Graph Learning Based Head Movement Prediction for Interactive 360 Video Streaming

Xue Zhang, *Member, IEEE*, Gene Cheung, *Fellow, IEEE*, Yao Zhao, *Senior Member, IEEE*, Patrick Le Callet, *Fellow, IEEE*, Chunyu Lin, and Jack Z. G. Tan

***Abstract*— Ultra-high definition (UHD) 360 videos encoded in fine quality are typically too large to stream in its entirety over bandwidth (BW)-constrained networks. One popular approach is to interactively extract and send a spatial sub-region corresponding to a viewer's current field-of-view (FoV) in a head-mounted display (HMD) for more BW-efficient streaming. Due to the non-negligible round-trip-time (RTT) delay between server and client, accurate head movement prediction foretelling a viewer's future FoVs is essential. In this paper, we cast the head movement prediction task as a sparse directed graph learning problem: three sources of relevant information—collected viewers' head movement traces, a 360 image saliency map, and a biological human head model—are distilled into a view transition Markov model. Specifically, we formulate a constrained maximum a posteriori (MAP) problem with likelihood and prior terms defined using the three information sources. We solve the MAP problem alternately using a hybrid iterative reweighted least square (IRLS) and Frank-Wolfe (FW) optimization strategy. In each FW iteration, a linear program (LP) is solved, whose runtime is reduced thanks to warm start initialization. Having estimated a Markov model from data, we employ it to optimize a tile-based 360 video streaming system. Extensive experiments show that our head movement prediction scheme noticeably outperformed existing proposals, and our optimized tile-based streaming scheme outperformed competitors in rate-distortion performance.**

***Index Terms*— 360 video streaming, directed graph learning, head movement prediction.**

I. Introduction

LEVERAGING on the recent advances in camera technologies and image-stitching algorithms [1], ultra-high definition (UHD) 360 virtual reality (VR) videos are now easily generated and observed on head-mounted displays (HMD) such as Oculus Quest[1] or HTC Vive Cosmos[2] for an immersive visual experience. However, a recorded 360 video is typically in high spatial resolution (*e.g.*, 10K resolution 10240×4320), and if encoded in fine quality, it cannot be streamed in its entirety over bandwidth (BW)-limited networks. It will be even more problematic in the future, since spatial resolution is expected to increase further to fulfill the visual acuity of human eyes (60 pixels per visual angle degree) [2].

One popular approach is to extract in real-time and transport only a spatial sub-region of the video corresponding to a viewer's current field-of-view (FoV) in an *interactive streaming* scenario [3]–[7]. See Fig. 1 for an illustration. For example, [3]–[6] first encode a 360 video into independent *tiles* covering non-overlapping spatial areas, and then transmit tile combinations corresponding to viewers' current FoVs. Note that ISO/IEC MPEG recently launched a standard on "Coded Representation of Immersive Media", namely MPEG-I [8], specifying the media transmission chain. MPEG-I emphasizes viewport-dependent random access and requires splitting the 360 video into segments to represent the local scene. It further validates the practicality of the interactive streaming scenario.

However, in real network settings, the *round-trip-time* (RTT) delay between server and client can be substantial, meaning that an interactive streaming system requires first a head movement prediction scheme foretelling a viewer's future FoV. Recent studies [3], [4], [9], [10] have applied regression models to predict future orientations, where dead-reckoning algorithms analyzing historical samples are used to extrapolate the trends of head movements. However, these methods are extremely sensitive to the time scale—*e.g.*, the prediction accuracy drops precipitously when RTT becomes large.

An alternative approach is data-driven learning [5], [11], [12]. Specifically, FoVs in the future are predicted using neural networks that learn from sensor- and content-related features collected from data [5]. Similarly, a deep reinforcement learning scheme is developed in [12], where offline and online data are combined to predict head movements. However, to train a large number of network parameters, a huge dataset of viewer traces is required, and training is typically specific to particular setups (*e.g.*, RTT mean and variance), leading to inflexible models that are hard to use in practice.

Manuscript received December 24, 2020; revised March 24, 2021; accepted April 7, 2021. Date of publication April 26, 2021; date of current version April 30, 2021. The work of Gene Cheung was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2019-06271 and Grant RGPAS-2019-00110. The work of Yao Zhao and Chunyu Lin was supported by the National Natural Science Foundation of China under Grant U1936212 and Grant 61772066. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aswin Sankaranarayanan. *(Corresponding author: Gene Cheung.)*

Xue Zhang and Gene Cheung are with the Department of EECS, York University, Toronto, M3J 1P3, Canada (e-mail: xuezhang@yorku.ca; genec@yorku.ca).

Yao Zhao and Chunyu Lin are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: yzhao@bjtu.edu.cn; cylin@bjtu.edu.cn).

Patrick Le Callet is with the Computer Science Department, Université de Nantes, 44035 Nantes, France (e-mail: patrick.lecallet@univ-nantes.fr).

Jack Z. G. Tan is with Kandao Technology Company Ltd., Shenzhen 518057, China (e-mail: jack@kandaovr.com).

Digital Object Identifier 10.1109/TIP.2021.3073283

[1]https://www.oculus.com/

[2]https://www.vive.com/

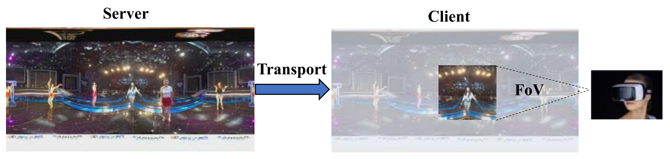


Fig. 1. An interactive 360 video streaming scenario.

Generally, estimating an observer's relative interest in media content is the *visual attention* (VA) problem [13], [14], which has a long history for conventional 2D contents like standard images and videos. Recent studies have targeted 360 contents, particularly driven by the series of ICME Grand Challenge "salient360!" for 360 images in 2017 and 360 videos in 2018. The series investigate how users visually explore 360 contents, establishing: 1) datasets [15], 2) toolbox to facilitate the development of VA models [16], 3) framework to evaluate VA models [17], and 4) ad hoc VA models for 360 contents [18], [19]. Four types of VA models are available for 360 contents: 1) head motion based saliency models that predict viewport heat map; 2) (head+eye)-motion based saliency models that predict ground-truth heat map derived from the movement of the head as well as from the movement of the eyes within the viewport; 3) scanpath models that predict sequences of head-gaze data; and 4) scanpath models that predict from head and eye-movement data. The saliency models aim to detect main objects of general interest, but are unable to customize tracking of individual users' unique observation behaviors. The scanpath models focus on generating a most likely sequence of head movements in a prediction window, rather than the probability distribution of all positions at a future time instant as we propose in this paper.

To predict head movement *at least* one RTT (typically 50 to 300ms) into the future, in this paper, we cast the prediction task as a *sparse directed graph learning* problem, where three relevant information sources—collected viewers' head movement traces, a 360 image saliency map, and a biological head rotation model—are aggregated into a unified Markov model. Specifically, we first discretize head angles in the 360 view into K unique locations, modeled as nodes in a graph. The observer's head movements are thus described stochastically by the *view transition probability matrix* $\mathbf{P} \in \mathbb{R}^{K\times K}$, with *stationary probability vector* $\mathbf{q} \in \mathbb{R}^{K}$. Next, we define likelihood and prior terms for the two optimization variables $\mathbf{P}$ and $\mathbf{q}$ using observed data traces, saliency map, and biological head model, respectively. See Fig. 2 for an illustration.

We formulate a constrained *maximum a posteriori* (MAP) problem to solve for $\mathbf{P}$ and $\mathbf{q}$. We propose a hybrid *iterative reweighted least square* (IRLS) [20] and *Frank-Wolfe* (FW) optimization [21] strategy that alternately optimizes $\mathbf{P}$ and $\mathbf{q}$ until convergence. Unlike popular first-order descent methods like *proximal gradient* (PG) [22] that require a projection at each step, recently revived FW [23], [24] is entirely projection-free. Moreover, the *linear program* (LP) at each step can benefit from warm start using the solution from the previous iteration for initialization, reducing runtime. Having

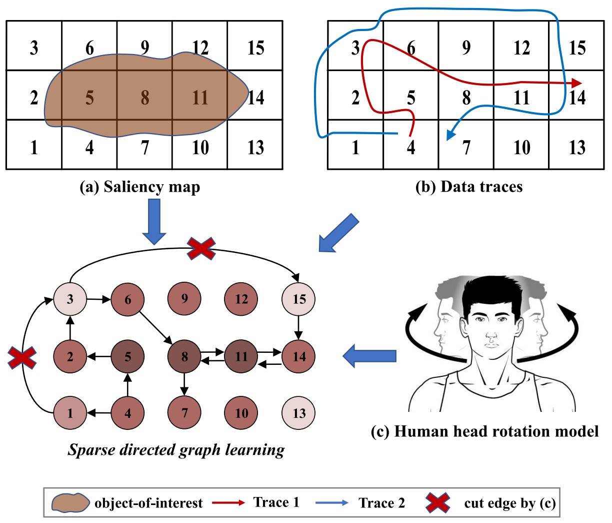


Fig. 2. Example of sparse directed graph learning. Three sources of relevant information are taken into account: a) a saliency map provides stationary probability estimates at the graph nodes, where the darker the color of a node, the larger the stationary probability; b) data traces provide transitions among the nodes, and c) a human head rotation model restricts possible transitions, which cuts down the switches from node 1 to node 3 and from node 3 to node 15 in the graph as examples.

an estimated Markov model from data, we employ it to optimize a tile-based 360 video streaming system. Extensive experiments show that our head movement prediction scheme noticeably outperformed existing proposals [3]–[5], [9]–[12], [16], [25], and our optimized tile-based streaming scheme outperformed competitors [3], [25], [26] in rate-distortion (RD) performance.

The outline of the paper is as follows. We first overview related works in Section II. We then describe our system model in Section III. In Section IV, we discuss our sparse directed graph learning problem and algorithm. In Section V, we discuss our tiling optimization problem and algorithm. Finally, experiments and conclusion are presented in Section VI and VII, respectively.

II. Related Work

To reduce the delivery BW in 360 video streaming system, viewport-adaptive streaming has been investigated. The ideal scenario is to transport only portions of the video corresponding to a user's current FoV in high quality. Towards this end, tile-based encoding and streaming is one practical strategy [6], [26]–[30]. In particular, optimal tiling was proposed in [6], [29], where rate-distortion of the transmitted FoV was optimized. [29] designed a non-uniform tiling method for 360 video streaming, where variable tile sizes were considered to optimize both storage and bandwidth costs. Due to RTT delay, predicting future FoVs is important in practical interactive streaming systems [5], [6]. Specifically, VA prediction can be applied to many aspects of 360 video streaming to improve performance, such as region-of-interest video coding [31], perceived quality assessment [32] and panorama interpolation [33]. We next review related works in two groups: (i) saliency detection; and (ii) head movement prediction.

A. Saliency Detection

Saliency detection for 360 contents has been studied very recently. [18] used 2D saliency prediction models on the viewport images, rather than the equirectangular projection. Then the saliency map of each viewport was reverse-mapped into the rectangular plane and merged to generate a single 360 saliency map. In contrast, handcrafted features were extracted to construct saliency maps of 360 images in [34].

Using large datasets, data-driven models have been developed for saliency map prediction of 360 contents. Specifically, a color dictionary based sparse representation approach for 360 saliency prediction was proposed in [35]: sub-images extracted from the 360 image were further divided into non-overlapping patches, such that sparse representations can be computed using a dictionary trained from 2D images. Thanks to the success of deep learning, many efficient architectures, such as convolutional neural network (CNN), recurrent neural network (RNN), and a special kind of RNN—long short-term memory (LSTM) network, have been employed for saliency detection. [36] developed a multi-task deep neural network (MT-DNN) method to predict viewport-dependent saliency on 360 videos, combining a CNN and a convolutional LSTM to extract features at a specific viewport. To eliminate distortion near the poles in equirectangular projection, [37] proposed a CNN based saliency detection approach with spherical coordinates in 360 videos. Kernels were defined on the spherical crown shape and shared across all patches on the sphere. Orthogonally, a saliency prediction network using the cube map format for video frames and optical flows was proposed in [38] for 360 videos, where a convolutional LSTM refined the rough estimates. The limitation of the above approaches is that they all detect objects of general interest but cannot track a specific user's unique observation behaviors.

B. Head Movement Prediction

The most typical head movement prediction method is the linear regression (LR) model, where the viewer's future FoV is extrapolated from motion information in the user's past observation history. To improve prediction accuracy, a viewport probabilistic model was proposed to compute a Gaussian probability distribution of LR prediction error [3]. Similarly, [4] trained an LR model but assigned larger weights to more recent samples. Rather than using all samples in a historical window, [9] considered the viewport speed in a specific measurement interval, and then predicted a future viewport in a short time horizon. Unfortunately, all models above suffer from low prediction accuracy as RTT increases. To address this problem, a K-Nearest-Neighbors (KNN) scheme was developed in [25], where the nearest K fixations were chosen to represent cross-users' behaviors. By integrating these K fixation points in LR results, view probability of each tile was computed. However, KNN clustering strongly assumed that viewers have the same areas/objects of interest. More generally, only users' observation history of head movements with no additional information was considered in all the aforementioned models for prediction.

Recently, neural-network-based prediction models have been proposed. [12] employed a deep reinforcement learning network to predict a heat map of potential head movement positions offline. Then, an online model estimated a future direction among eight candidates. Considering image saliency, motion feature, and viewed orientation, a fixation prediction algorithm using LSTM network was proposed in [5]. In particular, they considered virtual viewports, future contents, and larger datasets to improve prediction performance. However, these models require huge datasets for training and have high computational complexity. Moreover, training is performed under specific environment settings and not easily generalized.

To the best of our knowledge, our work is the first in the literature to combine a 360 image saliency map, collected viewer's head movement traces, and a biological human head model into a sparse directed graph learning problem for head movement prediction, one that is easily applicable to a wide range of realistic RTTs.

III. System Overview

We first overview our proposed tile-based interactive 360 video streaming system, illustrated in Fig. 3. We focus our discussion on the simple case when the Group of Pictures (GOP) size for video encoding of each tile is a single frame, meaning that an observer can request a different tile set every frame.[3] Table I summarizes the main notations used.

The 360° observation sphere is discretized uniformly into K unique angles. At the server, a captured 360 video is first mapped into a 2D rectangular plane via a projection model such as the equirectangular projection (component A). The rectangular frame is then encoded into M non-overlapping equal-size tiles covering the 360° sphere, where $M \ll K$. See Fig. 4 for an illustration. Each tile is coded into two versions: high and low quality. This is called *tile optimization* (component B), where the number of tiles M and the encoding qualities of the two versions for each tile are optimized.

When the most recent recorded view angle k of an observer arrives at the server, it transmits a set of tile versions—at most one version for each tile—corresponding to angle k. This is called *streaming optimization* (component C). At the client, after receiving the tile combination, the user reconstructs a FoV to display on the HMD (component E).

To forecast a viewer's FoV one RTT later, a head movement prediction model, parameterized via sparse directed graph learning, estimates the future viewing probability of each view angle, given the last reported view angle (component D). Specifically, we employ a discrete Markov model for prediction, expressed in a $K \times K$ stochastic probability transition matrix. Three sources of relevant information—a 360 image saliency map, collected viewers' head movement traces, and a biological head rotation model—are aggregated to learn this matrix. As the core component of our streaming system, the head movement prediction model is vital for the optimization of tile encoding (component B) as well as tile selection during streaming (component C), such that the expected distortion is minimized while satisfying a BW

[3]One can easily generalize to larger GOPs, similarly done in [7].

TABLE I
MAIN NOTATIONS

Symbol	**Definition**
$K, 1 \leq k \leq K$	number of angles in $360°$ observation sphere and specific angle, respectively
$M, 1 \leq m \leq M$	number of non-overlapping tiles and specific tile, respectively
H	number of view angles covered by each tile, *i.e.* K/M
T_o	operational RTT
N	number of observations in $\mathcal{X}$
N_k	number of occurrences of angle k in set $\mathcal{X}$
N_{kl}	number of occurrences of switching from angle k to angle l in $\mathcal{X}$
r	encoding rate of a tile
C	bandwidth capacity
$\mathcal{S}_m$	set of view angles covered by tile m
$\mathcal{X}$	training set of observed angle switches in traces
$\boldsymbol{\theta} = \{\{q_k\}, \{p_{kl}\}\}$	parameter set for a Markov model
$\mathcal{N}(k)$	neighborhood set of k
$\mathbf{P}, p_{kl}$	$K \times K$ view transition probability matrix and transition probability from angle k to l in $\mathbf{P}$, respectively
$\mathbf{q}, q_k$	length-K stationary probability vector and stationary probability of angle k, respectively
$\hat{\mathbf{q}}$	normalized saliency vector
$\tilde{\mathbf{q}}$	eigenvector of $\mathbf{P}$ corresponding to eigenvalue 1
$\mathbf{D}$	$3 \times M$ distortion matrix whose m-th column specifies the possible distortions $[d_m^H, d_m^L, d_{\max}]^\top$ for tile m.
$\mathbf{R}$	rate matrix corresponding to distortion matrix $\mathbf{D}$ with $R_{i,j} = r(D_{i,j}, H)$
$\mathbf{F}_k$	$M \times 3$ binary matrix whose m-th row specifies the chosen version of tile m given head angle k at time t
$\boldsymbol{\Psi}$	$K \times M$ binary matrix where $\Psi_{k,m} = 1$ if $k \in \mathcal{S}_m$ and all others being 0
$\mathbf{d}_{\mathbf{F}_k}, d_{F_k,k}$	length-K distortion vector and distortion of the tile covering angle k in $\mathbf{d}_{\mathbf{F}_k}$, respectively
$\mathbf{C}_a$	$K \times K$ binary convolution matrix used to account for a angles viewable inside a FoV

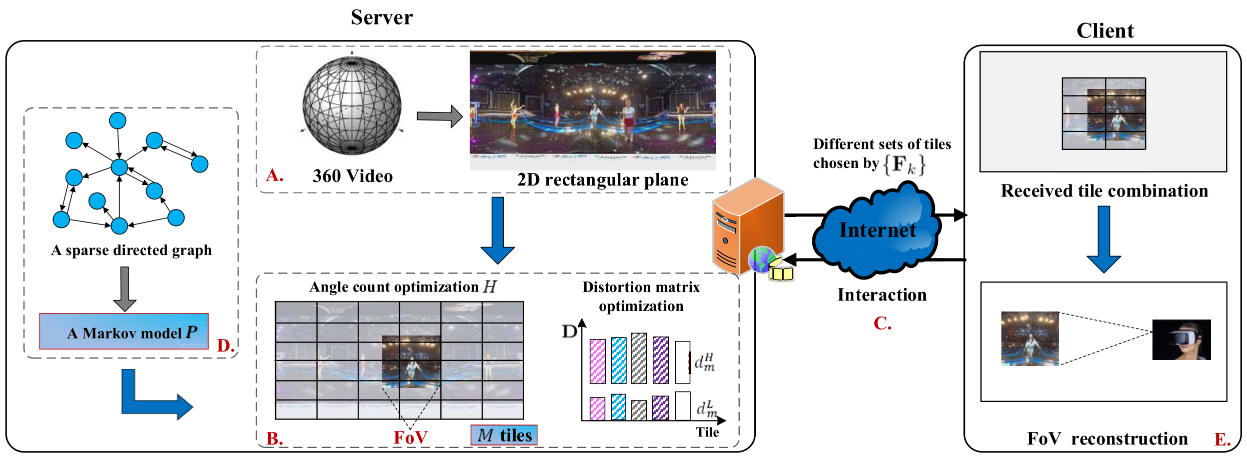


Fig. 3. The proposed tile-based interactive 360 video streaming system. At the server, component A is the projection from 3D space to 2D plane. Component B is tile optimization, where angle count per tile H (*i.e.* K/M) and distortion matrix $\mathbf{D}$ for M per-encoded tiles are optimized for storage. Component C is streaming optimization, in which the server transmits a set of tiles, whose versions are chosen by a binary matrix $\mathbf{F}_k$ corresponding to observer's last reported view angle k. Component D is head movement prediction model, parameterized via sparse directed graph learning, that computes a transition matrix $\mathbf{P}$ predicting a future viewing probability for each view angle. At the client, after receiving a tile combination, the user reconstructs a FoV for display on HMD in component E. Components D is our key contribution in this paper.

constraint. We next describe the two sets of optimization variables in our system.

A. View Transition Matrix for Head Movement Prediction

We first define *operational RTT* as follows. Denote by T_c the RTT in seconds between server and client, and by Δ the time interval between two consecutive video frames' playback times. We then define the operational RTT, $T_o \in \mathbb{Z}^+$, as the integer number of inter-frame intervals, *i.e.*, $T_o = \lceil T_c/\Delta \rceil$.

Denote by $\mathbf{P} \in \mathbb{R}^{K\times K}$ a *view transition probability matrix* that models a user's view angle switching tendencies for a target video sequence. Specifically, $\mathbf{P}$ describes a Markov model with K discrete states. Thus, an observer starting with view angle k has view probability distribution $\mathbf{1}_k\mathbf{P}^{T_o}$ one RTT later, where $\mathbf{1}_k$ is a length-K canonical row vector with the k-entry being 1 and all other entries being 0. As a probability transition matrix, $\mathbf{P}$ is row-stochastic (*i.e.*, $\sum_j P_{ij} = 1, \forall i$). Assuming $\mathbf{P}$ is irreducible,[4] $\mathbf{P}$ has a *stationary probability vector* (left eigenvector corresponding to eigenvalue 1) $\mathbf{q}$, *i.e.*, $\mathbf{q}\,\mathbf{P} = \mathbf{q}$, due to the Perron-Frobenius Theorem[5] [40]. We discuss the optimization of $\mathbf{P}$ and $\mathbf{q}$ in Section IV.

[4]A graph is irreducible if any node is reachable from any other node [39].

[5]$\mathbf{P}$ is corresponding to an ergodic (recurrent and aperiodic) finite-state Markov chain, and $\mathbf{q}$ thus has strictly positive entries, *i.e.*, $q_i > 0, \forall i$.

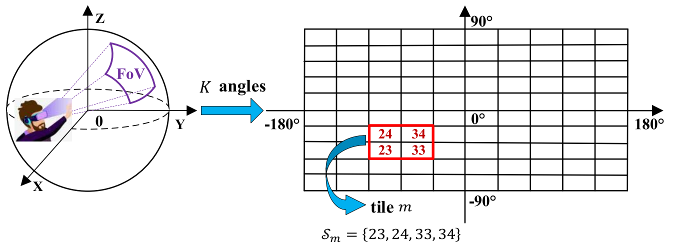


Fig. 4. Example of uniform discretization of the 360° sphere into K angles (small black boxes) and the tile m (red box) covering view angle set $\mathcal{S}_m$.

B. Tile Count and Distortions in Tile Optimization

Given that M tiles cover a total of K discrete view angles, on average, the number of view angles covered by each tile is $H = \frac{K}{M}$. Denote by $\mathcal{S}_m$ the set of view angles covered by tile m. See Fig. 4 for an illustration. Because the non-overlapping tiles together cover the entire discrete view angle set $\mathcal{S} = \{1, \ldots, K\}$, we can write

$$\bigcup_{m=1}^{M} \mathcal{S}_m = \mathcal{S}, \quad \text{s.t. } \mathcal{S}_m \cap \mathcal{S}_n = \emptyset, \quad m, n \in \{1, \ldots, M\} \text{ and } m \neq n. \quad (1)$$

Given that each tile m is pre-encoded into two versions stored at the server, we denote their respective distortions by d_m^H and d_m^L corresponding to high and low quality encoding. We assume that $0 \leq d_m^H < d_m^L < d_{\max}$, where $d_{\max}$ is the resulting distortion of this spatial region if no tile version is transmitted. A smaller distortion means a higher visual quality but induces a larger encoding rate, which may not be suitable for BW-constrained streaming. We discuss the optimization of H and distortions d_m^H and d_m^L for each tile m in Section V.

IV. Sparse Directed Graph Learning

We formulate a *maximum a posteriori* (MAP) optimization problem to estimate the view transition matrix $\mathbf{P}$ for head movement prediction (component D in Fig. 3), using three sources of information: i) a visual saliency map, ii) viewer head movement traces, and iii) a biological model for the human head. The visual saliency map provides estimates of the stationary probability vector $\mathbf{q}$, which is the left eigenvector of $\mathbf{P}$ corresponding to eigenvalue 1. The viewer traces provide estimates of individual transition probabilities P_{ij}'s. The human head model specifies sparsity patterns in $\mathbf{P}$.

We first define a likelihood term for P_{ij}'s given viewer traces. We then define the prior terms for $\mathbf{q}$ and $\mathbf{P}$, given a 360 saliency map and sparse matrix assumptions stemming from the head model, respectively. Finally, we write a MAP formulation using the defined likelihood and prior terms. We solve the problem by alternately optimizing $\mathbf{P}$ and $\mathbf{q}$ using the Frank-Wolfe (FW) optimization method [21].

A. Likelihood Term

The data traces are collections of angle switches by different viewers from one discrete angle to another for a target video sequence. Denote by $\mathcal{X}$ the training set of observed angles in traces, and N the total number of observations in $\mathcal{X}$. N_k is the occurrence count for angle k in set $\mathcal{X}$ where $1 \leq k \leq K$. Similarly, N_{kl} is the occurrence count for view-switch from angle k to angle l in $\mathcal{X}$, where $1 \leq k, l \leq K$.

From these definitions, N_k and N_{kl} must satisfy:

$$\sum_{k=1}^{K} N_k = N, \quad \sum_{l=1}^{K} N_{kl} = N_k. \quad (2)$$

q_k in row vector $\mathbf{q} \in \mathbb{R}^K$ is the stationary probability of view angle k. Similarly, p_{kl} in matrix $\mathbf{P} \in \mathbb{R}^{K \times K}$ is the transition probability from angle k to l. Both can be empirically estimated from data as

$$q_k = \frac{N_k}{N}, \quad p_{kl} = \frac{N_{kl}}{N_k}. \quad (3)$$

Defining observation set as $\mathcal{X} = \{\{N_k\}, \{N_{kl}\}\}$, we aim to find suitable parameters[6] $\boldsymbol{\theta} = \{\{q_k\}, \{p_{kl}\}\}$ for a Markov model that best describes $\mathcal{X}$. We first write the posterior probability of $\boldsymbol{\theta}$ given $\mathcal{X}$ via Bayes' rule:

$$\mathrm{P}(\boldsymbol{\theta}|\mathcal{X}) = \frac{\mathrm{P}(\mathcal{X}|\boldsymbol{\theta})\mathrm{P}(\boldsymbol{\theta})}{\mathrm{P}(\mathcal{X})} \quad (4)$$

where $\mathrm{P}(\mathcal{X}|\boldsymbol{\theta})$ is the likelihood of observing $\mathcal{X}$ given parameters $\boldsymbol{\theta}$, and $\mathrm{P}(\boldsymbol{\theta})$ is the prior which describes *a priori* knowledge about $\boldsymbol{\theta}$ without data traces. Given previous definitions, we define the likelihood term $\mathrm{P}(\mathcal{X}|\boldsymbol{\theta})$, similarly done in a previous Markovian parameter estimation [41], as

$$\begin{aligned}\mathrm{P}(\mathcal{X}|\boldsymbol{\theta}) &= \mathrm{P}(\{N_k\} \,|\, \{q_k\}) \cdot \mathrm{P}(\{N_{kl}\} \,|\, \{p_{kl}\}) \\ &= \prod_{k=1}^{K} q_k^{N_k} \cdot \prod_{k=1}^{K}\prod_{l=1}^{K} (p_{kl} + U(p_{kl} = 0))^{N_{kl}} \\ &= \prod_{k=1}^{K} \left(q_k^{N_k} \prod_{l=1}^{K} (p_{kl} + U(p_{kl} = 0))^{N_{kl}} \right) \quad (5)\end{aligned}$$

where $U(\mathbf{c})$ is a *step function* that evaluates 1 if clause $\mathbf{c}$ is true and 0 otherwise. Thus, if $p_{kl} > 0$, then $p_{kl} + U(p_{kl} = 0) = p_{kl}$, and if $p_{kl} = 0$, then $p_{kl} + U(p_{kl} = 0) = 1$.

B. Prior Term

We define a signal prior for $\mathbf{q}$, given a saliency map of the 360 image content as computed in [16], as follows:

$$\mathrm{P}(\mathbf{q}) = \prod_{k=1}^{K} \exp\left(\frac{-(q_k - \hat{q}_k)^2}{\sigma_q^2} \right) \quad (6)$$

where $\hat{q}_k$ is the normalized saliency of angle k (*i.e.*, $\sum_k \hat{q}_k = 1$), and σ_q is a parameter.

Further, we argue for a *sparse matrix assumption* for $\mathbf{P}$ based on human biological modeling [15], [42] as follows. A typical human observer has only a few points of interest in video contents, thus drawing head movements towards these points only. Consider the example in Fig. 5, where popular

[6]Though $\{q_k\}$ are functions of $\{p_{kl}\}$, we write $\{q_k\}$ separately here to be consistent with the alternating optimization for $\{q_k\}$ and $\{p_{kl}\}$ discussed later. Similarly, we write observations $\{N_k\}$ separately from $\{N_{kl}\}$, though $\{N_k\}$ are functions of $\{N_{kl}\}$.

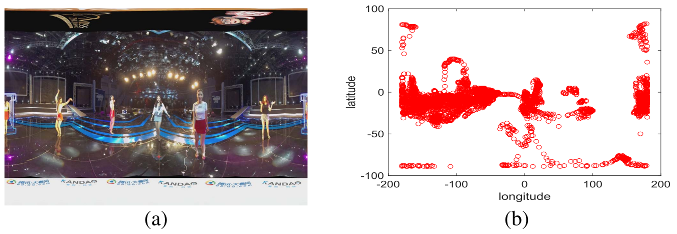


Fig. 5. (a) An 2D rectangular frame mapped from a captured 360 video frame via the popular equirectangular projection, and (b) the corresponding collections of discrete angles observed by a few viewers in data traces.

viewing angles are concentrated in the equatorial regions. We thus define a prior for $\mathbf{P}$ to promote a small ℓ_0 norm:

$$\mathrm{P}(\mathbf{P}) = \exp\left(\frac{-\parallel \mathbf{P} \parallel_0}{\sigma_p^2}\right) \tag{7}$$

where σ_p is a parameter.

In addition, given the physical limitations of a human head, a typical user is unlikely to drastically switch view angles in one discrete time unit. We thus define a neighborhood $\mathcal{N}(k)$ for each view angle k, so that for each angle $l \notin \mathcal{N}(k)$, $p_{k,l} = 0$ [43]. The notion of view neighborhood significantly reduces the number of non-zero transition probabilities p_{kl}'s in $\mathbf{P}$, thus reducing the risk of overfitting.

We summarize our prior term for parameters $\boldsymbol{\theta}$ as follows:

$$\begin{aligned}\mathrm{P}(\theta) &= \mathrm{P}(\mathbf{q})\mathrm{P}(\mathbf{P})\\ &= \prod_{k=1}^{K} \exp\left(\frac{-(q_k-\hat{q}_k)^2}{\sigma_q^2}\right)\exp\left(\frac{-\parallel \mathbf{P} \parallel_0}{\sigma_p^2}\right).\end{aligned} \tag{8}$$

C. MAP Estimation

We write an optimization for $\boldsymbol{\theta}$ via a MAP formulation:

$$\theta^* = \arg\max_{\boldsymbol{\theta}} \mathrm{P}(\mathcal{X}|\boldsymbol{\theta})\mathrm{P}(\boldsymbol{\theta}) \tag{9a}$$

$$= \arg\max_{\{\{q_k\},\{p_{kl}\}\}} \prod_{k=1}^{K}\left(q_k^{N_k}\prod_{l=1}^{K}(p_{kl}+U(p_{kl}=0))^{N_{kl}}\right)$$
$$\prod_{k=1}^{K}\exp\left(\frac{-(q_k-\hat{q}_k)^2}{\sigma_q^2}\right)\exp\left(\frac{-\parallel \mathbf{P}\parallel_0}{\sigma_p^2}\right) \tag{9b}$$

$$\text{s.t.} \sum_{k=1}^{K} q_k p_{kl} = q_l, \quad \forall l \tag{9c}$$

$$\sum_{k=1}^{K} q_k = 1, \qquad \sum_{l=1}^{K} p_{kl} = 1, \quad \forall k \tag{9d}$$

$$q_k \geq 0, \quad \forall k \qquad p_{kl} \geq 0, \quad \forall k, l \in \mathcal{N}(k) \tag{9e}$$

$$p_{kl} = 0, \quad \forall k, l \notin \mathcal{N}(k) \tag{9f}$$

where constraint (9c) can be written in matrix form as $\mathbf{q}\,\mathbf{P} = \mathbf{q}$. We omit the step function $U(\mathbf{c})$ in the sequel for notation simplicity.

To ease optimization, we minimize the negative log of (9b):

$$\theta^* = \arg\min_{\{\{q_k\},\{p_{kl}\}\}} -\sum_{k=1}^{K}\left(N_k \ln q_k + \sum_{l=1}^{K} N_{kl}\ln p_{kl}\right) + \sum_{k=1}^{K}\frac{(q_k-\hat{q}_k)^2}{\sigma_q^2} + \frac{\parallel \mathbf{P}\parallel_0}{\sigma_p^2}. \tag{10}$$

Since the last term in (10) is a non-convex ℓ_0 norm, we relax it to a sparsity-promoting weighted ℓ_2 norm via *iterative reweighted least square* (IRLS) [20]:

$$\arg\min_{\{\{q_k\},\{p_{kl}\}\}} -\sum_{k=1}^{K}\left(N_k \ln q_k + \sum_{l=1}^{K} N_{kl}\ln p_{kl}\right) + \sum_{k=1}^{K}\frac{(q_k-\hat{q}_k)^2}{\sigma_q^2} + \frac{1}{\sigma_p^2}\sum_{k,l}\omega_{kl}\,(p_{kl})^2 \tag{11}$$

where weight $\omega_{kl} = \frac{1}{(\tilde{p}_{kl}^2+\varepsilon_s)}$ is computed using previous estimate $\tilde{p}_{kl}$ to promote sparsity in $\mathbf{P}$, as (11) is computed iteratively.

While (11) is convex and differentiable, the relationship between q_k and p_{kl} in constraint (9c) is multiplicative. This makes the optimization of q_k and p_{kl} computationally difficult.

D. Optimization of Parameter $\boldsymbol{\theta}$

1) Optimizing $\mathbf{P}$*:* To minimize objective (11) while satisfying constraints in (9), we pursue an alternating strategy, where q_k's are fixed when p_{kl}'s are optimized, and vice versa. Specifically, when $\mathbf{q}$ is fixed, we optimize $\mathbf{P}$ as follows:

$$\begin{aligned}&\min_{\mathbf{P}} -\sum_{k=1}^{K}\sum_{l=1}^{K} N_{kl}\ln p_{kl} + \frac{1}{\sigma_p^2}\sum_{k=1}^{K}\sum_{l=1}^{K}\omega_{kl}\,(p_{kl})^2\\ &\text{s.t.} \sum_{k=1}^{K} q_k p_{kl} = q_l, \quad \forall l \quad \sum_{l=1}^{K} p_{kl} = 1, \quad \forall k\\ &\quad p_{kl} \geq 0, \ \forall k, l \in \mathcal{N}(k) \quad p_{kl} = 0, \ \forall k, l \notin \mathcal{N}(k).\end{aligned} \tag{12}$$

For fixed ω_{kl}, (12) has a convex and differentiable objective $f(\mathbf{P})$ with linear (in)equalities. Denote by $\mathcal{R}$ the convex set of feasible solutions $\mathbf{P}$'s that satisfy the linear constraints. To solve (12), we employ the FW method [21] as follows:

1) Initialize $\mathbf{P}(1)$ and $t \leftarrow 1$.
2) Find the optimal direction $\mathbf{s}$ by solving:

$$\min_{\mathbf{s}} \ \mathbf{s}^{\top}\nabla f(\mathbf{P}(t)) \ \text{ such that } \ \mathbf{s} \in \mathcal{R}. \tag{13}$$

3) Select step size $\gamma \leftarrow \frac{2}{t+2}$.
4) Update solution $\mathbf{P}(t+1) \leftarrow \gamma\,\mathbf{s} + (1-\gamma)\mathbf{P}(t)$.
5) Update $t \leftarrow t+1$. If $\|\mathbf{P}(t) - \mathbf{P}(t-1)\|_1^2 > \epsilon$, goto step 2.

$\nabla f(\mathbf{P}(t))$ denotes the gradient of the objective function $f(\mathbf{P})$ at t, and ϵ is a *convergence threshold* to signal solution convergence. A reasonable initial probability vector $\mathbf{p}_k(1)$ is:

$$p_{kl}(1) = \begin{cases}\frac{1}{|\mathcal{N}(k)|} & \text{if } \ l \in \mathcal{N}(k)\\ 0 & \text{otherwise}\end{cases}. \tag{14}$$

Because the constraints that define convex set $\mathbf{P}(t) \in \mathcal{R}$ are linear, (13) is a *linear program* (LP), which can be solved efficiently using a standard LP solver such as Simplex or the interior point method [44].

Note that each execution of the FW method terminates with a converged solution $\mathbf{P}$ for given weights ω_{kl}'s. To promote sparsity in $\mathbf{P}$, we update weights ω_{kl}'s as discussed earlier. We thus execute FW iteratively in an outer loop until $\mathbf{P}$ converges as weights ω_{kl}'s are updated.

2) Optimizing $\mathbf{q}$*:* When $\mathbf{P}$ is fixed, minimization (11) using $\mathbf{q}$ given constraints in (9) has no solution; the optimization has $K+1$ linear equality constraints (plus lower bound for each q_k) with only K variables for $\mathbf{q}$. Denote by $\tilde{\mathbf{q}}$ the eigenvector of $\mathbf{P}$ corresponding to eigenvalue 1. We compute the following optimization instead:

$$\begin{aligned} &\min_{\mathbf{q}} -\sum_{k=1}^{K} N_k \ln q_k + \sum_{k=1}^{K} \frac{(q_k - \hat{q}_k)^2}{\sigma_q^2} \\ &\text{s.t. } \max\{0, \tilde{q}_k - \delta\} \le q_k \le \max\{0, \tilde{q}_k + \delta\}, \quad \forall k \\ &\qquad \sum_{k=1}^{K} q_k = 1 \end{aligned} \tag{15}$$

where $\delta \ge 0$ is a parameter. Given $\tilde{\mathbf{q}}$ and δ, we employ a similar FW procedure to solve (15)—another optimization problem with convex objective and linear constraints. As we iterate between optimization of $\mathbf{P}$ and $\mathbf{q}$, δ is decreased incrementally, so that at termination the solution to (15) is $\tilde{\mathbf{q}}$ when $\delta = 0$. Specifically, each execution of the FW method terminates with a converged solution $\mathbf{q}$ for a given δ. We set $\delta = 1$ initially. To satisfy the constraint (9c), we update δ by $\delta(t) = \tau\delta(t-1)$ at iteration t, where τ is a multiplicative factor that controls the speed of decreasing δ towards 0 and $0 < \tau < 1$. As δ is updated, FW is executed iteratively in an outer loop until $\mathbf{q}$ converges. Note that our optimization is robust to various τ's, and we empirically determine $\tau = 0.04$ that has fast decrease speed in the experiments.

3) Complexity Reduction: Our proposed method achieves low complexity in two ways. First, unlike first-order descent methods like proximal gradient (PG) [22], FW is projection-free and hence does not require projection into the feasible region $\mathcal{R}$. Second, FW iterations after initialization can benefit from *warm start* [24]: since LP in (13) is typically solved in an iterative manner, initializing $\mathbf{s}(1)$ as the computed $\mathbf{s}$ from previous LP can reduce the number of iterations until an optimal solution is located.

In particular, suppose that the Simplex algorithm is used to solve LP in (13), where the feasible region is the intersection of multiple half-spaces, resulting in a *polytope*. Each vertex of this polytope is known as a *basic feasible solution* (bfs). Simplex moves from one polytope vertex (one bfs) to a neighboring one (another bfs), until it reaches an optimal vertex (bfs). Because feasible space $\mathcal{R}$ remains the same in different FW iterations, solution $\mathbf{s}$ from the previous LP remains a bfs in the new LP. Hence, by employing the previous solution $\mathbf{s}$ as an initial bfs for the next LP, our warm start strategy can significantly reduce the number of iterations until the optimal bfs is found. We demonstrate the speedup of our algorithm thanks to warm start in Section VI.

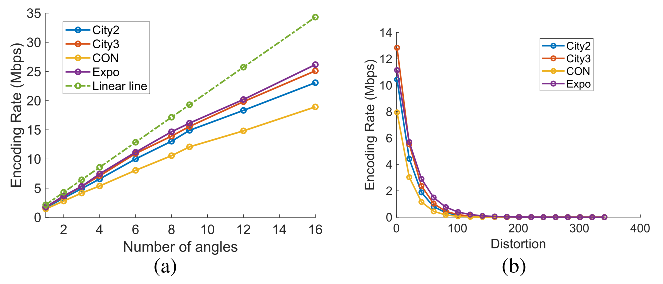


Fig. 6. (a) Encoding rate r vs angle count per tile H when QP = 30, and (b) encoding rate r vs distortion d for four 4K VR sequences.

V. Tile Optimization

To demonstrate the usefulness of our head prediction model for practical systems, we now use the estimated view transition matrix $\mathbf{P}$ to optimize a tile streaming system (component B and C in Fig. 3). There exists a basic tradeoff when optimizing tile size for 360 video interactive streaming. Larger tiles can be more efficiently encoded thanks to intra-prediction in modern image/video codecs [45], [46]. On the other hand, because of their coarser granularity, larger tiles make it harder to select a tile combination to exactly cover a predicted FoV, resulting in wasted bandwidth to transmit spatial areas outside FoV that will not be observed. To find the best tile size and distortions per tile, we formulate an optimization given $\mathbf{P}$.

We first define encoding rate used in our optimization framework. We then formulate two problems: i) a simpler streaming optimization problem given pre-encoded tiles, and ii) a more complex joint tile / streaming optimization problem.

A. Encoding Rate

As shown in Fig. 6(a), we observe empirically that encoding rate r of a tile grows *slower than linear* as the tile size (proportional to angle count per tile H) increases. This can be explained by the efficiency of intra-coding exploiting spatial redundancy, so that a tile size twice as large would not lead to twice the encoding rate. We characterize this rate behavior as a function of H as follows:

$$r(H) = b\, H^{\alpha} \tag{16}$$

where $0 < \alpha < 1$ specifies a slower-than-linear growth rate. We empirically determine $\alpha = 0.89$ for the most typical QP value of 30 used in our 360 video streaming system.

To specify the rate-distortion behavior, when H is fixed, we assume that rate r is a Laplacian function of distortion d as done in previous image/video compression works [7], [47], but we upper-bound maximum distortion at $d_{\max}$. Combining with (16), we define $r(d, H)$ as follows:

$$r(d, H) = U(d_{\max} - d) \exp\left(-\frac{|d|}{\sigma^2}\right) H^{\alpha} \tag{17}$$

where $U(x)$ is a step function; *i.e.*, $U(x) = 0$ for $x \le 0$. σ is a parameter chosen for a given 360 video content. See examples of $r(d)$ functions for different video sequences in Fig. 6 (b).

B. Streaming Optimization Problem Formulation

To facilitate understanding, we first formulate a streaming optimization problem to decide which tile versions to transmit given the pre-encoded tile versions for all M tiles stored at the server (component C in Fig. 3).

1) Streaming Decision: Denote by $\mathbf{D}$ a $3\times M$ matrix whose m-th column specifies the possible distortions $[d_m^H, d_m^L, d_{\max}]^\top$ for tile m. Denote by k an observer's head angle at time t. Next, denote by $\mathbf{F}_k$ a binary $M \times 3$ matrix whose m-th row specifies the chosen version of tile m to transmit given observer angle k. For example, m-th row [1 0 0] means that the high-quality version of tile m is chosen. Then $\text{diag}(\mathbf{F}_k\mathbf{D})$ is a length-M vector of distortions for the M chosen tile versions.

We next define a $K \times M$ binary matrix $\boldsymbol{\Psi}(\{\mathcal{S}_m\}) \in \{0,1\}^{K\times M}$ that maps M tiles to K view angles:

$$\Psi_{k,m} = \begin{cases} 1, & \text{if } k \in \mathcal{S}_m \\ 0, & \text{otherwise} \end{cases}. \tag{18}$$

Finally, we define a length-K distortion vector $\mathbf{d}_{\mathbf{F}_k}$ that specifies the distortion at each angle, computed as

$$\mathbf{d}_{\mathbf{F}_k} = \boldsymbol{\Psi}(\{\mathcal{S}_m\})\,\text{diag}(\mathbf{F}_k\mathbf{D}). \tag{19}$$

It means that the i-th entry $d_{\mathbf{F}_k,i}$ (distortion corresponding to angle i) is the distortion of the tile covering angle i.

2) Expected Distortion: Because an observer at view angle k sees a FoV centered at k, we must tabulate distortions for all angles in the FoV. Denote by $\mathbf{C}_a$ a binary $K \times K$ convolution matrix used to account for a angles viewable inside a FoV; $\mathbf{1}_k\mathbf{C}_a$ is thus equivalent to convolving an impulse at view angle k with a kernel of size a, resulting in 1 at angles inside the FoV centered at k and 0 elsewhere. For example, convolving a length-4 1D impulse $\mathbf{1}_2$ with $\mathbf{C}_3$ results in:

$$\mathbf{1}_2\mathbf{C}_3 = [0\ 1\ 0\ 0]\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} = [1\ 1\ 1\ 0]. \tag{20}$$

Assuming observer's most recent reported view angle is k, the probability distribution of the observer's view angle one RTT later is $\mathbf{1}_k\mathbf{P}^{T_o}$. Thus, the expected distortion at the client one RTT later is $\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\mathbf{d}_{\mathbf{F}_k}$. Consider all possible reported view angles k's with stationary probabilities q_k's, the expected distortion $D(\{\mathbf{F}_k\})$ for the set of decision matrices $\{\mathbf{F}_k\}$ is

$$D(\{\mathbf{F}_k\}) = \sum_{i=1}^{K} q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\mathbf{d}_{\mathbf{F}_k}. \tag{21}$$

3) Transmission Constraint: Denote by $\mathbf{R}$ a $3 \times M$ rate matrix corresponding to distortion $\mathbf{D}$, where $R_{i,j} = r(D_{i,j}, H) = r(D_{i,j}, \frac{K}{M})$. We can write the transmission constraint with BW capacity C as follows:

$$\sum_{k=1}^{K} q_k\,\text{tr}(\mathbf{F}_k\mathbf{R}) \leq C. \tag{22}$$

4) Problem Formulation: Putting the objective and constraint together, we formulate the streaming optimization problem as:

$$\begin{aligned} \min_{\{\mathbf{F}_k\}} & \sum_{k=1}^{K} q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\mathbf{d}_{\mathbf{F}_k} \\ \text{s.t.} & \sum_{k=1}^{K} q_k\,\text{tr}(\mathbf{F}_k\mathbf{R}) \leq C \end{aligned} \tag{23}$$

where $\mathbf{d}_{\mathbf{F}_k}$ is defined in (19).

C. Joint Optimization Problem Formulation

Our ultimate objective for tile optimization is to jointly optimize angle count per tile H which determines the coverage sets $\{\mathcal{S}_m\}$, distortion matrix $\mathbf{D}$ of pre-encoded M tiles, and streaming decision matrices $\{\mathbf{F}_k\}$ for different view angle k, such that the expected distortion experienced by the users is minimized while satisfying the BW constraint. We now formulate a tile streaming optimization problem as:

$$\begin{aligned} \min_{H,\mathbf{D},\{\mathbf{F}_k\}} & \sum_{k=1}^{K} q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\mathbf{d}_{\mathbf{F}_k} \\ \text{s.t.} & \sum_{k=1}^{K} q_k\,\text{tr}(\mathbf{F}_k\mathbf{R}) \leq C. \end{aligned} \tag{24}$$

Instead of the constrained problem, we write its corresponding unconstrained Lagrangian relaxed version as:

$$\min_{H,\mathbf{D},\{\mathbf{F}_k\}} \sum_{k=1}^{K} q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\mathbf{d}_{\mathbf{F}_k} + \mu\sum_{k=1}^{K} q_k\,\text{tr}(\mathbf{F}_k\mathbf{R}) \tag{25}$$

where μ is a Lagrangian multiplier chosen so that the transmission constraint is satisfied. We address the optimization (25) by alternately optimizing $\mathbf{F}_k$'s, $\mathbf{D}$ and H until convergence.

D. Optimization of $\mathbf{F}_k$, $\mathbf{D}$ *and* H

1) Optimizing $\mathbf{F}_k$*:* We first discuss how to compute $\mathbf{F}_k$'s when $\mathbf{D}$ and H are fixed. For this sub-problem, (25) can be optimized for each k and m separately:

$$\min_{\mathbf{F}_{k,m}} \quad q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\Psi_m\mathbf{F}_{k,m}\mathbf{D}_m + \mu\, q_k\,\mathbf{F}_{k,m}\mathbf{R}_m \tag{26}$$

where Ψ_m is the m-th column of matrix Ψ, and $q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\Psi_m$ is the probability that tile m is viewed one RTT later given the observer was in angle k. $\mathbf{F}_{k,m}$ is the m-th row of $\mathbf{F}_k$, $\mathbf{D}_m$ and $\mathbf{R}_m$ are the m-th columns of $\mathbf{D}$ and $\mathbf{R}$, respectively. Given $\mathbf{D}_m$ has only three possible values, we can compute (26) easily by trying each of the three possibilities.

2) Optimizing $\mathbf{D}$*:* We now consider the sub-problem of optimizing $\mathbf{D}$ given H and $\mathbf{F}_k$ are fixed. For each tile m, we simplify (25) to

$$\min_{d_m^L, d_m^H} \sum_{k=1}^{K} q_k\mathbf{1}_k\mathbf{P}^{T_o}\mathbf{C}_a\Psi_m\mathbf{F}_{k,m}\mathbf{D}_m + \mu\sum_{k=1}^{K} q_k\,\mathbf{F}_{k,m}\mathbf{R}_m. \tag{27}$$

Given $R_{i,j} = r(D_{i,j}, \frac{K}{M})$, we see that $\mathbf{F}_{k,m}\mathbf{R}_m = r(\mathbf{F}_{k,m}\mathbf{D}_m, \frac{K}{M})$, and hence we rewrite (27) as

$$\min_{d_m^L, d_m^H} \sum_{k=1}^{K} q_k \mathbf{1}_k \mathbf{P}^{T_o} \mathbf{C}_a \Psi_m \mathbf{F}_{k,m} \mathbf{D}_m + \mu \sum_{k=1}^{K} q_k\, r(\mathbf{F}_{k,m}\mathbf{D}_m, \frac{K}{M}). \tag{28}$$

We take the derivative of (28) with respect to a particular $d_m^i \in \{d_m^L, d_m^H\}$ and set it to 0:

$$\sum_{k|\mathbf{F}_{k,m}\mathbf{D}_m = d_m^i} \left(q_k \mathbf{1}_k \mathbf{P}^{T_o} \mathbf{C}_a \Psi_m + \mu q_k \frac{\partial\, r(d_m^i, \frac{K}{M})}{\partial\, d_m^i} \right) = 0. \tag{29}$$

Then the optimal distortion can be calculated as

$$d_m^{i\,*} = -\sigma^2 \left(\ln\left(\frac{\gamma \sigma^2}{\rho}\right) - \alpha \ln \frac{K}{M} \right) \tag{30}$$

where $\gamma = \sum_{k|\mathbf{F}_{k,m}\mathbf{D}_m = d_m^i} q_k \mathbf{1}_k \mathbf{P}^{T_o} \mathbf{C}_a \Psi_m$ and $\rho = \mu \sum_{k|\mathbf{F}_{k,m}\mathbf{D}_m = d_m^i} q_k$.

3) Optimizing H: Fixing $\mathbf{F}_k$'s and $\mathbf{D}$, since the gradient of objective (25) with respect to H is not easy to compute, we employ *Golden-section* (GS) search [48] to optimize H (and by extension $\{\mathcal{S}_m\}$). Without computing any gradients, GS requires only an assumption that the objective $f(H)$ is unimodal with a single extremum inside a specified interval.

GS is an iterative search procedure that always maintains a progressively narrowing interval $[H_l, H_r]$ containing the target minimum $H_{\min}$. Specifically, at any given time, GS retains three previously computed objective values $f(H_l), f(H_1), f(H_r)$ at three points $H_l < H_1 < H_r$, where the loop invariant is $f(H_1) < \min(f(H_l), f(H_r))$. At iteration t, GS tests one additional point $f(H_2)$, say $H_1 < H_2 < H_r$. If $f(H_1) < f(H_2)$, then interval is narrowed to $[H_l, H_2]$ and GS retains three points $H_l < H_1 < H_2$. On the other hand, if $f(H_1) > f(H_2)$, then interval is narrowed to $[H_1, H_r]$ instead and GS retains three points $H_1 < H_2 < H_r$. See Fig. 7 for an illustration. The algorithm terminates when the retained interval is sufficiently narrow.

The only remaining question is: given a starting interval $[H_l, H_r]$, how should the initial test point H_1 be chosen. To minimize the worst case of having too large a resulting narrowed interval between the above two cases, it is determined that the ratio of $|H_r - H_1|$ to $|H_1 - H_l|$ should be the Golden ratio of $\varphi = 1.618$, hence the name GS. In our specific optimization of angle count per tile H, we set $H_l = 1$ and H_r to the number of H candidates. See [48] for GS search implementation details.

VI. Experiments

A. Model-Based Prediction Comparison

1) Experimental Setup: We conducted simulations using six 360 VR sequences at 30fps with duration around 60 seconds each, captured using a professional stereo 360 camera called "Obsidian" manufactured by Kandao Technology,[7] and the corresponding viewers' head movement traces, also provided by Kandao. `Hill` ("landscape" video) and `City1`

[7]https://www.kandaovr.com/

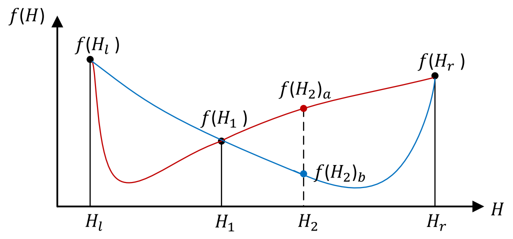


Fig. 7. The Golden-section search. The initial three points are H_l, H_1 and H_r, where $|H_r - H_1|/|H_1 - H_l| = \varphi$. If $f(H_2) = f(H_2)_a$, H_l, H_1 and H_2 are chosen for next iteration. If $f(H_2) = f(H_2)_b$, H_1, H_2 and H_r are chosen.

("streetscape" video) were 8K resolution sequences (7680 × 3840). `City2` ("streetscape" video), `City3` ("landscape" video), `CON` ("chamber concert" video) and `Expo` ("motor show" video) were 4K resolution (3840 × 1920) sequences. There were 420 and 280 traces for `Hill` and `City1`, respectively, and 50 traces for four 4K sequences, where each trace was roughly 30-second long, and the sample rate was 30 samples per second. The number of view angles K was set at 100, and FoV was assumed to be 108° × 90° [5]. RTT T_c was chosen from 0.05s, 0.3s, 1s and 2s. Each neighborhood $\mathcal{N}(k)$ had 15 angles around each angle k. We used the model for 360 images in [16] to compute a saliency map.

We first compared our head movement prediction algorithm against five representative regression models and a saliency-based approach labeled "Sal". The regression models utilize only historical samples in window $(t-m, t]$ to predict a future head position at $t + T_o$, where t is the current time instant. Three linear regression models are linear regression [3] labeled "LR", weighted linear regression [4] labeled "WLR", and heuristic [9] labeled "Heu". A more complex linear regression is ridge regression [10] labeled "RR". Support vector regression [10] labeled "SVR" is a nonlinear regression method. In particular, "LR" uses all samples in the window with the same weight, while "WLR" assigns larger weights for more recent samples. Instead of all samples, "Heu" considers one historical sample of 0.1s ago to calculate the speed of fixation point. By adding the regularization term, "RR" shrinks the coefficients, reduces model complexity and better prevents over-fitting probabilistically than simple linear regression. "SVR" uses Support Vector Machine (SVM) for regression.

As done in [3], given a predicted head position, we mapped it to a view probability distribution $g_t(T_o)$ for each instant t as follows. Assuming that the angle prediction error followed a Gaussian Distribution $e_t \sim \mathcal{N}(\mu_t, \sigma_t^2)$, $g_t(T_o)$ was computed as

$$g_t(T_o) = \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left\{ -\frac{(e_t - \mu_t)^2}{2\sigma_t^2} \right\} \tag{31}$$

where μ_t was the predicted head position, and standard deviation σ_t was learned from the dataset.

The saliency-based model is a memoryless baseline scheme, where the probability of the current angle is independent of the previous head positions. Thus, the stationary probability from the saliency map was used for each instant, *i.e.*, $g_t(T_o) = q_t$.

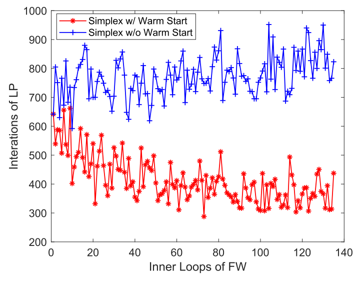


Fig. 8. Comparison of interation numbers in each inner loop of the FW algorithm with and without *warm start*.

For our proposed model, an observer starting with view angle k has view probability distribution $\mathbf{1}_k\mathbf{P}^{T_o}$ one RTT later. Thus we have $g_t(T_o) = \mathbf{1}_k\mathbf{P}^{T_o}\mathbf{1}_l^\top$, where l is the actual observer view angle one RTT later.

The aggregate probability G of each collected trace of length L was computed for different prediction models: $G = \prod_{t=1}^{L} g_t(T_o)$. To better visualize the comparisons, we took the average of the negative log as the prediction error for angles of each trace, *i.e.*, $Er = -\frac{\sum_{t=1}^{L}\ln g_t(T_o)}{L}$.

Moreover, we also considered a competing scheme exploiting cross-users' behaviors for head movement prediction, labeled "Cross-Users" [25]. K-nearest fixations representing cross-users' behaviors were integrated in LR results. To compare against "Cross-Users" and "Fixation" [5] (a learning-based comparison method to be introduced), we considered five sequences in their dataset [5]—`CERN`, `Elephants`, `London`, `Sharks` and `SnowBoarding` at 4K resolution. Each sequence had 30 head movement traces. To be consistent, we employed the same two metrics. *Accuracy* is the ratio of correct tiles to the union of predicted and ground-truth tiles. Since each tile contains a few angles in our case, the accuracy is the corresponding ratio in terms of angles covered in a FoV. *F-Score* is the harmonic mean of precision and recall. Precision and recall are ratios of correct angles to the predicted and ground-truth angles in a FoV, respectively. Since our models generate a view angle probability distribution, we computed the *expected* accuracy and F-Score.

2) Experimental Results: To show the benefit of our optimization reaped from warm start initialization, we compared the numbers of iterations in the inner loop of the FW algorithm when optimizing $\mathbf{P}$ in Section IV-D, with and without warm start. Specifically, we used the Simplex algorithm to solve each LP problem (13), where for warm start, the computed solution $\mathbf{s}$ from the previous LP was used as the initial solution for the next iteration LP. As shown in Fig. 8, as FW progressed, the LP iteration count for warm start decreased. As shown in Table II, we observe a large reduction in LP iterations when using warm start, meaning that our optimization was executed much faster.

In Fig. 9, we show the saliency map of the left view of `Hill` using saliency model [16] as an example. Using the saliency map of each sequence, the stationary probability $\hat{q}_k$ in the prior term (6) can be calculated.

TABLE II
THE TOTAL NUMBER OF INTERATIONS IN BOTH CASES

Algorithm	w/ *warm start*	w/o *warm start*
number of interations	55859	104252

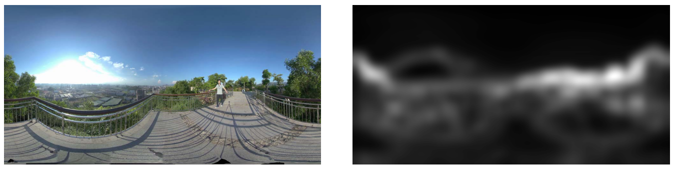


Fig. 9. (a) The 1st frame of `Hill`, (b) Saliency map.

TABLE III
AVERAGE Er OF DIFFERENT MODELS AT VARIOUS T_c'S

$T_c(s)$	Seq.	LR [3]	WLR [4]	Heu [9]	RR [10]	SVR [10]	Sal [16]	Ours
0.05	`Hill`	8.50	8.65	13.92	4.98	5.12	4.72	**0.42**
	`City1`	7.97	8.75	14.07	4.88	5.01	4.85	**0.45**
	`City2`	10.77	8.87	10.12	4.78	5.12	4.61	**0.37**
	`City3`	8.66	10.31	16.14	4.80	5.01	4.49	**0.57**
	`CON`	0.44	0.78	6.93	4.91	4.92	4.12	**0.05**
	`Expo`	4.27	4.59	8.44	4.74	4.83	4.93	**0.20**
	Average	6.77	6.99	11.60	4.85	5.00	4.62	**0.34**
0.3	`Hill`	22.89	22.77	24.45	5.09	5.22	4.66	**1.10**
	`City1`	20.35	18.22	21.85	5.01	5.13	4.84	**1.05**
	`City2`	32.55	30.46	33.19	4.96	5.17	4.70	**0.93**
	`City3`	25.41	22.60	33.32	5.20	5.37	4.41	**1.21**
	`CON`	4.11	4.82	5.08	4.93	4.96	4.10	**0.55**
	`Expo`	9.05	10.16	9.47	4.84	4.94	4.91	**0.75**
	Average	19.06	18.17	21.23	5.01	5.13	4.60	**0.93**
1	`Hill`	28.11	28.79	26.01	5.26	5.29	4.66	**1.39**
	`City1`	25.83	26.98	28.92	5.17	5.22	4.83	**1.40**
	`City2`	37.21	35.71	33.02	5.09	5.21	4.67	**1.39**
	`City3`	26.07	28.85	43.35	5.22	5.23	4.32	**1.58**
	`CON`	5.32	6.87	9.97	4.98	5.04	4.12	**0.72**
	`Expo`	22.11	25.61	23.92	5.01	5.00	4.93	**1.03**
	Average	24.11	25.47	27.53	5.12	5.17	4.59	**1.25**
2	`Hill`	43.55	41.61	43.60	5.36	5.27	4.66	**1.92**
	`City1`	42.30	38.81	41.10	5.23	5.25	4.79	**2.16**
	`City2`	32.95	39.26	53.50	5.10	5.29	4.58	**1.98**
	`City3`	78.16	59.89	58.51	5.89	5.54	4.40	**2.23**
	`CON`	6.64	15.52	17.34	5.06	5.09	4.14	**1.10**
	`Expo`	38.12	30.46	30.33	5.08	5.02	4.91	**1.45**
	Average	40.27	37.59	40.73	5.29	5.24	4.58	**1.81**

We first compared our proposed head movement prediction scheme with competitors when RTT T_c was chosen from 0.05s, 0.3s, 1s and 2s. The average prediction errors Er in terms of view angles for all traces are shown in Table III. Overall, our model achieved by far the best prediction. The prediction performances of three linear regression models "LR", "WLR" and "Heu" were highly sensitive to T_c. In particular, their prediction errors increased significantly as RTT T_c increased. Since only one previous sample was considered for the fixation speed computation instead of more samples in a historical window, "Heu" had the worst prediction.

To better visualize the relative performances, we show prediction errors of all traces for the best performing "RR" among five regression models, "Sal", and our scheme for `Hill` in Fig. 10. We observe that "RR" showed larger prediction fluctuations than "Sal" and ours when T_c =1s.

Note that when users had more extensive and frequent head motions, as shown in Fig. 11, the prediction task was more

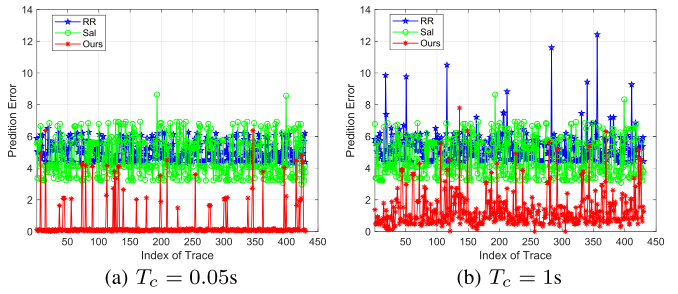


Fig. 10. Prediction error Er of each trace for `Hill` under different RTT's.

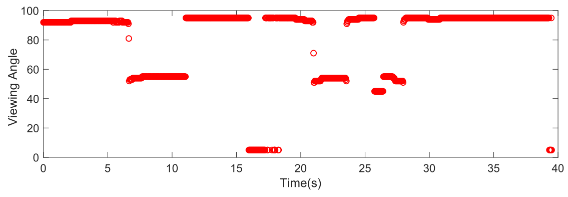


Fig. 11. Variation of the user's angles over time in one trace of `Hill`.

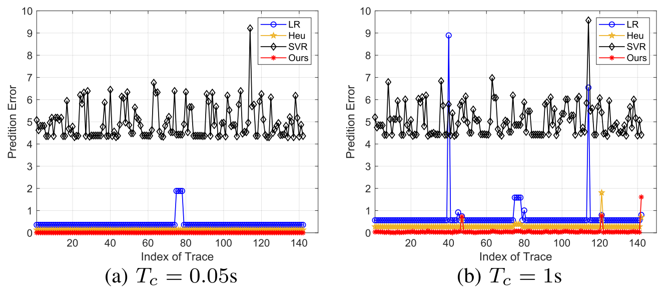


Fig. 12. Prediction error Er of each trace smaller than 10 for `Hill` under different RTT's.

challenging for three linear regression models "LR", "WLR" and "Heu". Larger prediction errors—up to 300 or more—incurred, making it difficult to visualize the differences among linear and nonlinear regression schemes. Hence, we zoomed in the figures to show prediction errors smaller than 10 in Fig. 12. We now observe that linear regression models "LR" and "Heu" generally achieved better prediction than the non-linear regression "SVR" in the case of smooth and steady movements.

Table IV reports the accuracy and F-score results for each sequence using our prediction scheme. Comparing against "Cross-Users" with different K values in KNN, Table V shows the average accuracy and F-score results of five sequences in Table IV. We observe that our method achieved the best prediction in both metrics and all K values, with accuracy and F-score increased by 9.57% and 39.17% on average, respectively.

All competing schemes above utilize either only past head movement history or saliency. In contrast, by aggregating three sources of information—a 360 image saliency map, viewers' head movement traces, and a human head model for sparse graph learning—our model maintained stable prediction performance and outperformed competitions in all cases.

TABLE IV
ACCURACY AND F-SCORE RESULTS OF OUR PREDICTION METHOD

Metric	CERN	Elephants	London	Sharks	SnowBoarding
Accuracy	82.3%	83.1%	81.6%	82.4%	82.5%
F-Score	87.5%	88.1%	87.1%	87.5%	87.5%

TABLE V
AVERAGE ACCURACY AND F-SCORE RESULTS OF DIFFERENT METHODS

Method	Cross-Users [25]				Fixation [5]	Ours
	K=0	K=2	K=5	K=10		
Accuracy	73.1%	73.0%	73.0%	72.2%	81.8%	**82.4%**
F-Score	31.0%	53.4%	54.3%	54.6%	63.1%	**87.5%**

B. Learning-Based Prediction Comparison

1) Experimental Setup: We compared against three deep learning based head movement prediction schemes, labeled "Fixation" [5], "Deep 360" [11] and "DHP" [12], respectively. Both *motion* (*e.g.*, head movement traces) and *content* (*e.g.*, saliency) are utilized in these three schemes. Specifically, "Fixation" derives the saliency map via a deep neural network and detects the motion feature from optical flow. Given saliency and motion features of frames $[t-m, t+T_0]$, as well as viewer orientations of frames $[t-m, t]$, the future orientations[8] of frames $[t+1, t+T_0]$ are predicted by a future-aware network. As mentioned, we used the same dataset and metrics in [5] as "Cross-Users" for comparison.

"Deep 360" detects the salient objects and selects the main object via RNN. Given the main object and view angles of previous t frames, the head position at frame $t+1$ is predicted by learning a regressor. "DHP" uses deep reinforcement learning (DRL) to predict head positions. Specifically, multiple DRL workflows are first run offline to generate a heat map of potential view positions at each frame, where each flow has four convolutional layers and one LSTM layer. Given the heat map and head view position at frame t, the position at frame $t+1$ is predicted via online learning.

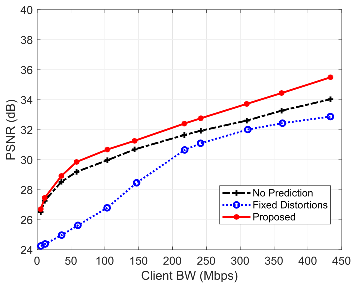
We trained our scheme using five sequences in their dataset [12]—`KingKong`, `StarryPolar`, `CMLauncher2` and `Sunset` at 4K resolution, and `Waterfall` at 8K resolution. Each sequence had 58 head movement traces, and the sampling rate was twice the video frame rate. To be consistent, we employed the same metric for prediction performance, *mean overlap* (MO), computed as

$$\mathrm{MO} = \frac{A(\mathrm{FoV}_l \cap \mathrm{FoV}_g)}{A(\mathrm{FoV}_l \cup \mathrm{FoV}_g)} \tag{32}$$

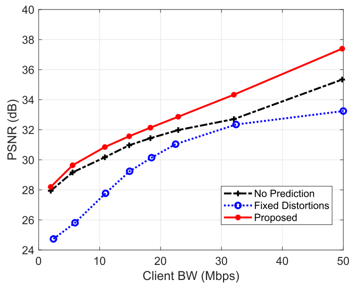
where FoV_l and FoV_g were predicted and ground-truth FoVs, respectively. A was the number of pixels included in the region. Similarly, since our model generates a view angle probability distribution given starting view angle k, we computed the *expected* MO instead, *i.e.*, $\sum_{l=1}^{K} p_{kl}\, \mathrm{MO}_l$.

2) Experimental Results: Comparing to "Fixation" in Table V, gains of 0.6% and 24.4% in terms of accuracy and F-score were achieved by our method, respectively.

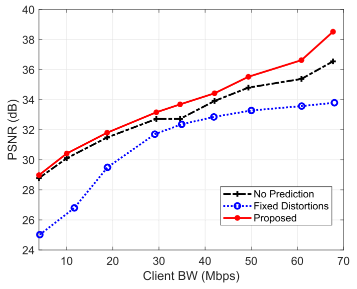
[8]The viewer orientation data includes yaw, pitch and roll, collected from HMD sensors.



(a) `Hill`



(b) `City2`



(c) `City3`

Fig. 13. Comparison of RD Performance.

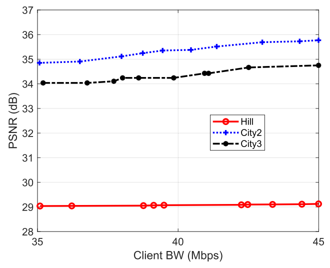


Fig. 14. Average PSNR for different client BW's at stream time assuming BW = 40Mbps during per-encoding.

TABLE VI
MO RESULTS OF DIFFERENT PREDICTION METHODS

Method	KingKong	StarryPolar	CMLauncher2	Waterfall	Sunset
Deep 360 [11]	0.34	0.32	0.21	0.08	0.46
DHP [12]	0.81	0.55	0.76	0.67	0.66
Ours	**0.97**	**0.91**	**0.93**	**0.92**	**0.89**

Table VI shows the average MO results of 58 traces for each sequence for both our scheme and others. We observe that our method achieved the best prediction with average gains of 0.64 and 0.23 with respect to "Deep 360" and "DHP", respectively.

To properly tune a large number of network parameters, these learning-based schemes require a large training dataset collected from multiple video sequences. This means that it is difficult to specialize to the statistics of a single sequence with limited collected data traces, and their prediction accuracy degraded as a result. In contrast, in our scheme, only a few parameters in the sparse transition matrix $\mathbf{P}$ required learning offline, which means we can specially tune our parameters for a target video sequence, even if the collected dataset for the one video is small. Moreover, "Deep 360" heavily relies on one salient object, which becomes challenging when more than one salient object is present. "DHP" predicts a future direction among only eight candidates, which is not sufficiently fine-grained for head position prediction. For these reasons, our scheme achieved a better prediction.

Finally, note that pure data-driven learning approaches are typically trained for particular setups, for example RTT is 1/fps in three competitors. In contrast, our scheme can be trained once and be used for different RTT's as shown in Table III, resulting in more flexibility in deployment.

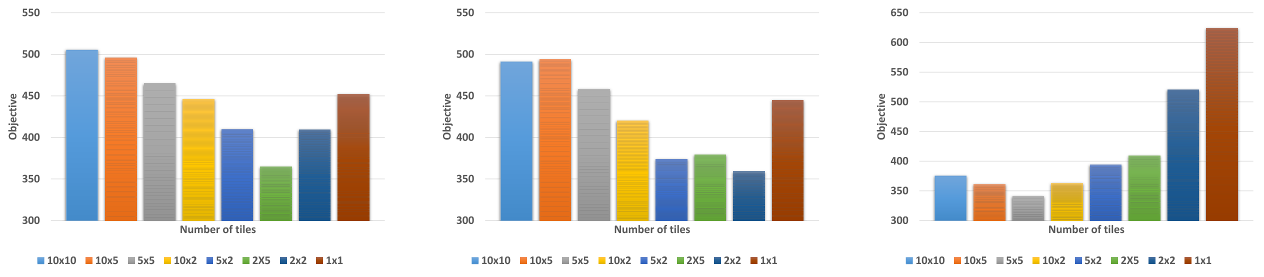
C. Tile Optimization Comparison

1) Experimental Setup: We evaluated the impact of sparse directed graph learning on our chosen tile streaming system. We first examined the performance of a scheme where tile encoding was optimized, but no head movement prediction was employed—we term this scheme "No Prediction" [26]. In details, given an expected client BW, we first optimized tile encoding at the server using the proposed scheme, where both the tile number and two distortion versions for each tile were optimized. Having the encoded tiles and given the observer's last reported view angle k, "No Prediction" transmits a combination of tiles satisfying the BW constraint. Specifically, high quality versions are allocated to the tiles covering the FoV centered at k as *first* priority. Then the immediate surrounding tiles are assigned high quality versions as well, if there is remaining BW. Finally, even further-away tiles are assigned low quality versions until the transmission budget is depleted. Given the true FoV one RTT later from the head movement traces, we computed the PSNR based on the transmitted tiles for the scheme.

We considered a second competing scheme where tile distortions were not optimized. Specifically, the same d_m^H and d_m^L chosen in an ad-hoc manner were used for all tiles, which is the common way used in the HTTP adaptive streaming framework [3], [25], [49]. We call this scheme "Fixed distortions". As done in our method, given a BW and user's last reported angle k, this scheme optimizes $\mathbf{F}_k$ to select versions of tiles to transmit based on predicted head movement as discussed in Section V-B. Based on these transmitted tiles, we computed the PSNR according to the true FoV one RTT later.

Finally, we evaluated the performance of our proposed scheme. Note that, in addition to optimization of tile distortions and transmission strategy based on head movement prediction, for a given BW, we optimized also tile number H as discussed in Section V-D.

2) RD Performance Results: We present evaluations of graph learning for our proposed tile streaming system. Using the viewers' head movement traces, we first compared our streaming system with two competitors in terms of PSNR: "No Prediction" and "Fixed distortions". For our proposal and "Fixed distortions", weight parameter μ was tuned to satisfy BW constraint at each point. Given $T_c = 1$s, we show the PSNR results of the true FoV one RTT later under different BW's for sequences `Hill`, `City2` and `City3` in Fig. 13,



(a) `Hill` (b) `City2` (c) `City3`

Fig. 15. Objective (25) comparison of variable tile numbers.

when the assumed client BW C during tile pre-encoding was also the stream time BW.

Overall, the performance of our proposed scheme substantially outperformed the two competitors, for all BW's and sequences. Specifically, using $\mathbf{P}$ for head movement prediction, our scheme outperformed "No Prediction" by up to 2.05dB when BW was 50Mbps for `City2`. Further, our scheme achieved bigger gains in general as BW increased; our scheme made better use of additional BW to optimize tile version selection. On the other hand, by optimizing tile encoding, our scheme outperformed "Fixed Distortions" by up to 4.73dB when the BW was 4Mbps for `City3`.

In Fig. 14, we show the average PSNR results for three sequences under different client BW's at stream time where the assumed BW during per-encoding was 40Mbps. We observe that the average PSNR fluctuated within a small range given that the BW fluctuated within a small range. Thus, we can conclude that the performance of our streaming system was not particularly sensitive to small changes in streaming rate.

3) Tile Optimization Results: We employed GS search to optimize H as described in Section V-D, and compared it with other H candidates in terms of objective (25). Given $T_c = 0.05$s, the objective values for different tile numbers (proportional to H) for three sequences are shown in Fig. 15. As expected, the objective was unimodal with a single extremum inside a specified interval. Thus the GS search can successfully find the unique optimal tile numbers minimizing the RD objective (25).

VII. Conclusion

To predict head movements for 360 video streaming, we formulate a sparse directed graph learning problem, where an image saliency map, collected viewers' head movement traces, and a human head rotation model are aggregated to parameterize a Markov model. We solve the problem alternately via a hybrid IRLS and FW convex optimization strategy. Experimental results show that our head movement prediction model outperformed existing approaches noticeably across a wide range of RTT, and our optimized tile-based streaming scheme outperformed competitors in rate-distortion performance.

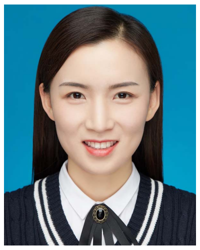
Acknowledgment

The authors thank Prof. Cheng-Hsin Hsu of National Tsing Hua University for his assistance in providing dataset of their paper [5].

References

[1] J. Zaragoza, T. Chin, Q. Tran, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1285–1298, Jul. 2014.
[2] M. Kanazawa *et al.*, "An ultrahigh-definition display using the pixel-offset method," *J. Soc. Inf. Display*, vol. 12, no. 1, pp. 93–103, 2004.
[3] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 315–323.
[4] 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.
[5] C.-L. Fan, S.-C. Yen, C.-Y. Huang, and C.-H. Hsu, "Optimizing fixation prediction using recurrent neural networks for 360° video streaming in head-mounted virtual reality," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 744–759, Jul. 2019.
[6] J. Chakareski, "Viewport-adaptive scalable multi-user virtual reality mobile-edge streaming," *IEEE Trans. Image Process.*, vol. 29, pp. 6330–6342, 2020.
[7] G. Cheung, Z. Liu, Z. Ma, and J. Z. G. Tan, "Multi-stream switching for interactive virtual reality video streaming," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 2179–2183.
[8] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, "Standardization status of immersive video coding," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 5–17, Mar. 2019.
[9] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An HTTP/2-based adaptive streaming framework for 360° virtual reality videos," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 306–314.
[10] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 99–114.
[11] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun, "Deep 360 pilot: Learning a deep agent for piloting through 360° sports videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1396–1405.
[12] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2693–2708, Nov. 2019.
[13] Z. Che, A. Borji, G. Zhai, X. Min, G. Guo, and P. Le Callet, "How is gaze influenced by image transformations? Dataset and model," *IEEE Trans. Image Process.*, vol. 29, pp. 2287–2300, 2020.
[14] X. Min, G. Zhai, J. Zhou, X.-P. Zhang, X. Yang, and X. Guan, "A multimodal saliency model for videos with high audio-visual correspondence," *IEEE Trans. Image Process.*, vol. 29, pp. 3805–3819, 2020.
[15] Y. Rai, J. Gutiérrez, and P. Le Callet, "A dataset of head and eye movements for 360 degree images," in *Proc. 8th ACM Multimedia Syst. Conf.*, Jun. 2017, pp. 205–210.
[16] J. Gutiérrez, E. David, Y. Rai, and P. L. Callet, "Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images," *Signal Process., Image Commun.*, vol. 69, pp. 35–42, Nov. 2018.
[17] J. Gutierrez, E. J. David, A. Coutrot, M. P. Da Silva, and P. L. Callet, "Introducing UN salient360! Benchmark: A platform for evaluating visual attention models for 360° contents," in *Proc. QoMEX*, Rome, Italy, May 2018, pp. 1–3.

[18] P. Lebreton and A. Raake, "GBVS360, BMS360, ProSal: Extending existing saliency prediction models from 2D to omnidirectional images," *Signal Process., Image Commun.*, vol. 69, pp. 69–78, Nov. 2018.
[19] Y. Zhu, G. Zhai, X. Min, and J. Zhou, "The prediction of saliency map for head and eye movements in 360 degree images," *IEEE Trans. Multimedia*, vol. 22, no. 9, pp. 2331–2344, Sep. 2020.
[20] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math. A, J., Courant Inst. Math. Sci.*, vol. 63, no. 1, pp. 1–38, Jan. 2010.
[21] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logistics Quart.*, vol. 3, nos. 1–2, pp. 95–110, 1956.
[22] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.
[23] M. Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 2017, pp. 427–435.
[24] E. A. Yildirim and S. J. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM J. Optim.*, vol. 12, no. 3, pp. 782–810, Jan. 2002.
[25] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "CUB360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.
[26] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using MPEG-DASH," in *Proc. 7th Int. Conf. Multimedia Syst.*, May 2016, pp. 1–3.
[27] 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.
[28] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and VR experiences," *IEEE Trans. Multimedia*, vol. 23, pp. 716–731, 2021.
[29] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "OpTile: Toward optimal tiling in 360-degree video streaming," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 708–716.
[30] X. Zhang, G. Cheung, P. Le Callet, and J. Z. G. Tan, "Sparse directed graph learning for head movement prediction in 360 video streaming," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2678–2682.
[31] L. Li, Z. Li, X. Ma, H. Yang, and H. Li, "Advanced spherical motion model and local padding for 360° video compression," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2342–2356, May 2019.
[32] C. Li, M. Xu, L. Jiang, S. Zhang, and X. Tao, "Viewport proposal CNN for 360° video quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10169–10178.
[33] Q. Zhao, L. Wan, W. Feng, J. Zhang, and T.-T. Wong, "Cube2video: Navigate between cubic panoramas in real-time," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1745–1754, Dec. 2013.
[34] F. Battisti, S. Baldoni, M. Brizzi, and M. Carli, "A feature-based approach for saliency estimation of omni-directional images," *Signal Process., Image Commun.*, vol. 69, pp. 53–59, Nov. 2018.
[35] J. Ling, K. Zhang, Y. Zhang, D. Yang, and Z. Chen, "A saliency prediction model on 360 degree images using color dictionary based sparse representation," *Signal Process., Image Commun.*, vol. 69, pp. 60–68, Nov. 2018.
[36] M. Qiao, M. Xu, Z. Wang, and A. Borji, "Viewport-dependent saliency prediction in 360° video," *IEEE Trans. Multimedia*, vol. 23, pp. 748–760, 2020.
[37] Z. Zhang, Y. Xu, J. Yu, and S. Gao, "Saliency detection in 360 videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 488–503.
[38] Y. Zhang, F. Dai, Y. Ma, H. Li, Q. Zhao, and Y. Zhang, "Saliency prediction network for 360° videos," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 27–37, Nov. 2020.
[39] M. Milgram, "Irreducible graphs," *J. Combinat. Theory, Ser. B*, vol. 12, no. 1, pp. 6–31, Feb. 1972.
[40] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
[41] A. Zheng, G. Cheung, and D. Florencio, "Context tree-based image contour coding using a geometric prior," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 574–589, Feb. 2017.
[42] M. L. Steven, *Virtual Reality*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[43] N. C. Nilsson *et al.*, "15 years of research on redirected walking in immersive virtual environments," *IEEE Comput. Graph. Appl.*, vol. 38, no. 2, pp. 44–56, Mar. 2018.
[44] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[45] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.
[46] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[47] X. Li, N. Oertel, A. Hutter, and A. Kaup, "Laplace distribution based lagrangian rate distortion optimization for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 193–205, Feb. 2009.
[48] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[49] X. Zhang, L. Toni, P. Frossard, Y. Zhao, and C. Lin, "Adaptive streaming in interactive multiview video systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1130–1144, Apr. 2019.



Xue Zhang (Member, IEEE) received the Ph.D. degree in signal and information processing from Beijing Jiaotong University (BJTU), Beijing, China, in 2019. From 2015 to 2017, she was a Visiting Ph.D. student with the Signal Processing Laboratory (LTS4), Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. She was a Visiting Ph.D. student with the National Institute of Informatics (NII), Tokyo, Japan, in 2018. She is currently a Postdoctoral Fellow with York University, Toronto, ON, Canada. Her research interests include 3D image/video processing, interactive media navigation, and graph signal processing.

Gene Cheung (Fellow, IEEE) received the B.S. degree in electrical engineering from Cornell University, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 1998 and 2000, respectively.

From 2000 to 2009, he was a senior researcher in Hewlett-Packard Laboratories Japan, Tokyo. From 2009 to 2018, he was an Assistant then an Associate Professor with the National Institute of Informatics (NII), Tokyo, Japan. He is currently an Associate Professor with York University, Canada. His research interests include 3D imaging and graph signal processing. He served as a member of the Multimedia Signal Processing Technical Committee (MMSP-TC) in IEEE Signal Processing Society, from 2012 to 2014, and a member of the Image, Video, and Multidimensional Signal Processing Technical Committee (IVMSP-TC), from 2015 to 2017, and from 2018 to 2020. He is the coauthor of several paper awards, including the Best Student Paper Award in ICIP 2013, ICIP 2017, and IVMSP 2016, the Best Paper Runner-up Award in ICME 2012, and the IEEE Signal Processing Society (SPS) Japan Best Paper Award 2016. He was a recipient of the Canadian NSERC Discovery Accelerator Supplement (DAS) 2019. He has served as Associate Editor for multiple journals, including the IEEE TRANSACTIONS ON MULTIMEDIA, from 2007 to 2011, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, from 2016 to 2017, and the IEEE TRANSACTIONS ON IMAGE PROCESSING, from 2015 to 2019.

Yao Zhao (Senior Member, IEEE) received the B.S. degree from the Radio Engineering Department, Fuzhou University, Fuzhou, China, in 1989, the M.E. degree from the Radio Engineering Department, Southeast University, Nanjing, China, in 1992, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996.

He became an Associate Professor at BJTU, in 1998, and became a Professor in 2001. From 2001 to 2002, he was a Senior Research Fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He is a Fellow of the IET. He serves on the Editorial Boards of several international journals, including as an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS, a Senior Associate Editor of the IEEE SIGNAL PROCESSING LETTERS, and an Area Editor of *Signal Processing: Image Communication*. He was named a Distinguished Young Scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013.

Patrick Le Callet (Fellow, IEEE) is currently a Full Professor with the Computer Science Department, Engineering School (electrical engineering), Polytech Nantes / Université de Nantes. He is a member of the Steering Board of the CNRS LS2N Laboratory (450 researchers). He is also the Scientific Director of the cluster Ouest Industries Créatives gathering more than ten institutions (including three universities). Ouest Industries Créatives aims to strengthen Research, Education and Innovation of the Region Pays de Loire in the field of experience. He is mostly engaged in research dealing with the application of human vision modeling in image and video processing and cognitive computing. His current research interests include quality of experience assessment, visual attention modeling and applications, perceptual video coding, and immersive media processing. He is the coauthor of more than 300 publications and communications and co-inventor of 16 international patents on these topics. He is one of the founding members of EURASIP TAC (Special Areas Team) on Visual Image Processing. He was a co-recipient of an Emmy Award in 2020 for his work on development of Perceptual metrics for video encoding optimization. He serves or has been served as an Associate Editor or a Guest Editor for several journals, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING (IEEE TIP), the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (IEEE STSP), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (IEEE TCSVT), *EURASIP Journal on Image and Video Processing* (Springer), and *JEI* (SPIE). He has been serving in IEEE IVMSP-TC since 2015 and IEEE MMSP-TC since 2015.

Chunyu Lin received the Ph.D. degree from Beijing Jiaotong University (BJTU), Beijing, China, in 2011.

From 2009 to 2010, he was a Visiting Researcher with the ICT Group, Delft University of Technology, The Netherlands. From 2011 to 2012, he was a Postdoctoral Researcher with the Multimedia Laboratory, Gent University, Belgium. He is currently a Full Professor with BJTU. His research interests include image/video compression and robust transmission, 3D image and video processing, virtual reality video processing, and vision-based ADAS.

Jack Z. G. Tan received the M.Eng. degree from the State Key Laboratory of Modern Optics, Zhejiang University, China, in 2005, and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2009. From 2013 to 2014, he was a Chief Algorithm Engineer with Shenzhen Jieshun Technology. From 2014 to 2015, he worked as a Research Scientist and the Vice Director with Meizu Telecom Equipment Company Ltd. He currently works as the Director of Kandao Australia Research and Development Center and CTO of Shenzhen Kandao Technology Company Ltd.