington, Seattle, WA 98195 USA (e-mail: huiliu@sjtu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2019.2920594

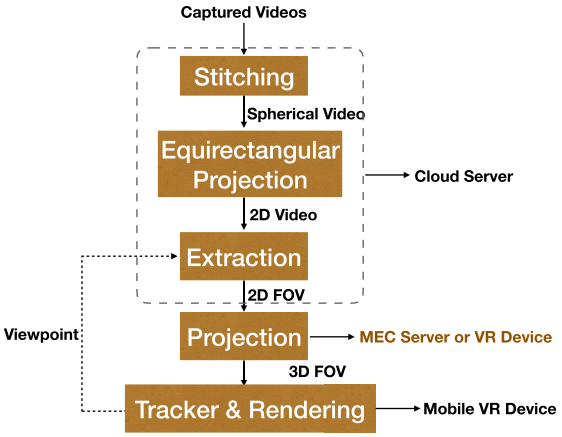


Fig. 1. A typical framework of 360° VR video production [10].

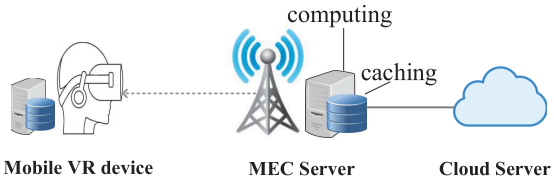


Fig. 2. MEC-based VR delivery model.

MEC network as illustrated in Fig. 2, which consists of one MEC server (e.g., base station) and one mobile VR device, both equipped with certain caching and computing capabilities. First, without doubt, the tracker and rendering components must be computed at the mobile VR device. Secondly, we consider that the first three pre-processing procedures including stitching, equirectangular projection and extraction components are computed offline at the cloud server. Since such three components require the entire 360° video as inputs, realizing them at the cloud server can release both MEC server and mobile VR device from heavy computation process as well as alleviate the traffic burden within the wireless network. Then, 2D FOVs of all the viewpoints extracted at the cloud server are assumed to be cached at the MEC server in advance, thereby reducing the traffic burden on the backhaul link and also the response latency. Moreover, the projection component is computed offline at the MEC server, and then the obtained 3D FOVs of all the viewpoints are assumed to be proactively cached at the MEC server.

A key observation is that the projection component can be offloaded from the MEC server to the mobile VR device due to its low computational complexity [11] and the increasing computing capability of the mobile VR device [5]. Specifically, compared with downloading the requested 3D FOV from the MEC server, named as MEC downloading, downloading 2D FOV from the MEC server and then computing the projection at the mobile VR device can reduce at least half of the traffic load on the wireless link. This is due to the fact that in order to create a stereoscopic vision, the projection component has to be computed twice, one of which is exposed to the viewer's left eye and the other to the viewer's right eye, and hence the data size of 3D FOV is at least twice larger than that of the 2D FOV [11], [12]. However, computing at the mobile VR device incurs additional computation latency. Thus, *the computing policy*, i.e., whether to compute the projection at the mobile VR device or not, requires careful design. In addition, caching capability of the mobile VR device can be utilized to store 2D FOVs or 3D FOVs of some viewpoints for future requests. Specifically, compared with caching a 2D FOV, caching 3D FOV can help reduce both latency and power consumption, since the 3D FOV request can be directly served and without the need of transmission or computing. However, 3D FOV caching consumes at least twice larger caching resource at the mobile VR device than 2D FOV caching. Thus, *the caching policy*, i.e., caching 2D FOVs or 3D FOVs at the mobile VR device, also requires careful design.

Main contributions of this paper are summarized as follows.

- *A novel MEC-based framework for mobile VR delivery.* We propose a realization method for mobile VR delivery, as mentioned above. This method allows the pre-processing procedures computed at the cloud server and post-processing procedure, i.e., projection component, computed at the mobile VR device, thereby significantly reducing the transmission data within the wireless network as well as required latency.
- *Optimal joint caching and computing policy.* Based on the proposed realization method, when FOVs are homogeneous, we formulate joint caching and computing decision optimization problem to minimize the average transmission rate, under the latency, cache size and average power consumption constraints. By analyzing the optimal properties and solving several linear programming problems, a closed-form expression for the optimal joint policy is obtained and provides useful guidelines for network designers on how to make the best use of caching and computing capabilities of the mobile VR device.
- *Communications-caching-computing (3C) tradeoff.* Based on the optimal joint policy, we derive the minimum required transmission rate and theoretically illustrate the 3C tradeoff. Analytical results show that compared with MEC downloading, the transmission rate gain under the optimal joint policy comes from the following three aspects: local 3D caching, local computing with local 2D caching and local computing without local caching. We theoretically reveal that caching resource at the mobile VR device is exploited more efficiently when utilizing the local computing as well, i.e., via local computing with local 2D caching. On the other hand, computing resource at the mobile VR device is exploited more efficiently when utilizing the local caching as well, i.e., via local computing with local 2D caching. In addition, we find that such three gains are exploited opportunistically in different computing frequency regimes. For example, when the computation frequency at the mobile VR device is relatively small, there is no local computing gain without local caching, and transmission rate gain comes from local 3D caching and local computing with local 2D caching. On the other hand, when the computation frequency is large enough, the gain comes from local 3D caching and local computing with/without caching simultaneously. The power efficiency of the

mobile VR device is also shown to play an important role in the transmission rate via determining the local computing gain directly. More details can be seen in Section IV.

- *Heterogeneous scenario optimization.* We extend the joint caching and computing optimization problem to the scenario where FOVs are heterogeneous. In particular, we first show the NP-hardness of the joint policy optimization problem, and then obtain a local optima of the problem via transforming it into an equivalent linearly constrained indefinite quadratic problem (IQP) and using concave convex procedure (CCCP) [13]. The proposed CCCP is shown to require low-computation complexity. Numerical results demonstrate the proposed CCCP can significantly reduce communication bandwidth while meeting low latency requirement (e.g., 63.6% gain over MEC downloading at normalized cache size 30%).

C. Related Works

Researchers in both academia and industry have made great efforts in order to achieve mobile VR delivery. First of all, at any given time, since each user only watches a portion of the 360° VR video, the requested FOV is chosen to be transmitted instead of the entire panoramic video, thereby saving bandwidth significantly. Then, by knowing each user's FOV, multi-view and tile-based video streaming have been investigated in [14], [15]. To further improve the quality of experience, motion-prediction-based transmission is also being studied based on dataset collected from real users [16]–[19]. However, [11], [14]–[19] mainly focus on the VR video-level design, and have not investigated the opportunities for mobile VR delivery potentially obtained via efficiently using the MEC network architecture.

The opportunities for mobile VR delivery that can be potentially obtained via efficiently utilizing resources at MEC network, i.e., 3C, have been studied in [5], [10], [22]–[28]. Specifically, [5], [22], [23] envision joint computing and caching as the key enablers for mobile VR delivery and illustrate the potential gain via simulation results. Reference [10] provides an explicit VR framework, based on which the insights on how to deliver 360° video in MEC network are illustrated. However, [5], [10], [22], [23] do not establish explicit theoretical formulation or propose any efficient algorithms. On the other hand, [24] proposes a collaborative cache allocation and computation offloading policy, where the MEC servers collaborate for executing computation tasks and data caching. Reference [25] extends the results in [24] to a big data MEC network. Reference [26] proposes hybrid control algorithms in smart base stations along with devised communication, caching, and computing techniques based on game theory. However, the joint caching and computing designs developed in [24]–[26] do not exploit specific nature of VR delivery and look deeper into the VR delivery framework. Thus, the performances are limited. Reference [27] formulates an optimization framework for VR video delivery in a cache-enabled cooperative multi-cell network and explores the fundamental 3C tradeoffs for VR/AR applications. Reference [28] proposes joint policy based on millimeter wave communication for interactive VR game applications.

It is worthy to note that all the works in [24]–[28] try to utilize the caching and computing resources at the MEC servers to alleviate the computation burdens at the mobile devices. However, as mentioned above, for the mobile VR delivery, computing at the MEC server may incur more transmission data since the computation results are generally larger than the inputs. In addition, all the above mentioned works in [5], [10], [22]–[28] have not investigated a specific realization method of the VR video production framework within MEC network. Thus, in this paper, we propose the aforementioned realization method and focus on utilizing the caching and computing capabilities at the mobile VR device to alleviate the communication burden on the wireless link. [29] exploits the caching and computing capabilities at the mobile VR device to minimize the traffic load over wireless link. However, [29] does not capture the specific nature of VR delivery and thus the performance is limited. Compared with our previous work [1], we extend the caching model into more general one via considering not only 2D FOV caching but also 3D FOV caching, leading to fundamental difference of problem formulation, proposed algorithm as well as analytical and numerical results. The analytical and numerical results have shown that taking into account both 2D and 3D FOV caching helps further improve the system performance.

D. Outline

An outline of the remainder of the paper is as follows. Section II describes the system model for the MEC-based mobile VR delivery system under consideration. Section III formulates the joint policy optimization problem for the homogeneous scenario. Section IV obtains the optimal policy and 3C tradeoffs. Section V formulates the optimization problem for the heterogeneous scenario and obtains the local optima via CCCP. Section VI concludes the paper.

II. System Model

As illustrated in Fig. 2, we consider a novel MEC-based mobile VR delivery system consisting of one MEC server and one mobile VR device, both with certain caching and computing capabilities. In this paper, we focus on a single 360° VR video streaming. As mentioned above, instead of transmitting the whole 360° video, the MEC server only delivers the requested FOV at each time. Key notations in this paper are summarized in Table I.

A. VR Task Model

Denote with $\mathcal{N} \triangleq \{1, \cdots, N\}$ the space of viewpoints, which are obtained via the extraction of the 2D video, as illustrated in Fig. 1. The projection from 2D FOV into 3D FOV of each viewpoint $i \in \mathcal{N}$ is characterized by a $3-$tuple (D^I, D^O, w), where D^I and D^O are the data sizes (*in bit*) of the 2D FOV and 3D FOV, respectively, and w is the number of computation cycles required to process one bit input (*in cycle/bit*). Denote with $\alpha \triangleq \frac{D^O}{D^I}$ the ratio of the size of 3D

TABLE I
KEY NOTATIONS

Notation	Meaning
$\mathcal{N}, N, i$	set of viewpoints, number of viewpoints, viewpoint index
k	power efficiency of CPU at the mobile VR device
D^I, w, D^O, τ	data size of 2D FOV, computation load, data size of 3D FOV, maximum tolerable service latency
$C, \bar{P}, f_V$	cache size, average available power, computation frequency at the mobile VR device
$\frac{N\bar{P}\tau}{kf_V^2 D^I w}$	computing size: the maximum number of FOVs that can be computed at the mobile VR device
R_S	the least required transmission rate when the projection is computed at the MEC server
R_V	the least required transmission rate when the projection is computed at the mobile device without caching
F	the computation frequency obtained when $R_S = R_V$
$c_i^I \in \{0, 1\}$	$c_i^I = 1$ means that the 2D FOV of viewpoint i is stored at the mobile VR device and not otherwise
$c_i^O \in \{0, 1\}$	$c_i^O = 1$ means that the 3D FOV of viewpoint i is stored at the mobile VR device and not otherwise
$d_i \in \{0, 1\}$	$d_i = 1$ means that projection is computed at the mobile VR device and not otherwise

FOV to that of 2D FOV. Typically, $\alpha \geq 2$ in order to create a stereoscopic vision [11], [12].

B. Request Model

Similar to [6], [16]–[19], [27], the request stream at the mobile VR device conforms to the independent reference model (IRM) based on the following assumptions: i) the viewpoints that the mobile VR device requests are fixed to the set $\mathcal{N}$; ii) the probability of the request for viewpoint $i \in \mathcal{N}$ at the mobile VR device at each time, denoted as P_i, is constant and independent of all the past requests, capturing how often viewpoint i is selected by the VR user as he or she navigates the scene and satisfying $\sum_{i=1}^{N} P_i = 1$. We consider uniform distribution, i.e., $P_i = \frac{1}{N}$ for each $i \in \mathcal{N}$.[1] In addition, in order to avoid dizziness and nausea, each request at the mobile VR device must be satisfied within the deadline of τ (*in second*).

C. Caching and Computing Model

First, consider the cache placement at the mobile VR device. We assume that the mobile VR device is equipped with a cache size CD^I (*in bit*), where C is an integer, and is able to store both 2D and 3D FOVs of some viewpoints. Denote with $c_i^I \in \{0, 1\}$ the caching decision for 2D FOV of viewpoint i, where $c_i^I = 1$ means that the 2D FOV of viewpoint i is cached at the mobile VR device and $c_i^I = 0$ otherwise. Denote with $c_i^O \in \{0, 1\}$ the caching decision for 3D FOV of viewpoint i, where $c_i^O = 1$ means that the 3D FOV of viewpoint i is cached at the mobile VR device and $c_i^O = 0$ otherwise. Under the cache size constraint of the mobile VR device, we have

$$\sum_{i=1}^{N} D^I c_i^I + \alpha D^I c_i^O \leq CD^I. \tag{1}$$

For the cache placement at the MEC server, we assume that both 2D and 3D FOVs of all the viewpoints are cached at the MEC server. This is reasonable due to the fact that the storage size at the MEC server is much larger than that of the mobile VR device.

[1]The scenario with nonuniform data size and popularity distribution is considered in Section V.

Next, consider the computing decision for the projection component at the mobile VR device. The mobile VR device is assumed to run at a given CPU-cycle frequency, denoted as f_V (*in cycle/s*). The energy consumed for computing one cycle with frequency f_V at the mobile VR device is kf_V^2, where k is the effective switched capacitance related to the chip architecture and can indicate the power efficiency of CPU at the mobile VR device [31], [32]. The mobile VR device is assumed to be equipped with fixed and finite energy capacity. In order to make sure that the duration for the mobile user to experience VR video is no shorter than certain time on average, requests for other applications can also get served and the lifetime of the mobile device battery can last longer, a long-term time averaged power consumption constraint, denoted as $\bar{P}$ (*in W*), is considered.[2] Denote with $d_i \in \{0, 1\}$ the computing decision for viewpoint i, where $d_i = 1$ indicates that the projection from 2D FOV to 3D FOV is executed at the mobile VR device upon viewpoint request and $d_i = 0$ otherwise. Under the average power consumption constraint of the mobile VR device, we have

$$\frac{kf_V^2 D^I w}{N\tau} \sum_{i=1}^{N} d_i \leq \bar{P}. \tag{2}$$

From (2), note that $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ corresponds to the maximum number of projections that will be computed at the mobile VR device if requested, named as *computing size* of the mobile VR device. $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ is assumed to be an integer throughout this paper for simplicity of analysis.

Last, denote with $(\mathbf{c}^O, \mathbf{c}^I, \mathbf{d})$ the joint caching and computing decision at the mobile VR device, where $\mathbf{c}^O \triangleq (c_i^O)_{i\in\mathcal{N}}$ denotes the caching decision vector for 3D FOVs of all the viewpoints and $\mathbf{c}^I \triangleq (c_i^I)_{i\in\mathcal{N}}$ denotes the caching decision vector for 2D FOVs of all the viewpoints, satisfying the cache size constraint in (1), and $\mathbf{d} \triangleq (d_i)_{i\in\mathcal{N}}$ denotes the computing decision vector, satisfying the local average power consumption constraint in (2).

[2]Note that the finite experience time assumption does not contradict the long-term average. Take experience time 5 minutes as an example. The number of time slots is $\frac{5*60}{20*10^{-3}} = 15000$, which can be deemed as infinite horizon.

D. Service Mechanism and Transmission Rate Requirement

Based on the joint caching and computing decision $(\mathbf{c}^O, \mathbf{c}^I, \mathbf{d})$, we can see that request for viewpoint $i \in \mathcal{N}$ can be served via the following four routes, each of which yields a unique minimum transmission rate requirement for satisfying the latency constraint, denoted as R_i (*in bit/s*).

- **Local 3D caching.** If $c_i^O = 1$, the 3D FOV of viewpoint i can be obtained from the local cache and without the need of the transmission or computing. In this way, the required latency is negligible and the minimum required transmission rate is $R_i = 0$.
- **Local computing with local 2D caching.** If $c_i^O = 0$, $d_i = 1$ and $c_i^I = 1$, the mobile VR device obtains the 2D FOV of viewpoint i from the local cache and without the need of transmission, and then projects it to 3D FOV using its local CPU processor. Thus, the overall consumed latency is $\frac{D^I w}{f_V}$ (*in second*) and the minimum required transmission rate is $R_i = 0$. In this paper, we assume that $\frac{D^I w}{f_V} < \tau$ for feasibility, i.e., computing the projection at the mobile VR device can be completed within the deadline.
- **Local computing without local caching.** If $c_i^O = 0$, $d_i = 1$ and $c_i^I = 0$, the mobile VR device downloads the 2D FOV of viewpoint i from the MEC server and then projects it to 3D FOV using its local CPU processor. Thus, the overall consumed latency is $\frac{D^I}{R_i} + \frac{D^I w}{f_V}$ (*in second*), where $\frac{D^I}{R_i}$ corresponds to the 2D FOV transmission latency over the wireless link and $\frac{D^I w}{f_V}$ corresponds to the computation latency at the mobile VR device. Under the latency constraint $\frac{D^I}{R_i} + \frac{D^I w}{f_V} \le \tau$, the minimum required transmission rate is $R_i = R_V \triangleq \frac{D^I}{\tau - \frac{D^I w}{f_V}}$.
- **MEC downloading.** If $c_i^O = 0$ and $d_i = 0$, the mobile VR device downloads the 3D FOV of viewpoint i from the MEC server. Then, the overall consumed latency can be represented as $\frac{D^O}{R_i}$ (*in second*). Under the latency constraint $\frac{D^O}{R_i} \le \tau$, the minimum required transmission rate is $R_i = R_S \triangleq \frac{D^O}{\tau}$.

By combining all the above cases, for any given joint caching and computing decision $(\mathbf{c}^O, \mathbf{c}^I, \mathbf{d})$, the minimum long-term time averaged required transmission rate to deliver the requested 3D FOV under the latency constraint, denoted as $\bar{R}$ (*in bit/s*), is given by

$$\bar{R} \triangleq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} R_t \overset{(a)}{=} \frac{1}{N} \sum_{i=1}^{N} \left(R_i^S (1-d_i) + R_i^V d_i \left(1 - c_i^I\right)\right)\left(1 - c_i^O\right), \quad (3)$$

where (a) holds based on the assumption that the request stream at the mobile VR device conforms to IRM, specified in Section II-B. Note that minimizing $\bar{R}$ is equivalent to minimizing the average required bandwidth for a given spectral efficiency. This is important for network operators because when a fixed band of spectrum is available to the MEC server (large enough to satisfy the FOV request under latency constraint), the smaller the average bandwidth is consumed, the more portion of the available spectrum is more often underutilized. Then, the underutilized bandwidth can be allocated to serve some other mobile user requests via spectrum sensing. Therefore, minimizing the average rate improves the bandwidth utilization and network capacity.

TABLE II

TRANSMISSION RATES VS. LOCAL CACHING AND COMPUTING COSTS

Joint Decision	Rate	Caching	Computing
Local 3D caching $c_i^O = 1, c_i^I = 0, d_i = 0$	0	αD^I	0
Local computing with local 2D caching $c_i^O = 0, c_i^I = 1, d_i = 1$	0	D^I	$\frac{kD^I w f_V^2}{N\tau}$
Local computing without local caching $c_i^O = 0, c_i^I = 0, d_i = 1$	$\frac{R_V}{N}$	0	$\frac{kD^I w f_V^2}{N\tau}$
MEC downloading $c_i^O = 0, c_i^I = 0, d_i = 0$	$\frac{R_S}{N}$	0	0

Remark 1: As illustrated in Table II, for each viewpoint $i \in \mathcal{N}$, we have

- *compared with local 3D caching, local computing with local 2D caching achieves the same rate gain and saves at least half of the cache size consumed by local 3D caching, but incurs additional power consumption;*
- *compared with local computing with local 2D caching, local computing without local caching saves cache cost, but incurs larger transmission rate requirement;*
- *compared with local computing without local caching, MEC downloading saves local caching and computing cost, but relationship between its incurred transmission rate, i.e., R_S, and that incurred by local computing without local caching, i.e., R_V, depends on the local computing frequency f_V.*

Thus, joint caching and computing design requires careful thinking.

III. PROBLEM FORMULATION AND OPTIMAL PROPERTY ANALYSIS

In this section, we formulate the joint caching and computing optimization problem to minimize the average required transmission rate and analyze the optimal properties, based on which we obtain an equivalent problem.

A. Problem Formulation

Problem 1 (Joint Caching and Computing Optimization):

$$\min_{\mathbf{c}^O, \mathbf{c}^I, \mathbf{d}} \frac{1}{N} \sum_{i=1}^{N} \left(R_V d_i (1 - c_i^I) + R_S (1 - d_i)\right)\left(1 - c_i^O\right)$$

$$s.t. \sum_{i=1}^{N} c_i^I + \alpha c_i^O \le C, \quad (4)$$

$$\sum_{i=1}^{N} d_i \le \frac{N\bar{P}\tau}{k f_V^2 D^I w}, \quad (5)$$

$$c_i^O \in \{0,1\}, \quad c_i^I \in \{0,1\}, \; d_i \in \{0,1\}, \; i \in \mathcal{N},$$

where (4) and (5) correspond to the cache size constraint in (1) and average power consumption constraint in (2), respectively.

The optimization variables are caching decision at the mobile VR device, i.e., $\left(\mathbf{c}^O = (c_i^O)_{i\in\mathcal{N}}, \mathbf{c}^I = (c_i^I)_{i\in\mathcal{N}}\right)$, and local computing decision, i.e., $\mathbf{d} = (d_i)_{i\in\mathcal{N}}$. The objective function is average transmission rate requirement, i.e., $\bar{R}$ in (3). Denote with $(\mathbf{c}^{O^*}, \mathbf{c}^{I^*}, \mathbf{d}^*)$ the optimal joint caching and computing decision, and R^* the optimal objective value of Problem 1.

B. Optimal Properties and Equivalent Formulation

In this subsection, we analyze the optimal properties of the joint caching and computing policy, based on which we obtain an equivalent optimization. Denote with $c^O \triangleq \sum_{i=1}^N c_i^O$, $c^I \triangleq \sum_{i=1}^N c_i^I$ and $d \triangleq \sum_{i=1}^N d_i$ the number of locally cached 3D FOVs, that of locally cached 2D FOVs and that of locally computed projections, respectively. From (4) and (5), we have $c^O \in \left\{0, 1, \cdots, \frac{C}{\alpha}\right\}$, $c^I \in \left\{0, 1, \cdots, C - \alpha c^O\right\}$ and $d \in \left\{0, 1, \cdots, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}$, respectively. Considering that the projection tuple (D^I, D^O, w, P_i, τ) of each viewpoint $i \in \mathcal{N}$ is the same, it does not matter which 3D FOV is cached. Therefore, for any given c^O, we can let

$$c_i^O = \begin{cases} 1 & i = 1, \cdots, c^O, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

without loss of optimality.

We first obtain the optimality property between local 2D and 3D FOV caching.

Property 1: For any $i \in \mathcal{N}$ such that $c_i^O = 1$, we have $c_i^I = 0$ without loss of optimality.

This property indicates that if 3D FOV of viewpoint i is already cached at the mobile VR device, there is no need to cache the 2D FOV, since the request for viewpoint i can be directly served from local cache.

Property 2: For any given c^O, we have $c^I = C - \alpha c^O$ without loss of optimality.

Property 2 can be obtained by observing that the equality holds in the cache size constraint (4) for minimizing the required transmission rate. Based on Property 1 and Property 2, when $\mathbf{c}^O$ is given by (6), we can let

$$c_i^I = \begin{cases} 0 & i = 1, \cdots, c^O, \\ 1 & i = c^O + 1, \cdots, c^O + c^I, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where $c^I = C - \alpha c^O$.

We next analyze the optimality between local caching and local computing as follows.

Property 3: For any viewpoint $i \in \mathcal{N}$, we have $c_i^O + d_i \leq 1$ and $c_i^I \leq d_i$ without loss of optimality.

Property 3 can be obtained by contradiction. First, suppose that $c_i^O + d_i > 1$. Then, when $c_i^O = 1$, we have $d_i = 1$. By setting d_i from 1 into 0, $R_i = 0$ does not change and power consumption cost is saved. Since d_j, $\forall\ j \in \mathcal{N}$ has to satisfy the average power consumption constraint $\sum_{i=1}^N d_i \leq \frac{N\bar{P}\tau}{kf_V^2 D^I w}$, the feasible domain for d_j, $\forall\ j \in \mathcal{N} \setminus i$ becomes larger. Therefore, by setting d_i from 1 into 0, the achieved average rate $\bar{R}$ will not increase, i.e., $c_i^O + d_i \leq 1$ is without loss of optimality. Secondly, suppose that $c_i^I > d_i$. Then, when $d_i = 0$, we have $c_i^I = 1$. By setting c_i^I from 1 into 0, based on (3), R_i does not change and caching cost is saved. Since c_j^I, $\forall\ j \in \mathcal{N}$ has to satisfy the cache size constraint $\sum_{i=1}^N c_i^I + \alpha c_i^O \leq C$, the feasible domain for c_j^I, $\forall\ j \in \mathcal{N} \setminus i$ becomes larger. Thus, by setting c_i^I from 1 into 0, the achieved average rate $\bar{R}$ will not increase, i.e., $c_i^I \leq d_i$ is without loss of optimality.

Property 3 indicates that if 3D FOV of viewpoint i is already cached at the mobile VR device, there is no gain from local computing, since the request for viewpoint i can be directly served from the local cache. Similarly, if 2D FOV is already cached at the mobile VR device, it would be a waste of caching resource if the locally cached 2D FOV is not utilized to compute the projection component at the mobile VR device.

Based on Property 3, when $\mathbf{c}^O$ and $\mathbf{c}^I$ are given by (6) and (7), for any given d, we can let

$$d_i = \begin{cases} 0 & i = 1, \cdots, c^O, \\ 1 & i = c^O + 1, \cdots, c^O + d, \\ 0 & \text{otherwise}. \end{cases} \tag{8}$$

Finally, for ease of structural property analysis, by rewriting $\mathbf{c}^O$, $\mathbf{c}^I$ and $\mathbf{d}$ as (6), (7) and (8), Problem 1 is equivalent to Problem 2.

Problem 2 (Equivalent Joint Policy Optimization):

$$\min_{c^O, c^I, d} R_S - \frac{R_S}{N} c^O - \frac{R_S}{N} \min\left\{c^I, d\right\} - \frac{R_S - R_V}{N}\left(d - \min\left\{c^I, d\right\}\right)$$

$$s.t.\ c^I \in \{0, 1, \cdots, C\}, \tag{9}$$

$$c^O = \frac{C - c^I}{\alpha}, \tag{10}$$

$$d \in \left\{0, 1, \cdots, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}. \tag{11}$$

Denote with $\left(c^{O^*}, c^{I^*}, d^*\right)$ the optimal solution to Problem 2. Based on (6), (7) and (8), we can obtain the corresponding optimal joint policy, i.e., $\left(\mathbf{c}^{O^*}, \mathbf{c}^{I^*}, \mathbf{d}^*\right)$.

From the objective function of Problem 2, the first term R_S corresponds to the average transmission rate required via MEC downloading. The second term $\frac{R_S}{N} c^O$ corresponds to the local 3D caching gain, which increases with the number of locally cached 3D FOVs, i.e., c^O. The third term $\frac{R_S}{N} \min\left\{c^I, d\right\}$ corresponds to the local computing gain with local 2D caching, which increases with the minimum of the number of locally cached 2D FOVs, i.e., c^I, and that of locally computed projections, i.e., d. The last term $\frac{R_S - R_V}{N}\left(d - \min\left\{c^I, d\right\}\right)$ corresponds to the local computing gain without local caching, which depends on the difference between R_S and R_V. When $R_S = \frac{D^O}{\tau}$ is equal to $R_V = \frac{D^I}{\tau - \frac{D^I w}{f_V}}$, we obtain $f_V = F \triangleq \frac{D^O w}{(\alpha - 1)\tau}$. Note that if $f_V < F$, $R_S < R_V$ indicating there is no gain from local computing without local caching. Thus, we name $f_V < F$ as *local computing limited region*. Otherwise, there is gain from local computing without local caching and we name $f_V \geq F$ as *MEC downloading limited region*. In summary, the total number of viewpoint requests that can

be served locally is $c^O + \min\{c^I, d\} + (d - \min\{c^I, d\}) = c^O + d$. For the interest of joint caching and computing design, we assume that $\frac{C}{\alpha} + \frac{N\bar{P}\tau}{kf_V^2 D^I w} \leq N$.

IV. Optimal Policy and Tradeoff Analysis

In this section, we obtain the optimal joint caching and computing policy and the minimum transmission rate, yielding the fundamental relationship between communications, caching and computing, defined as **3C tradeoff**, in the local computing limited region, i.e., $f_V < F$, and MEC downloading limited region, i.e., $f_V \geq F$, respectively.

A. Local Computing Limited Region

Theorem 1 (Optimal Joint Policy and 3C Tradeoff When $f_V < F$): The optimal joint policy (c^{O^*}, c^{I^*}, d^*) is given as

$$c^{O^*} = \frac{C - c^{I^*}}{\alpha}, \tag{12}$$

$$c^{I^*} = \min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}, \tag{13}$$

$$d^* = \min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}, \tag{14}$$

and the minimum transmission rate R^* is given as

$$R^* = R_S - \frac{R_S}{N}\left(\frac{C}{\alpha} + \left(1 - \frac{1}{\alpha}\right)\min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}\right). \tag{15}$$

Proof: Proof can be seen in Appendix A. □

B. MEC Downloading Limited Region

Theorem 2 (Optimal Joint Policy and 3C Tradeoff When $F \leq f_V$): The optimal joint policy, i.e., (c^{O^*}, c^{I^*}, d^*), is given as

$$c^{O^*} = \frac{C - c^{I^*}}{\alpha}, \tag{16}$$

$$c^{I^*} = \min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}, \tag{17}$$

$$d^* = \frac{N\bar{P}\tau}{kf_V^2 D^I w}, \tag{18}$$

and the minimum transmission rate R^* is given as

$$R^* = R_S - \frac{R_S}{N}\left(\frac{C}{\alpha} + \left(1 - \frac{1}{\alpha}\right)\min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}\right) - \frac{R_S - R_V}{N}\left(\frac{N\bar{P}\tau}{kf_V^2 D^I w} - \min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}\right). \tag{19}$$

Proof: Proof can be seen in Appendix B. □

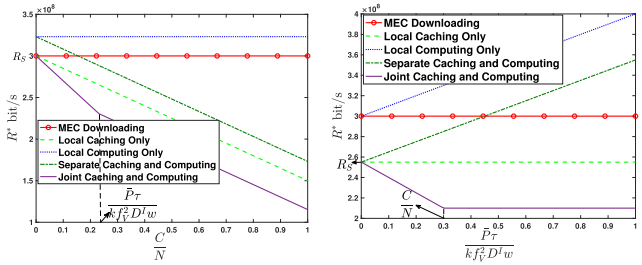
C. Numerical Results and Tradeoff Analysis

First, we would like to concrete an implementation example for the VR delivery system as below. Firstly, the 2D FOVs are extracted from the 2D video as follows. In particular, after obtaining the 2D video via equirectangular projection, each 360° 2D image is divided into 24 tiles, 6 columns and 4 rows. 24 tiles are chosen because it provides a good tradeoff between encoding efficiency and bandwidth saving [7]. Each tile is encoded using ffmpeg H.264 encoder at resolution 640×540 pixels. So resolution of complete 360° video is 3840×2160 pixels. Each tile is encoded with Constant Rate Factor (CRF) of 21 and set the maximum bitrate per tile to 3 Mbps [6]. After encoding, the 360° video is divided into segments with duration of 1 second using GPAC MP4Box tool. So each tile of a segment would be available in a separate file of size $D^I = 3$ Mbits, which corresponds to 2D FOV file mentioned in the paper. In addition, we consider the aforementioned division of each 360° 2D image at any time into 24 tiles is conducted 100 times, each with a unique combination of 24 tiles. This is in order to keep adaptive to user's motion of head, since each user during each VR experience may request different set of viewpoints. We consider a 360° video of 5 minutes. Thus, there are overall $N = 24 \times 5 \times 60 \times 100 = 7.2 \times 10^5$ FOV files. The size of all the 2D FOV files is $3\ M \times 7.2 \times 10^5 = 2160$ Gbits. Similar parameters have been selected in [6]. Secondly, in order to create the stereoscopic vision, the projection of the 2D FOV has to be computed twice, one of which is exposed to the viewer's left eye and the other to the viewer's right eye [11], [12]. Therefore, the size of each 3D FOV file is at least twice larger than that of 2D FOV, and we choose $\alpha = 2$, $D^O = 2 \times 3 = 6$ Mbits. Thirdly, $\tau = 20$ ms based on the fact that if the reaction delay is longer than 20 ms, it generally incurs diszziness and nausea [3]. Lastly, $k = 10^{-27}$ [32], $w = 10$ cycles/bit [28], [29], $f_V = 1.5 \sim 5$ GHz [33] and $\bar{P} = 5$ W [34]. Note that simulation parameters are chosen as mentioned above unless otherwise stated.

Then, based on Theorem 1, we analyze the impacts of cache size C, computing size $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$, computation frequency f_V and average power $\bar{P}$ on the average transmission rate R^* when $f_V < F$ by plotting numerical results. We compare the optimal policy named as "Joint Caching and Computing" with the following four baselines.

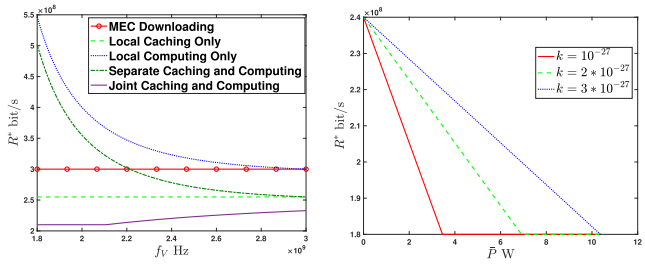
- **MEC Downloading:** requests are satisfied via downloading the 3D FOVs from the MEC server, i.e., $c_i^O = 0, c_i^I = 0, d_i = 0$ for all $i \in \mathcal{N}$.
- **Local Caching Only:** the caching resource at the mobile VR device is totally utilized for 3D caching and computing resource is not utilized, i.e., $c_i^O = 1,\ c_i^I = 0,\ d_i = 0$ for $i = 1, \cdots, \frac{C}{\alpha}$; $c_i^O = 0,\ c_i^I = 0,\ d_i = 0$, otherwise.
- **Local Computing Only:** the caching resource at the mobile VR device is not utilized and computing resource is totally utilized for computing 2D FOV downloaded from the MEC server, i.e., $c_i^O = 0,\ c_i^I = 0,\ d_i = 1$ for $i = 1, \cdots, \frac{N\bar{P}\tau}{kf_V^2 D^I w}$; $c_i^O = 0,\ c_i^I = 0,\ d_i = 0$, otherwise.
- **Separate Caching and Computing**: the caching resource at the mobile VR device is totally utilized for 3D FOV caching and computing resource at the mobile VR device is totally utilized for computing 2D FOV downloaded from the MEC server, i.e., $c_i^O = 1,\ c_i^I = 0,\ d_i = 0$ for $i = 1, \cdots, \frac{C}{\alpha}$; $c_i^O = 0,\ c_i^I = 0,\ d_i = 1$ for $i = \frac{C}{\alpha} + 1, \cdots, \frac{C}{\alpha} + \frac{N\bar{P}\tau}{kf_V^2 D^I w}$; $c_i^O = 0,\ c_i^I = 0,\ d_i = 0$, otherwise.

Fig. 3 (a) illustrates the impact of the cache size C on the optimal rate R^* when $f_V < F$. Intuitively, the average rate



(a) Cache size $\frac{C}{N}$ when $\bar{P}$ = 2 W, f_V = 2.4 GHz.

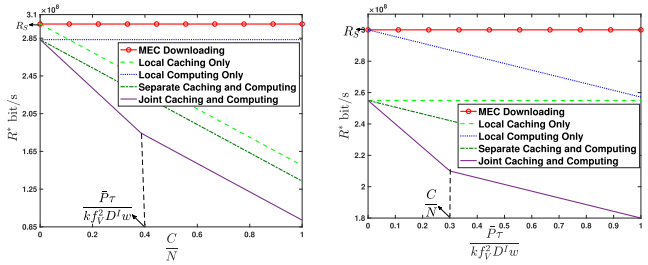
(b) Computing size $\frac{\bar{P}\tau}{kf_V^2 D^I w}$ when $\frac{C}{N}$ = 30%.



(c) Computation frequency f_V when $\bar{P}$ = 2 W and $\frac{C}{N}$ = 30%.

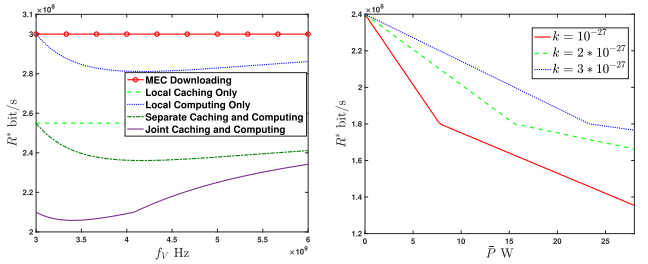
(d) Average power $\bar{P}$ when f_V = 2.4 GHz and $\frac{C}{N}$ = 30%.

Fig. 3. Cache size, computing size, computation frequency and average power when $f_V < F$.



(a) Cache size when $\frac{\bar{P}\tau}{kf_V^2 D^I w}$ = 30%.

(b) Computing size when $\frac{C}{N}$ = 30%.



(c) Computation frequency f_V when $\bar{P}$ = 7.5 W and $\frac{C}{N}$ = 30%.

(d) Average power $\bar{P}$ when f_V = 3.6 GHz and $\frac{C}{N}$ = 30%.

Fig. 4. Cache size, computing size, computation frequency and average power when $f_V \geq F$.

monotonically decreases with C. We can observe that joint caching and computing achieves good performance gains over the four baselines. Also, the gains are shown to depend on the relationship between the caching C and computing $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ sizes of the mobile VR device. In particular, when $C = 0$, the average rate achieved via local computing only is larger than that obtained via MEC downloading. This is because when $f_V < F$, $R_S < R_V$. When $0 < C \leq \frac{N\bar{P}\tau}{kf_V^2 D^I w}$, $R^* = R_S - R_S\frac{C}{N}$ and decreases with $\frac{C}{N}$ with a slope of R_S, larger than that of local caching only $\frac{R_S}{\alpha}$. This is because all the cache resource at the mobile VR device is exploited for the service route local computing with local 2D caching, and local 2D caching helps reduce the caching cost compared with local 3D caching. When $\frac{N\bar{P}\tau}{kf_V^2 D^I w} < C$, $R^* = R_S - \frac{R_S}{N}\left(\frac{C}{\alpha} + \left(1 - \frac{1}{\alpha}\right)\frac{N\bar{P}\tau}{kf_V^2 D^I w}\right)$ and decreases with $\frac{C}{N}$ with a slope of $\frac{R_S}{\alpha}$, the same as that of local caching only. This is because the caching gain also comes from local 3D caching and the 3D FOV size is α times larger than the 2D FOV size. This indicates that computing at the mobile device facilitates the utilization of caching at the mobile device when $f_V < F$.

Fig. 3 (b) illustrates the impact of the computing size $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ on the optimal rate R^* when $f_V < F$. We can see that R^* first decreases with $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ when $\frac{N\bar{P}\tau}{kf_V^2 D^I w} \leq C$ and then remains unchanged otherwise. This is because when $f_V < F$, $R_S < R_V$ and thus local computing only can not bring performance gain. This indicates that caching at the mobile device facilitates the utilization of computing at the mobile device when $f_V < F$.

Fig. 3 (c) illustrates the impact of the computation frequency f_V on the optimal rate R^* when $f_V < F$. We can see that R^* under local computing only or separate caching and computing decreases with f_V. This is because increasing f_V decreases the number of requests that can be computed locally and then more requests are satisfied via MEC downloading. Since $R_S < R_V$, the average rate decreases. On the other hand, R^* under joint caching and computing increases with f_V. This is because increasing f_V decreases the number of requests that can be satisfied via local computing with 2D caching, and then caching is more likely to be used for local 3D caching rather than 2D caching. This indicates that computing at the mobile device also facilitates the utilization of caching at the mobile device when $f_V < F$.

Fig. 3 (d) illustrates the impacts of the average power $\bar{P}$ and k on the optimal rate R^* when $f_V < F$ under joint caching and computing. We can see that R^* first decreases with $\bar{P}$ and then remains unchanged. This is because increasing $\bar{P}$ increases the number of projections that can be satisfied via local computing with 2D caching, but the effect is limited by local cache size. We also observe that the decreasing rate of R^* w.r.t. $\bar{P}$ decreases with k. This is because increasing k corresponds to decreasing the power efficiency. Therefore, improving the power efficiency of the mobile VR device can help facilitate utilizing the local computing resource and thereby reduce the transmission rate requirement.

Lastly, based on Theorem 2, we analyze the impacts of cache size C, computing size $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$, computation frequency f_V and average power $\bar{P}$ on R^* when $f_V \geq F$ via plotting numerical results.

Fig. 4 (a) illustrates the impact of C on the optimal rate R^* when $F \leq f_V$. As expected, the average rate monotonically decreases with C. We can observe that joint caching and computing still achieves good performance gains over the four baselines. Also, the gains are shown to depend on the relationship between the caching C and computing $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$

sizes of the mobile VR device. In particular, when $C = 0$, local computing only achieves performance gain over MEC downloading, i.e., $\frac{R_S - R_V}{N} \frac{N\bar{P}\tau}{kf_V^2 D^I w}$. This is because when $F \leq f_V$, $R_S \geq R_V$. When $0 < C \leq \frac{N\bar{P}\tau}{kf_V^2 D^I w}$, the decreasing slope is R_V, larger than that of local caching only $\frac{R_S}{\alpha}$. When $\frac{N\bar{P}\tau}{kf_V^2 D^I w} < C$, the decreasing slope is $\frac{R_S}{\alpha}$, the same as that of local caching only. Thus, we conclude that computing at the mobile device helps facilitate the utilization of the caching at the mobile device when $F \leq f_V$.

Fig. 4 (b) illustrates the impact of the computing size $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ on the optimal rate R^* when $F \leq f_V$. We can observe that joint caching and computing still achieves good performance gains over the four baselines. When $\frac{N\bar{P}\tau}{kf_V^2 D^I w} = 0$, local caching only, separate caching and computing or joint caching and computing achieves performance gain over MEC downloading or local computing only. When $0 < \frac{N\bar{P}\tau}{kf_V^2 D^I w} < C$, the decreasing slope of R^* is $R_S(1 - \frac{1}{\alpha})$, larger than that of local computing only, i.e., $R_S - R_V$. When $\frac{N\bar{P}\tau}{kf_V^2 D^I w} \geq C$, the decreasing slope of R^* is $R_S - R_V$, the same as that of local computing only. Thus, we conclude that caching at the mobile device helps facilitate the utilization of the computing at the mobile device when $F \leq f_V$.

Fig. 4 (c) illustrates the impact of f_V on the optimal rate R^* when $F \leq f_V$. We can see that R^* first decreases and then increases with f_V. This is mainly because when f_V is relatively small, increasing f_V alleviates the transmission rate requirement by reducing the local computation latency. On the other hand, when f_V is relatively large, increasing f_V decreases the number of projections that can be computed at the mobile VR device. From the first-order derivative of R^* w.r.t. f_V, we obtain the following remark.

Remark 2: When $C = 0$ and $F \leq f_V$, f_V^* minimizing R^* is given by

$$f_V^* = \left(1 - \frac{D^I}{4D^O}\right)F + \sqrt{\left(1 - \frac{D^I}{4D^O}\right)^2 F^2 - \frac{D^I w}{\tau}F}. \quad (20)$$

Equation (20) indicates that the optimal computation frequency is independent of the average power $\bar{P}$ and power efficiency k of the mobile device, and depends on the projection parameters (D^I, D^O, w, τ) only. Remark 2 provides a theoretical guideline for the mobile VR device designer who chooses the CPU frequency when caching is not available.

Fig. 4 (d) illustrates the impacts of $\bar{P}$ and k on the optimal rate R^*. We observe that R^* decreases with $\bar{P}$ and the decreasing slope decreases with $\bar{P}$. This is because when $\bar{P}$ is relatively small, the computing size is smaller than the caching size and thus the computing resource is jointly utilized with local 2D caching. When $\bar{P}$ is so large that the computing size is larger than the caching size, in addition to local computing with 2D caching, the remaining computing resource achieves performance gain independently. Similar to the case when $f_V < F$, we also observe that the decreasing slope of R^* w.r.t. $\bar{P}$ decreases with k, which again indicates that improving the power efficiency of the mobile VR device can help facilitate utilizing the local computing resource and thereby reduce the transmission rate requirement.

V. Problem Formulation in Heterogeneous Scenario

In this section, we consider a heterogeneous scenario, where the parameters of each viewpoint $i \in \mathcal{N}$, generalized as $(D_i^I, D_i^O, w_i, \tau_i, P_i)$, are different from each other. This scenario is more practical since there are generally some FOVs more probable to be requested than the others. Similar to the homogeneous scenario, we would like to design the optimal joint caching and computing policy, i.e., $(\mathbf{c}^{O^*}, \mathbf{c}^{I^*}, \mathbf{d}^*)$, to minimize the average transmission rate $\bar{R}$ under both the cache size and average power consumption constraints. In particular, the optimization problem is formulated as below.

Problem 3 (Joint Caching and Computing Policy Optimization in Heterogeneous Scenario):

$$\min_{\mathbf{c}^I, \mathbf{c}^O, \mathbf{d}} \sum_{i=1}^{N} P_i \left(R_i^S (1 - d_i) + R_i^V d_i \left(1 - c_i^I\right)\right)\left(1 - c_i^O\right)$$

$$s.t. \sum_{i=1}^{N} P_i \frac{kf_V^2 D_i^I w_i}{\tau} d_i \leq \bar{P}, \quad (21)$$

$$\sum_{i=1}^{N} D_i^I c_i^I + \alpha D_i^I c_i^O \leq C',$$

$$c_i^O \in \{0,1\}, \quad c_i^I \in \{0,1\}, \; d_i \in \{0,1\}, \; i \in \mathcal{N}, \quad (22)$$

where $R_i^S \triangleq \frac{D_i^O}{\tau_i}$ (*in bit/s*) and $R_i^V \triangleq \frac{D_i^I}{\tau_i - \frac{D_i^I w_i}{f_V}}$ (*in bit/s*) denote the minimally required transmission rates to satisfy the latency constraint when the projection of viewpoint $i \in \mathcal{N}$ is computed at the MEC server and at the mobile VR device, respectively. The objective function is obtained via generalizing (3). C' (*in bit*) denotes the cache size at the mobile VR device.

For each viewpoint $i \in \mathcal{N}$, we list the transmission rate gain compared with the MEC downloading, local caching and computing costs in Table III of each service route, which is obtained via directly generalizing Table II. In the following, we will show that Problem 3 is NP-hard in strong sense and transform Problem 3 into an equivalent IQP, which can be solved via CCCP efficiently.

A. Computational Intractability

To show that Problem 3 is NP-hard in strong sense, we transform Problem 3 into a multiple choice multiple dimensional knapsack problem (MMKP) equivalently. For each viewpoint $i \in \mathcal{N}$ and service route $j \in \{1,2,3,4\}$, introduce variable $x_{i,j} \in \{0,1\}$ where $x_{i,j} = 1$ indicates that the request for viewpoint i is served via the j-th route and $x_{i,j} = 0$ otherwise. Based on Table II, $\left(\mathbf{c}^{O^*}, \mathbf{c}^{I^*}, \mathbf{d}^*\right)$ can be obtained from $(x_{i,j})_{i \in \mathcal{N}, j \in \{1,2,3,4\}}$. Without loss of equivalence, Problem 3 can be rewritten as Problem 4.

Problem 4 (Equivalent Joint Policy Optimization):

$$\max_{(x_{i,j})_{i \in \mathcal{N}, j \in \{1,2,3,4\}}} \sum_{i=1}^{N} \sum_{j=1}^{4} v_{i,j} x_{i,j}$$

$$s.t. \sum_{i=1}^{N} \sum_{j=1}^{4} w_{i,j}^1 x_{i,j} \leq C', \quad (23)$$

TABLE III
GAINS VS. CACHING AND COMPUTING COSTS IN HETEROGENEOUS SCENARIO

Route	Joint Decision	Rate Gain	Caching	Computing
Route 1	Local 3D caching $c_i^O = 1, c_i^I = 0, d_i = 0$	$v_{i,1} = P_i R_i^S$	$w_{i,1}^1 = \alpha D_i^I$	$w_{i,1}^2 = 0$
Route 2	Local computing with local 2D caching $c_i^O = 0, c_i^I = 1, d_i = 1$	$v_{i,2} = P_i R_i^S$	$w_{i,2}^1 = D_i^I$	$w_{i,2}^2 = P_i \frac{k D_i^I w_i f_V^2}{\tau}$
Route 3	Local computing without local caching $c_i^O = 0, c_i^I = 0, d_i = 1$	$v_{i,3} = P_i \left(R_i^S - R_i^V\right)$	$w_{i,3}^1 = 0$	$w_{i,3}^2 = P_i \frac{k D_i^I w_i f_V^2}{\tau}$
Route 4	MEC downloading $c_i^O = 0, c_i^I = 0, d_i = 0$	$v_{i,4} = 0$	$w_{i,4}^1 = 0$	$w_{i,4}^2 = 0$

$$\sum_{i=1}^{N}\sum_{j=1}^{4} w_{i,j}^2 x_{i,j} \le \bar{P}, \tag{24}$$

$$\sum_{j=1}^{4} x_{i,j} = 1, \quad i \in \mathcal{N}, \tag{25}$$

$$x_{i,j} \in \{0,1\}, \quad i \in \mathcal{N},\ j \in \{1,2,3,4\}, \tag{26}$$

where

$$v_{i,j} \triangleq \begin{cases} P_i R_i^S & j = 1,2, \\ P_i \left(R_i^S - R_i^V\right) & j = 3, \\ 0 & j = 4, \end{cases} \tag{27}$$

denotes the transmission rate gain for the choice of j for viewpoint i,

$$w_{i,j}^1 \triangleq \begin{cases} \alpha D_i^I & j = 1, \\ D_i^I & j = 2, \\ 0 & j = 3,4, \end{cases} \tag{28}$$

denotes the caching cost for the choice of j for viewpoint i, and

$$w_{i,j}^2 \triangleq \begin{cases} P_i \dfrac{k D_i^I w_i f_V^2}{\tau} & j = 2,3, \\ 0 & j = 1,4, \end{cases} \tag{29}$$

denotes the power cost for the choice of j for viewpoint i.

We can see that Problem 4 corresponds to a 4-choice 2-dimensional knapsack problem. Since MMKP is NP-hard in strong sense [36], we conclude that Problem 3 is NP-hard in strong sense.

B. Equivalent IQP and CCCP

In the following, we transform Problem 4 into an equivalent linearly constrained IQP and solve it using CCCP. First, without loss of equivalence, (26) can be rewritten as

$$x_{i,j} \in [0,1], \quad i \in \mathcal{N},\ j \in \{1,2,3,4\}, \tag{30}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{4} x_{i,j}(1 - x_{i,j}) \le 0. \tag{31}$$

Then, by substituting (26) with (30) and (31), we transform Problem 4 into Problem 5 equivalently.

Problem 5 (Equivalent Joint Policy Optimization):

$$\min_{(x_{i,j})_{i\in\mathcal{N},j\in\{1,2,3,4\}}} \sum_{i=1}^{N}\sum_{j=1}^{4} -v_{i,j}x_{i,j}$$
$$s.t.\ (23), (24), (25), (30), (31).$$

Note that Problem 5 is a continuous optimization problem, the computation complexity of which is much less than that of solving Problem 4 directly. However, considering $\sum_{i=1}^{N}\sum_{j=1}^{4} x_{i,j}(1 - x_{i,j})$ in (31) is a concave function, (31) is not a convex constraint and thus obtaining an efficient algorithm for solving Problem 5 is still very challenging.

Next, to facilitate the solution, we transform Problem 5 into Problem 6 by penalizing the concave constraint in (31) to the objective function.

Problem 6 (Penalized Joint Policy Optimization):

$$\min_{(x_{i,j})_{i\in\mathcal{N},j\in\{1,2,3,4\}}} \sum_{i=1}^{N}\sum_{j=1}^{4} -v_{i,j}x_{i,j} - \mu\sum_{i=1}^{N}\sum_{j=1}^{4} x_{i,j}(x_{i,j} - 1)$$
$$s.t.\ (23), (24), (25), (30),$$

with the penalty parameter $\mu > 0$. Denote with $\bar{R}(\mu)$ the optimal objective value.

Note that the objective function of Problem 6 is a difference of a linear function and a quadratic convex function, and the constraints of Problem 6 are linear. From [13], Problem 6 is an IQP, a special case of general difference of convex (DC) problem, and the local optima of Problem 6 can be obtained in finite steps via DC algorithms (DCA). In addition, since the second term of the objective function of Problem 6 is differentiable, DCA exactly reduces to CCCP [39], as shown in Algorithm 1. CCCP involves iteratively solving a sequence of convex problems, each of which is obtained via linearizing the second term of the objective function of IQP. Specifically, at each iteration t, we approximate $\sum_{i=1}^{N}\sum_{j=1}^{4} x_{i,j}\left(x_{i,j} - 1\right)$ with $\sum_{i=1}^{N}\sum_{j=1}^{4} x_{i,j}^{(t)}\left(x_{i,j}^{(t)} - 1\right) + \sum_{i=1}^{N}\sum_{j=1}^{4}\left(2x_{i,j}^{(t)} - 1\right)\left(x_{i,j} - x_{i,j}^{(t)}\right)$. Thus, as for our problem, CCCP involves iteratively solving a sequence of linear problems, as shown in Algorithm 1.

Last, based on Theorem 1 in [37], we show the equivalence between Problem 5 and Problem 6 in the following lemma.

Lemma 1 (Exact Penalty): For all $\mu > \mu_0$ where

$$\mu_0 \triangleq \frac{\sum_{i=1}^{N}\sum_{j=1}^{4} -v_{i,j}x_{i,j}^0 - \bar{R}(0)}{\max_{\boldsymbol{x}}\left\{\sum_{i=1}^{N}\sum_{j=1}^{4} x_{i,j}(x_{i,j}-1) : (23), (24), (25), (30)\right\}}, \tag{32}$$

with any $(x_{i,j}^0)_{i\in\mathcal{N},j\in\{1,2,3,4\}}$ satisfying (23), (24), (25) and (30), Problem 6 and Problem 5 have the same optimal solution.

Algorithm 1 CCCP for Solving Problem 6

1: **Initialization**. Find an initial feasible point $\mathbf{x}^{(0)}$ of Problem 6 and set $t = 0$.
2: **Repeat**
3: Set $\mathbf{x}^{(t+1)}$ to be an optimal solution to the following convex problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & G(\mathbf{x}) - \mu \check{H}(\mathbf{x}; \mathbf{x}^{(t)}) \\ s.t. \quad & (23), (24), (25), (30), \end{aligned}$$

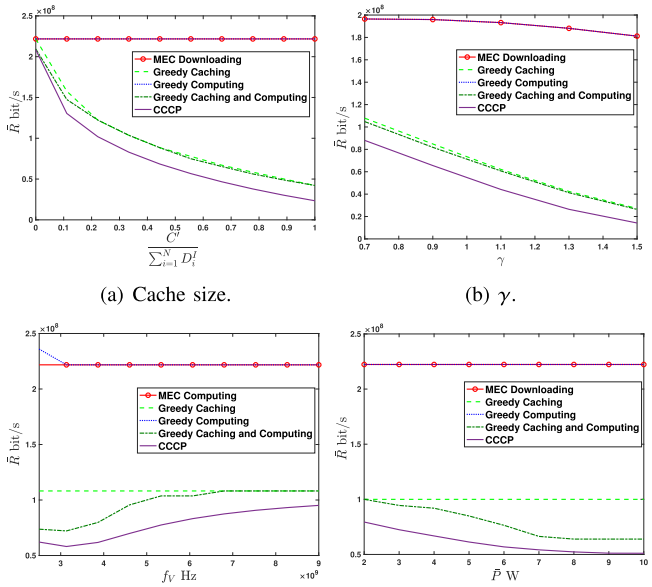
where $G(\mathbf{x}) \triangleq \sum_{i=1}^{N} \sum_{j=1}^{4} -v_{i,j} x_{i,j}$ and $\check{H}\left(\mathbf{x}; \mathbf{x}^{(t)}\right) \triangleq \sum_{i=1}^{N} \sum_{j=1}^{4} x_{i,j}^{(t)} \left(x_{i,j}^{(t)} - 1\right) + \sum_{i=1}^{N} \sum_{j=1}^{4} \left(2x_{i,j}^{(t)} - 1\right) \left(x_{i,j} - x_{i,j}^{(t)}\right)$.
4: Set $t \leftarrow t + 1$.
5: **until** $\left[G\left(\mathbf{x}^{(t-1)}\right) - \mu \check{H}\left(\mathbf{x}^{(t-1)}; \mathbf{x}^{(t-2)}\right)\right] - \left[G\left(\mathbf{x}^{(t)}\right) - \mu \check{H}\left(\mathbf{x}^{(t)}; \mathbf{x}^{(t-1)}\right)\right] \leq \delta$.

Proof: Lemma 1 can be obtained directly from Theorem 1 in [37]. □

Lemma 1 illustrates that Problem 6 is equivalent to Problem 5 if the penalty parameter μ is sufficiently large. Thus, we can solve Problem 6 instead of Problem 5 by using CCCP. However, it may not always be a feasible solution to Problem 5. In order to obtain a global optima of Problem 5, we obtain multiple local optimal solutions of Problem 6 via performing CCCP multiple times, each with a unique initial feasible point of Problem 6, and then choose the one which achieves the minimum average value [38].

Remark 3 (CCCP Optimality and Computation Complexity Analysis): The proposed CCCP guarantees a local optima of Problem 4 and a global optima via performing it multiple times, each with a different starting point. This is because transforming Problem 4 to Problem 6 is without loss of equivalence, and the local optima of Problem 6 can be obtained in finite steps via CCCP [37], [38]. For the computation complexity, based on CCCP, the optimization problem is reduced into a finite sequence of linear programming problems, each of which is with fixed dimension and can be solved in $\mathcal{O}(N)$ time [40]. Compared with the computation complexity of the original problem, i.e., $\mathcal{O}(2^{4N})$, CCCP helps greatly reduce the computation complexity.

Remark 4 (Practical Implementation of Algorithm 1): Firstly, we consider the caching and computing decisions are made offline based on priori knowledge of user demand process and the related system parameters. Next, the obtained caching decisions (both 3D and 2D) are implemented when the traffic load is relatively low (e.g., nighttime) and then the cache state at the mobile VR device remains unchanged. In this way, the transmission cost of caching is negligible in the long term perspective [27], [35]. On the other hand, the obtained computing decision table can be stored in the mobile VR device. When a user starts using the mobile VR device, at each time slot, the FOV request gets satisfied based on the cache state at the mobile VR device and according to the computing decision table stored in the mobile VR device.



(a) Cache size. (b) γ.

(c) Computation frequency when $\bar{P}$ = 10 W. (d) Average power under CCCP.

Fig. 5. Heterogeneous scenario analysis at $f_V = 3$ G Hz, $N = 100$, $D_i^I \in [1, 3]$ M bits, $\alpha = 2$, $w = 10$ cycle/bit, $\tau = 20$ ms, $P_i \propto \frac{1}{i^\gamma}$ with $\gamma = 0.8$, $\frac{C'}{\sum_{i=1}^{N} D_i^I} = 0.3$, $\bar{P} = 2$ W, $\mu = 10^5$ unless otherwise stated.

C. Numerical Results

In this section, we illustrate the performance of CCCP via numerical results, as shown in Fig. 5. Specifically, CCCP is obtained via performing Algorithm 1 with $\delta = 0.001$ 100 times, each starting with a random initial feasible point, and then selecting the local optima with the lowest average transmission rate value. We compare it with the following three baselines:

- MEC downloading: 3D FOVs of all the viewpoints are downloaded from the MEC server, i.e., $c_i^O = 0$, $c_i^I = 0, d_i = 0$ for all $i \in \mathcal{N}$;
- Greedy caching: 3D FOVs are cached at the mobile VR device via greedy algorithm, as illustrated in Algorithm 2;
- Greedy computing: 2D FOVs are computed at the mobile VR device via greedy algorithm, as illustrated in Algorithm 2;
- Greedy caching and computing: first, local computing with local 2D caching is determined via greedy algorithm; secondly, if there still exists underutilized cache size, then 3D FOVs of the rest of viewpoints are cached at the mobile VR device via greedy algorithm. Otherwise, if there still exists underutilized computing size, local computing without caching is decided via greedy algorithm, as illustrated in Algorithm 2.

Fig. 5 (a) illustrates the average rate versus cache size C' in the heterogeneous scenario. We can see that CCCP exhibits great promises in saving communication bandwidth compared with the baselines. For example, compared with greedy 3D caching, greedy computing as well as greedy caching and computing, CCCP brings larger transmission rate gain over MEC

Algorithm 2 Greedy Caching, Greedy Computing, Greedy Caching and Computing

1: Sort $\mathcal{N}$ according to $v_i,\ i \in \mathcal{N}$ in descending order:

$$v_i \triangleq \begin{cases} \frac{P_i R_i^S}{D_i^O} & \text{Greedy Caching} \\ \frac{P_i(R_i^S - R_i^V)}{P_i k D_i^I w_i f_V^2/\tau} & \text{Greedy Computing} \\ \frac{P_i R_i^S}{D_i^I + P_i k D_i^I w_i f_V^2/\tau} & \text{Greedy Caching and Computing.} \end{cases} \tag{33}$$

2: Obtain the split index s_c satisfying $\sum_{j=1}^{s_c-1} w_{\lfloor j \rfloor} \leq S$ and $\sum_{j=1}^{s_c} w_{\lfloor j \rfloor} > S$, where $\lfloor j \rfloor$ represents the index $i \in \mathcal{N}$ with the j-th maximal value of v_i, w_i is defined as

$$w_i \triangleq \begin{cases} D_i^O & \text{Greedy Caching} \\ P_i k D_i^I w_i f_V^2/\tau & \text{Greedy Computing} \\ D_i^I / P_i k D_i^I w_i f_V^2/\tau & \\ & \text{Greedy Caching and Computing,} \end{cases} \tag{34}$$

and S is defined as

$$S \triangleq \begin{cases} C & \text{Greedy Caching} \\ \bar{P} & \text{Greedy Computing} \\ C \ / \ \bar{P} & \text{Greedy Caching and Computing.} \end{cases} \tag{35}$$

* For greedy caching and computing, s_c denotes the split index satisfying $\sum_{j=1}^{s_c-1} D_{\lfloor j \rfloor}^I \leq C$ and $\sum_{j=1}^{s_c} D_{\lfloor j \rfloor}^I > C$ or $\sum_{j=1}^{s_c-1} P_{\lfloor j \rfloor} k D_{\lfloor j \rfloor}^I w_{\lfloor j \rfloor} f_V^2/\tau \leq \bar{P}$ and $\sum_{j=1}^{s_c} P_{\lfloor j \rfloor} k D_{\lfloor j \rfloor}^I w_{\lfloor j \rfloor} f_V^2/\tau > \bar{P}$.

3: Caching and computing decision making:
Greedy caching: set $c_{\lfloor j \rfloor}^O = 1, c_{\lfloor j \rfloor}^I = 0, d_{\lfloor j \rfloor} = 0,\ j \in \{1, \cdots, s_c\}$; $c_{\lfloor j \rfloor}^O = 0, c_{\lfloor j \rfloor}^I = 0, d_{\lfloor j \rfloor} = 0$, otherwise;
Greedy computing: set $c_{\lfloor j \rfloor}^O = 0, c_{\lfloor j \rfloor}^I = 0, d_{\lfloor j \rfloor} = 1,\ j \in \{1, \cdots, s_c\}$; $c_{\lfloor j \rfloor}^O = 0, c_{\lfloor j \rfloor}^I = 0, d_{\lfloor j \rfloor} = 0$, otherwise;
Greedy caching and computing: set $c_{\lfloor j \rfloor}^O = 0, c_{\lfloor j \rfloor}^I = 1, d_{\lfloor j \rfloor} = 1$ for all $j \in \{1, \cdots, s_c - 1\}$ and $c_{\lfloor j \rfloor}^O = 0, c_{\lfloor j \rfloor}^I = 0, d_{\lfloor j \rfloor} = 0$, otherwise; if there still exists underutilized cache size, i.e., $\sum_{j=1}^{s_c-1} D_{\lfloor j \rfloor}^I < C$, then 3D FOVs of the rest of viewpoints are cached at the mobile VR device via greedy caching. Otherwise, if $\sum_{j=1}^{s_c-1} P_{\lfloor j \rfloor} k D_{\lfloor j \rfloor}^I w_{\lfloor j \rfloor} f_V^2/\tau < \bar{P}$, then local computing without caching is decided via greedy computing.

downloading (e.g., 47.7%, 0 %, 50% vs 63.6% at $\frac{C'}{\sum_{i=1}^N D_i^I} = 30\%$). Fig. 5 (b) illustrates the average rate versus γ. We can see that the average rate monotonically decreases with γ, since the probability of the requested FOV is popular and is cached or computed locally increases with γ. Fig. 5 (c) illustrates the average rate versus f_V. We can see that the average rate first decreases and then increases with f_V. This is because when f_V is relatively small, increasing f_V decreases the rate required via local computing without caching. Otherwise, increasing f_V decreases the number of FOV requests that can be computed locally due to the limitation of average available power. Fig. 5 (d) illustrates the average rate versus $\bar{P}$ and k. We can see that $\bar{R}$ decreases with $\bar{P}$, since the number of FOVs that can be computed locally $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ increases with $\bar{P}$. $\bar{R}$ increases with k, since the number of FOVs that can be computed locally $\frac{N\bar{P}\tau}{kf_V^2 D^I w}$ decreases with k.

VI. Conclusion and Future Work Discussion

In this paper, we develop a novel MEC-based mobile VR delivery framework by jointly utilizing the caching and computing capacities of the mobile VR device. When FOVs are homogeneous, a closed-form expression for the optimal joint policy is derived, which reveals a fundamental tradeoff between the three primary resouces, i.e., communications, caching and computing. The tradeoff results show that:

- When $f_V < F$, R^* increases with f_V if $\frac{N\bar{P}\tau}{kf_V^2 D^I w} \leq C$ and stays unchanged with f_V, otherwise; R^* decreases with C at the rate of $\frac{R_S}{\alpha N}$ when $\frac{N\bar{P}\tau}{kf_V^2 D^I w} \leq C$ and $\frac{R_S}{N}$, otherwise;
- When $F \leq f_V$, R^* first decreases and then increases with f_V if $\frac{N\bar{P}\tau}{kf_V^2 D^I w} > C$ and increases with f_V, otherwise; R^* decreases with C at the rate of $\frac{R_S}{\alpha N}$ when $\frac{N\bar{P}\tau}{kf_V^2 D^I w} \leq C$ and $\frac{R_V}{N}$, otherwise.

In the heterogeneous scenario, we transform the NP-hard problem into an equivalent IQP and solve it via CCCP, which obtains a local optima and is shown to achieve good performance in numerical results. Future work directions may include extending the current work into scenarios with more than one video streaming, multiple mobile devices and multiple servers, dynamic caching as well as implementing the proposed algorithm in practical system.

Appendix A
Proof of Lemma 1

When $f_V < F$, $R_S - R_V < 0$ and the objective function of Problem 2 increases with $d - \min\{c^I, d\}$. Thus, we can see that $d - \min\{c^I, d\} = 0$, i.e., $d \leq c^I$. In addition, based on Property 2, by replacing c^O with $\frac{C - c^I}{\alpha}$, Problem 2 can be rewritten as

$$\begin{aligned} \textit{Problem 7}: \ \min_{c^I, d} \ & R_S\left(1 - \frac{C}{\alpha N}\right) + \frac{R_S}{\alpha N} c^I - \frac{R_S}{N} d \\ s.t. \ & c^I \in \{0, 1, \cdots, C\}, \qquad (36) \\ & d \in \left\{0, 1, \cdots, \min\left\{c^I, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}\right\}. \qquad (37) \end{aligned}$$

In the following, we analyze the optimal solution to Problem 7 from the following two aspects.

- If $c^I \leq \frac{N\bar{P}\tau}{kf_V^2 D^I w}$, (36) and (37) can be rewritten as

$$c^I \in \left\{0, 1, \cdots, \min\left\{C, \frac{N\bar{P}\tau}{kf_V^2 D^I w}\right\}\right\}, \tag{38}$$

$$d \in \left\{0, 1, \cdots, c^I\right\}. \tag{39}$$

Since the objective function of Problem 7 decreases with d, we have $d = c^I$ without loss of optimality.

By replacing d with c^I, and (36) with (38), Problem 7 can be rewritten as

$$\textit{Problem 8}: \min_{c^I} \; R_S\left(1-\frac{C}{\alpha N}\right)-(\alpha-1)\frac{R_S}{\alpha N}c^I$$
$$s.t.\ c^I \in \left\{0,1,\cdots,\min\left\{C,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}\right\}.$$

Since $\alpha > 1$, we can see that the objective function of Problem 8 decreases with c^I, and thus $c^{I^*}=\min\left\{C,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}$. Accordingly, we have $d^*=\min\left\{C,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}$ and $c^{O^*}=\frac{C-c^{I^*}}{\alpha}$.

- If $c^I \geq \frac{N\bar{P}\tau}{kf_V^2D^Iw}$, (36) and (37) can be rewritten as

$$c^I \in \left\{\frac{N\bar{P}\tau}{kf_V^2D^Iw},\cdots,C\right\}, \tag{40}$$
$$d \in \left\{0,1,\cdots,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}. \tag{41}$$

Since the objective function of Problem 7 decreases with d and increases with c^I, we have $d^*=\frac{N\bar{P}\tau}{kf_V^2D^Iw}$ and $c^{I^*}=\frac{N\bar{P}\tau}{kf_V^2D^Iw}$. Accordingly, we have c^{O^*} via $c^{O^*}=\frac{C-c^{I^*}}{\alpha}$. Since $c^I \geq \frac{N\bar{P}\tau}{kf_V^2D^Iw}$ holds only when $C \geq \frac{N\bar{P}\tau}{kf_V^2D^Iw}$, we have $c^{I^*}=\min\left\{C,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}$, $d^*=\min\left\{C,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}$ and $c^{O^*}=\frac{C-c^{I^*}}{\alpha}$.

Thus, (12), (13) and (14) hold. By substituting (12), (13) and (14) into the objective function of Problem 2, (15) holds. The proof ends.

APPENDIX B
PROOF OF LEMMA 2

When $F \leq f_V$, $R_V \leq R_S < \alpha R_V$. We analyze the optimal solution to Problem 2 from the following two aspects.

- If $c^I \leq d$, Problem 2 can be rewritten as

$$\min_{c^I,d} \; R_S\left(1-\frac{C}{\alpha N}\right)-\frac{\alpha R_V-R_S}{\alpha N}c^I-\frac{R_S-R_V}{N}d$$
$$s.t.\ c^I \in \{0,1,\cdots,\min\{d,C\}\}, \tag{42}$$
$$d \in \left\{0,1,\cdots,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}. \tag{43}$$

Since $R_V \leq R_S < \alpha R_V$, we have the objective function decreases with c^I and d. Thus, we have $c^{I^*}=\min\{d^*,C\}$, $d^*=\frac{N\bar{P}\tau}{kf_V^2D^Iw}$ and $c^{O^*}=\frac{C-c^{I^*}}{\alpha}$.

- If $c^I \geq d$, Problem 2 can be rewritten as

$$\min_{c^I,d} \; R_S\left(1-\frac{C}{\alpha N}\right)+\frac{R_S}{\alpha N}c^I-\frac{R_S}{N}d$$
$$s.t.\ c^I \in \{d,\cdots,C\}, \tag{44}$$
$$d \in \left\{0,1,\cdots,\frac{N\bar{P}\tau}{kf_V^2D^Iw}\right\}. \tag{45}$$

Since the objective function increases with c^I and decreases with d, we have $c^{I^*}=d^*$, $d^*=\frac{N\bar{P}\tau}{kf_V^2D^Iw}$ and $c^{O^*}=\frac{C-c^{I^*}}{\alpha}$. In addition, since $c^{I^*}=d^*$ holds only when $C \geq d^*$, c^{I^*} can also be rewritten as $c^{I^*}=\min\{d^*,C\}$.

Thus, (16), (17) and (18) hold. By substituting (16), (17) and (18) into the objective function of Problem 2, (19) holds. The proof ends.

REFERENCES

[1] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communication, computing and caching for mobile VR delivery: Modeling and trade-off," in *Proc. IEEE ICC*, May 2018, pp. 1–7.

[2] R. Begole. *Why the Internet Pipes Will Burst When Virtual Reality Takes Off*. Accessed: Feb. 2016. [Online]. Available: https://www.forbes.com/sites/valleyvoices/2016/02/09/why-the-internet-pipes-will-burst-if-virtual-reality-takes-off/

[3] ABI Research Qualcomm. (Feb. 2017). *Augmented and Virtual Reality: The First Wave of 5G Killer Apps*. [Online]. Available: https://www.qualcomm.com/documents/augmented-and-virtual-reality-first-wave-5g-killer-apps

[4] Juniper. (2017). *Virtual Reality Markets Hardware, Content & Accessories 2017–2022*. [Online]. Available: https://www.juniperresearch.com/researchstore/innovation-disruption/virtual-reality/hardware-content-accessories

[5] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[6] A. Mahzari, A. T. Nasrabadi, A. Samiei, and R. Prakash, "FoV-aware edge caching for adaptive 360° video streaming," in *Proc. 26th Int. Conf. Multimedia*, Oct. 2018, pp. 173–181.

[7] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation," in *Proc. 8th Multimedia Syst. Conf.*, 2017, pp. 261–271.

[8] Huawei. (2018). *Cloud VR Solution White Paper*. [Online]. Available: https://www.huawei.com/en/press-events/news/2018/9/cloud-vr-solution-white-paper

[9] Y. Bao, T. Zhang, A. Pande, H. Wu, and X. Liu, "Motion prediction-based multicast for 360-degree video transmissions," in *Proc. IEEE SECON*, Jun. 2017, pp. 1–9.

[10] S. Mangiante, G. Klas, A. Navon, G. Zhuang, J. Ran, and M. D. Silva, "VR is on the edge: How to deliver 360° videos in mobile networks," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, Aug. 2017, pp. 30–35.

[11] S. Reichelt, R. Häussler, G. Fütterer, and N. Leister, "Depth cues in human visual perception and their realization in 3D displays," *Proc. SPIE*, vol. 7690, pp. 1–6, May 2010.

[12] X. Cao, A. C. Bovik, Y. Wang, and Q. Dai, "Converting 2D video to 3D: An efficient path to a 3D experience," *IEEE Multimedia*, vol. 18, no. 4, pp. 12–17, Apr. 2011.

[13] L. T. H. An and P. D. Tao, "Solving a class of linearly constrained indefinite quadratic problems by D.C. Algorithms," *J. Global Optim.*, vol. 11, no. 3, pp. 253–285, 1997.

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

[15] T. El-Ganainy and M. Hefeeda, "Streaming virtual reality content," 2016, *arXiv:1612.08350*. [Online]. Available: https://arxiv.org/abs/1612.08350

[16] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. 5th Workshop All Things Cellular Oper. Appl. Challenges*, Oct. 2016, pp. 1–6.

[17] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Fixation prediction for 360° video streaming in head-mounted virtual reality," in *Proc. NOSSDAV*, Jun. 2017, pp. 67–72.

[18] M. Chen, W. Saad, and C. Yin, "Resource management for wireless virtual reality: Machine learning meets multi-attribute utility," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–7.

[19] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5621–5635, Nov. 2018.

[20] H. Liu, Z. Chen, and L. Qian, "The three primary colors of mobile systems," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 15–21, Sep. 2016.

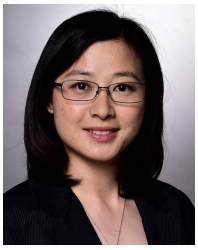
[21] H. Liu, Z. Chen, X. Tian, X. Wang, and M. Tao, "On content-centric wireless delivery networks," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 118–125, Dec. 2014.

[22] M. Erol-Kantarci and S. Sukhmani, "Caching and computing at the edge for mobile augmented reality and virtual reality (AR/VR) in 5G," *Ad Hoc Netw.*, vol. 223, pp. 169–177, Jan. 2018.

[23] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.
[24] A. Ndikumana, S. Ullah, T. LeAnh, N. H. Tran, and C. S. Hong, "Collaborative cache allocation and computation offloading in mobile edge computing," in *Proc. IEEE APNOMS*, Sep. 2017, pp. 366–369.
[25] A. Ndikumana *et al.*, "Joint communication, computation, caching, and control in big data multi-access edge computing," 2018, *arXiv:1803.11512*. [Online]. Available: https://arxiv.org/abs/1803.11512
[26] S. Kim, "5G network communication, caching, and computing algorithms based on the two-tier game model," *ETRI J.*, vol. 40, no. 1, pp. 61–71, Feb. 2018.
[27] J. Chakareski, "VR/AR immersive communication: Caching, edge computing, and transmission trade-offs," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, Los Angeles, CA, USA, Aug. 2017, pp. 36–41.
[28] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018, pp. 1–6.
[29] X. Yang *et al.*, "Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff," *IEEE Access*, vol. 6, pp. 16665–16677, 2018.
[30] Accessed: Apr. 2017. [Online]. Available: https://www.roadtovr.com/understanding-pixel-density-retinal-resolution-and-why-its-important-for-vr-and-ar-headsets/
[31] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, nos. 2–3, pp. 203–221, 1996.
[32] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1–6.
[33] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, and N. Dai, "Furion: Engineering high-quality immersive virtual reality on today's mobile devices," in *Proc. MobiCom*, 2017, pp. 409–421.
[34] *Energy Calcula*. [Online]. Available: http://energyusecalculator.com/electricity.htm
[35] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
[36] M. Hifi, M. Michrafy, and A. Sbihi, "Heuristic algorithms for the multiple-choice multidimensional knapsack problem," *J. Oper. Res. Soc.*, vol. 55, no. 12, pp. 1323–1332, 2004.
[37] H. A. Le Thi, T. P. Dinh, and H. Van Ngai, "Exact penalty and error bounds in DC programming," *J. Global Optim.*, vol. 52, no. 3, pp. 509–535, 2012.
[38] H. A. Le Thi, T. P. Dinh, H. M. Le, and X. T. Vo, "DC approximation approaches for sparse optimization," *Eur. J. Oper. Res.*, vol. 244, no. 1, pp. 26–46, 2015.
[39] B. Sriperumbudur and G. Lanckriet, "On the convergence of concave-convex procedure," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 1759–1767.
[40] N. Megiddo, "Linear programming in linear time when the dimension is fixed," *J. ACM*, vol. 31, no. 1, pp. 114–127, 1984.

Yaping Sun received the B.E. degree in communication engineering from Xidian University, China, in 2015. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Shanghai Jiao Tong University, China. Her research interest includes mobile 3C networks.

Zhiyong Chen received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011. From 2009 to 2011, he was a Visiting Ph.D. Student at the Department of Electronic Engineering, University of Washington, Seattle, WA, USA. He is currently an Associate Professor with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University (SJTU), Shanghai, China. His research interests include mobile communications-computing-caching (3C) networks, mobile VR/AR delivery, and mobile AI systems. He currently serves as an Associate Editor for IEEE ACCESS, and served as the Student Volunteer Chair for the IEEE ICC 2019, the Publicity Chair for the IEEE/CIC ICCC 2014, and a TPC member for major international conferences.



Meixia Tao (S'00–M'04–SM'10–F'19) received the B.S. degree in electronic engineering from Fudan University, Shanghai, China, in 1999, and the Ph.D. degree in electrical and electronic engineering from The Hong Kong University of Science and Technology in 2003.

She is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. Prior to that, she was a member of professional staff at the Hong Kong Applied Science and Technology Research Institute (2003–2004), and a Teaching Fellow, then an Assistant Professor at the Department of Electrical and Computer Engineering, National University of Singapore from 2004 to 2007. Her current research interests include wireless caching, edge computing, physical layer multicasting, and resource allocation.

Dr. Tao served as a member of the Executive Editorial Committee of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. She was a recipient of the IEEE Marconi Prize Paper Award in 2019, the IEEE Heinrich Hertz Award for Best Communications Letters in 2013, the IEEE/CIC International Conference on Communications in China (ICCC) Best Paper Award in 2015, and the International Conference on Wireless Communications and Signal Processing (WCSP) Best Paper Award in 2012. She also receives the IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award in 2009. She serves as the Symposium Oversight Chair of the IEEE ICC 2019, the Symposium Co-Chair of the IEEE GLOBECOM 2018, the TPC Chair of the IEEE/CIC ICCC 2014, and the Symposium Co-Chair of the IEEE ICC 2015. She was on the Editorial Board of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2007–2011), and IEEE TRANSACTIONS ON COMMUNICATIONS (2012–2018), IEEE COMMUNICATIONS LETTERS (2009–2012), and IEEE WIRELESS COMMUNICATIONS LETTERS (2011–2015).

Hui Liu (F'09) received the B.S. degree in electrical engineering from Fudan University, Shanghai, China, in 1988, and the Ph.D. degree in electrical engineering from The University of Texas at Austin, Austin, TX, USA, in 1995. He was a Full Professor and the Associate Chairman of the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, and the Chair Professor and the Associate Dean of the School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University. He was one of the principal designers of the 3G TD-SCDMA mobile technologies. He was the Founder of Adaptix, which pioneered the development of OFDMA-based mobile broadband networks (mobile WiMAX and 4G LTE). He is currently the President and the CTO of Silkwave Holdings, and an Affiliate Professor with the University of Washington. He has authored over 80 journal articles and two textbooks and holds 70 awarded patents. His research interests include broadband wireless networks, satellite communications, digital broadcasting, and multimedia signal processing. He contributed to the global standards for broadband cellular and mobile broadcasting. He was a recipient of the 1997 NSF CAREER Award, the Gold Prize Patent Award in China, three IEEE best conference paper awards, and the 2000 ONR Young Investigator Award.