User Navigation Modeling, Rate-Distortion Analysis, and End-to-End Optimization for Viewport-Driven 360° Video Streaming

Jacob Chakareski, Xavier Corbillon, Gwendal Simon, and Vishwanathan Swaminathan

***Abstract*—The emerging technologies of Virtual Reality (VR) and 360° video introduce new challenges for state-of-the-art video communication systems. Enormous data volume and spatial user navigation are unique characteristics of 360° videos that necessitate a space-time effective allocation of the available network streaming bandwidth over the 360° video content to maximize the Quality of Experience (QoE) delivered to the user. Towards this objective, we investigate a framework for viewport-driven rate-distortion optimized 360° video streaming that integrates the user view navigation patterns and the spatiotemporal rate-distortion characteristics of the 360° video content to maximize the delivered user viewport video quality, for the given network/system resources. The framework comprises a methodology for assigning dynamic navigation likelihoods over the 360° video spatiotemporal panorama, induced by the user navigation patterns, an analysis and characterization of the 360° video panorama's spatiotemporal rate-distortion characteristics that leverage preprocessed spatial tilling of the content, and an optimization problem formulation and solution that capture and aim to maximize the delivered expected viewport video quality, given a user's navigation patterns, the 360° video encoding/streaming decisions, and the available system/network resources. We formulate a Markov model to capture the navigation patterns of a user over the 360° video panorama and simultaneously extend our actual navigation datasets by synthesizing additional realistic navigation data. Moreover, we investigate the impact of using two different tile sizes for equirectangular tiling of the 360° video panorama. Our experimental results demonstrate the advantages of our framework over the conventional approach of streaming a monolithic uniformly-encoded 360° video and a state-of-the-art navigation-speed based reference method. Considerable average and instantaneous viewport video quality gains of up to 5 dB are demonstrated in the case of five popular 4K 360° videos. In addition, we explore the impact of two different popular 360° video quality metrics applied to evaluate the streaming performance of our system framework and the two reference methods. Finally, we demonstrate that by exploiting the unequal rate-distortion characteristics of the different spatial sectors of the 360° video panorama, we can enable spatially more uniform and temporally higher 360° video viewport quality delivered to the user, relative to monolithic streaming.**

***Index Terms*—Omnidirectional video, quality of experience, viewport-adaptive 360° video streaming, rate-distortion analysis and optimization, user navigation modeling.**

I. Introduction

The emerging technologies of virtual reality and 360° video are helping to introduce novel immersive digital experiences. It is anticipated that related products and applications will represent a $62 billion market by 2027 [1]. Gaming and entertainment, as well as education and training represent the main application domains of these technologies at present, with a broader set of societal applications spanning remote sensing, the environmental and weather sciences, disaster relief, and transportation anticipated in the future [2]. The recent feature article [3] provides a tutorial coverage of the diversity of virtual reality applications expected in the future across the spectrum of our society and their benefits that can broadly advance quality of life, energy conservation, and the economy, as well as the related research opportunities that can be pursued towards enabling them.

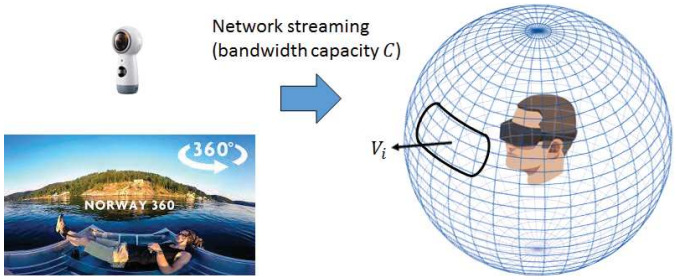


Fig. 1: 360° video streaming: Viewport V_i on the 360° sphere.

360° video is a recent video format captured by an omnidirectional camera that records incoming light rays from every direction. It enables a 360° look-around of the surrounding scene for a remote user, virtually placed at the camera location, on his VR device, as illustrated in Figure 1. Presently, existing systems stream the entire monolithic 360° view panorama to a user, who can, at any time, only experience a small portion of it denoted as viewport V_i, as also illustrated in Figure 1. However, this results in a huge network overhead/bottleneck and unnecessary computational/bandwidth loading of the device, which, in turn, considerably penalizes the user's quality of experience. Moreover, to apply traditional state-of-the-art video coding, the 360° view sphere is first mapped to a planar shape using a sphere-to-plane projection, as illustrated in the bottom left portion of Figure 1.

Two types of sphere-to-plane projections are commonly used: (i) direct projections such as the equirectangular projection (ERP) that map the latitude/longitude of a spherical point to planar coordinates on an equirectangle, or (ii) projections that use intermediate 3D objects such as a pyramid, cube, or dodecahedron, comprised of planar faces that are encoded independently. The latter have been considered since around 30% pixel replication is introduced when the sphere is mapped using an ERP [4, 5]. However, they have their own deficiencies, e.g., introduction of projection distortions around the planar shape's edges. In this paper, we only consider the equirectangular mapping, as it is one of the most widely used and one of the two (the other one being the cube map) currently considered by the MPEG's Omnidirectional File Format Standard (OMAF) [6].

The growing popularity of VR technologies stimulates an equivalent increasing demand for 360° video content, which today can be accessed through over-the-top online content

providers such as YouTube and Facebook 360 [7, 8]. However, present 360° video streaming practices necessitate excessive data rates that even anticipated broadband network access technologies would not be able to support [9, 10], due to the heuristic design shortcomings of the former outlined above. On the other hand, delivering the entire 360° view sphere is necessary to avoid simulator/motion sickness [11] that would degrade the quality of experience, as the *intuitive approach of sending only* V_i using traditional server-client delivery architectures, where the server responds to client updates, would preclude application interactivity, due to the inherent network round-trip-time induced latency.

This apparent impasse between 360° application requirements and present technology capabilities/design essentially stems from the direct application to the 360° domain of existing video coding/streaming technologies that treat 360° content as conventional video. Thus, recent studies have considered uneven spatial quality encoding of 360° videos, to minimize the data rate assigned to 360° regions not navigated by the user presently, thereby considerably reducing the induced network overhead [4]. This is the general strategy we also follow. In this paper, we present the following contributions:

1) We formulate a framework for viewport-driven rate optimized 360° video streaming that integrates the user view navigation patterns and the spatiotemporal rate-distortion characteristics of the 360° video content to maximize the delivered quality of experience of the user, for the given network/system resources. It comprises: (i) a methodology for assigning dynamic navigation likelihoods to the spatiotemporal 360° panorama that capture the probabilities of navigating different spatial segments of the 360° video content over time, by the user, (ii) an analysis and formulation of the spatiotemporal rate-distortion characteristics of compressed 360° video tiles that leverage preprocessed spatial tilling of the 360° view sphere, and (iii) an optimization problem formulation and solution that respectively characterize and aim to maximize the delivered QoE of the user, given the user's navigation patterns, 360° video encoding decisions, and the available system/network resources.
2) We formulate and analyze a user navigation Markov model to investigate the head navigation movements of a user in detail, and extend our dataset, at the same time. The proposed navigation models advances our system framework, by providing further insights into navigation aspects fundamentals of 360-degree video, and the capability to generate realistic navigation data without the need to capture one in reality, which can be quite costly and thus always of limited scale in practice. Thereby, experiments can be carried out over broader sets of navigation traces and user conditions, to provide further knowledge and insights.
3) We explore the impact of two different popular 360° video quality metrics applied to evaluate the streaming performance of our system framework.
4) We investigate the impact of two different tile sizes used in equirectangular tiling of the 360° video panorama, on the performance of our framework and 360° video streaming in general. These insights can help gain understanding of the benefits and drawbacks of using smaller tile sizes, and advance existing systems and implementations for 360° video streaming.
5) Finally, we study the spatial distribution of 360° video quality across the user viewport and demonstrate how our 360° rate-distortion optimization results in spatially more uniform viewport video quality, which in turn can considerably augment the user's quality of experience.

The rest of the paper is organized as follows. In Section II, we first review related work. Subsequently, we present the building components of our system framework and navigation model in Section III. The problem formulation that aims to maximize the delivered 360° user quality of experience given the user navigation patterns, 360° video encoding decisions, and the available system/network resources, is presented in Section IV. Experimental analysis of the performance of our framework and validation of our system models is carried out in Section VI. Finally, concluding remarks and a summary of envisioned future work are provided in Section VII.

II. Related Work

Studies of 360° video streaming to date have generally considered diverse challenges encountered in rendering the user viewport, streaming the 360° video panorama, and mapping the native 360° view sphere to a planar shape [12].

In particular, Afzal et al. carry out an empirical characterization of diverse characteristics of compressed 360° videos highlighting their main features, e.g., their lower temporal rate variability compared to conventional videos [13]. Moreover, a number of studies have considered splitting the 360° video into spatial tiles as part of the encoding process, leveraging the tiling feature of the latest High Efficiency Video Coding (HEVC) standard [14]. The encoding data rate of each tile can then be controlled independently to reduce the overall network streaming bandwidth usage [15–18]. Concretely, [15] explores a heuristic method for allocating bitrate to tiles comprising the 360° panorama, where viewport tiles are assigned a fraction of the available bandwidth, weighted by the proportion of their pixels present in the viewport area. Tiles entirely outside the viewport area are assigned a fraction of the remaining bandwidth weighted by their distance from the center point of the viewport. Similarly, [16] streams the tiles in the viewport at the highest possible quality, while the remaining tiles in the 360° panorama are assigned an equal portion of the remaining network bandwidth. The server push feature of the latest HTTP/2 protocol is used to deliver the tiled video faster and with better bandwidth utilization. Moreover, [18] adopts a similar view-aware approach, where tiles overlapping with the viewport are streamed at higher quality, while non-overlapping tiles are streamed at lower quality, to enable considerable network bandwidth savings. Finally, [17] proposes a regression based method for predicting the user viewport over the next streaming segment of the content, as part of a broader streaming framework that enables a client to request tiles overlapping with the predicted viewing direction of the user.

Xiao et al. propose using variable-sized tiles for an optimal viewport coverage and explore a linear programming and deep learning approach to identify the optimal tiling configuration over the 360° video panorama [19]. [20] explores DASH-based streaming of tiled 360° video, where different quality representations of the tiled content are available and coupled with several tile quality selection mechanisms to choose from. [21] exploits the stream prioritization and termination features of the HTTP/2 protocol to enable selective video frame dropping and scheduling within a content segment of a compressed 360° video, to maximize the amount of video data received on time by the client. [22] explores a probabilistic model for pre-fetching 360° video tiles that captures the distribution of the viewport prediction error and formulates a QoE-driven optimization framework that minimizes the total expected distortion of pre-fetched tiles. Similarly, [4] addresses the impact of the uncertainty of the upcoming user navigation actions in 360° video streaming by designing multiple representations of the 360° video content, characterized with different quality emphasized regions, which are adaptively served to the user. Rigorous analysis is carried out to select the number of representations and the locations and spatial sizes of the quality emphasized centers of each.

[23–25] explore neural network based strategies for accurate viewport prediction in 360° video and VR streaming. Moreover, live 360° video multicast over an LTE network is investigated in [23, 26], addressing aspects such as user clustering and allocation of resources to different parts of the 360° video content. [23] also explores the use of scalable 360° video tiling and rigorous rate-distortion optimization of the resource allocation problem. Finally, in three recent more distantly related studies, [27] explores a system framework for aerial multi-viewpoint 360° video streaming of a remote scene to a VR user, integrating reinforcement learning, resource scheduling, and 360° video representation advances, to maximize the quality of experience of the user, given the available system resources. On the other hand, [28] explores viewport-adaptive multi-user VR mobile-edge streaming in 5G small-cell systems, where a collection of 5G small cell base station can pool their communication, computing, and storage resources to collectively deliver scalable 360° video content to mobile VR clients at much higher quality. Yet, [29] studies the integration of millimeter wave (or free-space optics) and sub-6 GHz links for dual-connectivity streaming of six-degrees of freedom VR content to multiple mobile users navigating the content across the spatial area of an indoor VR arena.

In contrast to the few studies cited above that consider HEVC 360° tiling, we employ preprocessed spatial tiles of the 360° view panorama, as introduced later, which has several advantages in the form of lower complexity at multiple critical aspects of a server-client 360° streaming architecture [4]. Moreover, formal analysis of the spatiotemporal rate-distortion characteristics of 360° tiling that integrates the user navigation patterns and the available network/system resources has not been carried out towards optimal selection of 360° encoding/streaming decisions. The framework of our paper aims to fill this gap. The present paper builds upon our preliminary work in [30], introducing the following additional major advances over [30]: (i) Analysis of the impact of the equirectangular tile size and the enabled gains by using smaller tiles. (ii) User navigation Markov model developed from real user head navigation traces. (iii) Analysis of temporal and spatial viewport quality variation. (iv) Exploration of the impact of two different 360° video quality metrics. (v) A comprehensive experimental analysis over a broad range of operating conditions and an extensive 360° video dataset.

III. System Models

Our 360° network system architecture comprises several major component blocks and is illustrated in Figure 2. In particular, based on the tiling preprocessing of the 360° video content that is carried out ahead of time, rate-distortion analysis is carried out for each GOP tile. In parallel, navigation likelihoods are developed based on existing navigation traces from the current and prior VR users, as well as synthetic traces generated from the actual navigation traces corpus using the first order user navigation Markov model that we explore. Finally, using these two system components, viewport-adaptive streaming optimization is carried out to assign the appropriate network data rate to each tile of the current GOP of the content delivered to the user, as illustrated in Figure 2. We describe each system component in detail in the following subsections.

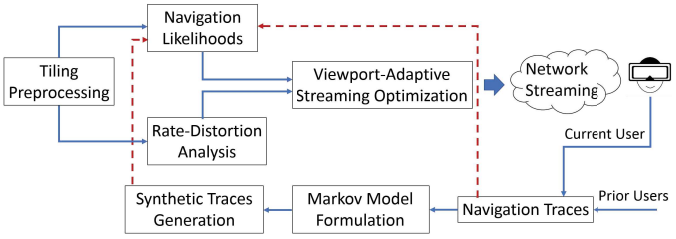


Fig. 2: Key components of our 360° network streaming system.

A. Tiling Preprocessing

We partition an equirectangular 360° video into a set of $N \times M$ spatial tiles. In particular, we partition the raw 360° video frames into spatial tiles and consider the collection of thereby constructed (smaller) video frames for each tile as separate videos. The tiles are then separately encoded off-line (ahead of time) and streamed to the user on-demand, according to our analysis and optimization. As explained earlier, carrying out the tiling as a preprocessing step has several advantages over tiling the video as part of the encoding process, as enabled by the tiling feature of the latest video coding standard HEVC. In our experiments, we preprocessed 4K 360° videos into two different spatial tiling settings: large 6×4 tiles, as illustrated in Figure 3, and small 8×8 tiles (Figure 4) where the first and second dimension respectively refer to the horizontal and vertical number of identical tiles in the video. Each tile is indexed in a raster fashion, left-to-right and top-to-bottom.

We selected these tiling settings based on empirical analysis, as a reasonable choice between the complexity and compression efficiency introduced by a given tiling set. Using a large tile size has the advantage of compression efficiency [31] and

Fig. 3: 360° video panorama 6×4 spatial tiling.

Fig. 4: 360° video panorama 8×8 spatial tiling.

smaller number of additional Media Presentation Description (MPD) files generated for each independent video tile [4]. Small tiling, on the other hand, is less complex in terms of video compression carried out per tile. In addition, using smaller tiles allows for constructing a more accurate expected user viewport by having smaller building blocks spanning the 360° video panorama. Compared to using a bigger tile size, this will result in a more optimal network resource usage and is expected to have higher average viewport quality delivered to the user, when integrated with a viewport-driven resource allocation such as the one we pursue here. Lastly, we note that though using smaller tiling will penalize video compression efficiency, it appears that in practice the induced penalty may not be severe, as reported in a recent study carried out by a VR streaming spin-off company, where 100 tiles were used to partition an 8K 360° video panorama [32].

B. 360° VR Head Movement Data

We collected head-movement data that describes how a user navigates a 360° video over time. In particular, a VR device outputs the direction of the current viewpoint of the user V_i on the 360° view sphere up to 250 times per second, with the user considered to be placed at the sphere center, as described earlier in Figure 1. Precisely, this is the surface normal of V_i on the 360° sphere that is uniquely described by the spherical coordinates azimuth and polar angles $\varphi \in [0°, 360°]$ and $\theta \in [0°, 180°]$ it spans on the sphere, in a spherical coordinate system with the 360° unit sphere center as its origin, as illustrated in Figure 5 (right). These two angles are equivalently denoted as yaw and pitch in the VR community, captured as rotation angles around the Z and Y axes, as denoted in Figure 5 (left). We collected the pairs (φ_j, θ_j) that coincided with the discrete temporal instances t_j of

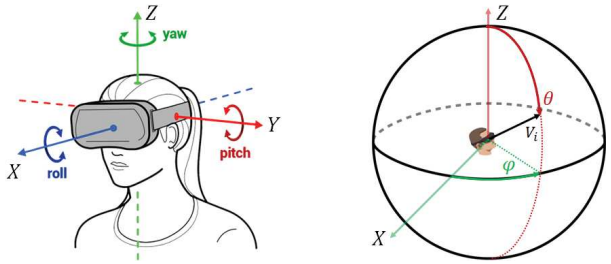


Fig. 5: 360° head movement navigation data of current viewport V_i. Left: Rotation angles yaw, pitch, and roll around the three coordinate axis. Right: Azimuthal and polar angles (φ, θ) in spherical coordinates.

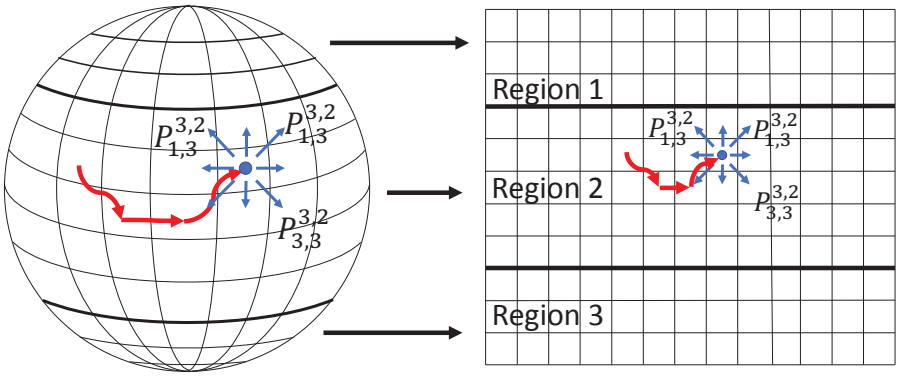
subsequent 360° video frames j displayed to the user as she navigates the content. They are the navigation data points relevant for our analysis. We note that we omitted capturing the head orientation angle *roll* as part of our navigation traces for simplicity and most importantly, because its values are predominantly very small and thus have insignificant impact on establishing the user viewport on the 360° view sphere or panorama [33, 34]. We provide further details about our navigation data capture procedure and overall experimental setup in Section VI-A later.

In addition to our own recorded data traces, we used the navigation dataset of Corbillon et al. [35]. Having navigation traces for a bigger number of captured 360° videos will extend the dataset we can work with, to help produce statistically more significant/reliable results devoid from any related anomalies that can arise from working with a smaller statistical sample. Relatedly, more recently, in another study, we have carried out an extensive data collection and preparation initiative, to share with the broader community full UHD (8K) 360° video navigation traces and rate-distortion compression information, to facilitate research of next generation 360° video and VR streaming systems, and beyond [34].

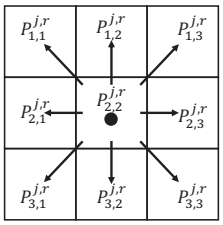
C. 360° Navigation Model

Having actual data traces of a user watching a 360° video recorded using a Head-Mounted Display (HMD) device is necessary to analyze the navigation patterns of a user and the induced expected viewport quality, in diverse settings. However, in order to accurately predict the expected user viewport from user actions, a large number of navigation traces are required, especially in the case of small video tiles. In order to generate a large navigation trace dataset, we investigate the statistical dependencies of user navigation traces.

We are interested in a user navigation model based on user actions made between consecutive video frames. We develop a parameterized first order Markov model that captures the present user action based on the previous action of the user and the location of the present action on the 360° view sphere, integrated as a parameter of the model. In the real user navigation traces we have worked with [34, 35], more than 99% of the time, a user does not navigate more than one macroblock, which has a size of 64 pixels in both vertical and horizontal directions. We note that a macroblock is a basic unit of planar video compression that has been used in every subsequent video coding standard. Its default size in the present HEVC standard is 64×64 pixels [14] in the



(a) Video regions and navigation actions on the 360° view sphere and associated equirectangular 2D panorama. Video region $r = 2$ and index of last navigation action (diagonally to the upper right macroblock) is $j = 3$.



(b) Prospective choices for the subsequent navigation action, formulated by our model, and their respective likelihoods.

Fig. 6: Proposed navigation model: Sample trajectory and next navigation action options.

2D plane of a video frame. We also note that the concept of a macroblock is different from the concept of a 360° video tile that was introduced earlier. A macroblock is spatially much smaller than a video tile, macroblocks are an inherent component of video compression, while tiling can be used optionally, and a tile will comprise multiple macroblocks.

In our model, we first divided the 360° video panorama into macroblocks, using the default macroblock size noted above. This choice will allow us to capture accurately the center of the user's attention and the user's direction of navigation with a limited number of macroblocks, to control the complexity of our model. Using a smaller macroblock size, e.g., 16 ×16, as used in previous codecs, would unnecessarily increase the model's complexity, without any benefit to its accuracy. Now, a viewport center at a given time has 9 possible navigation actions: it can navigate to one of 8 neighbors of its current macroblock position or stay in the present location/macroblock (see Figure 6b). The probabilities of each action $P^{j,r}_{m,n}$ are determined using actual data traces where *j* and *r* represent the previous action and region, respectively, and *m* and *n* represent the horizontal and vertical indices of the (adjacent or same) macroblock to which a transition can be made at the next navigation action/step, as explained above. We integrate these two aspects to make our model more realistic: the previous navigation action and the video region where it took place.

In particular, a user navigating a 360° video is considered by Sitzmann et al. [36] to be in one of the following two states: (i) *attention* and (ii) *re-orientation*. In the *attention* state, the user tends to navigate less and in the *re-orientation* state the user explores the panorama by constantly moving. In order to capture these two states, we calculated the probabilities of each prospective subsequent action given the most recent navigation action of the user already taken. Because prior actions will influence the outcome of subsequent actions, by formulating our model thereby, we can account for persistent *re-orientation* and stationary *attention* navigation actions/states of the user. In our model, there are nine prospective subsequent actions/states ($m, n \in \{1, 2, 3\}$), for every prospective outcome for the most recent prior action $j \in \{1, \ldots, 9\}$.

Moreover, head navigation movements around the equator and near the poles are usually not same in the equirectangular 2D plane. For instance, a yaw navigation movement of one macroblock size near the equator corresponds to a size of several macroblocks near the poles of the 360° view sphere. This distinction affects the recurrence of various navigation movements, depending on the location of a movement on the 360° view sphere. To capture this phenomenon, we divided the 360° panorama into 3 regions: 2 polar regions and one equatorial region, indexed by r, i.e., $r \in \{1, 2, 3\}$. Then, based on the region where a navigation action is initiated, the probability of each prospective choice for the subsequent navigation action/decision will depend on its recurrence in that region only. Now, we note that although small angular head movements may lead to multi-macroblock distances in polar regions, the fraction of such navigation movements in our dataset is negligible. Thus, in our modeling, we kept using a one macroblock size movement also for polar regions, with region-specific probabilities.

By integrating the two aspects described above, our navigation model is as follows, and illustrated in Figure 6. First, the movement region *r* in the 360° panorama is determined. Then, based on the previous action (red arrows in Figure 6a) and the corresponding region r, the probability model $P^{j,r}$ is selected. Alike to the prospective choices for the current navigation action, we denote the previous action's nine options as: moving to one of the eight neighboring macroblock ($j \in (1, ..., 4) \cup (6, ..., 9)$) and making no movement ($j = 5$), i.e., staying in the same macroblock. The probabilities of each prospective next navigation movement, indexed by (m, n), (blue arrows and blue dot in Figure 6a) are $P^{j,r}_{m,n}$, where *m* and *n* indicate the indices of the navigation probabilities in the 3×3 matrix $[P^{j,r}]$, parameterized by the navigation region (*r*) and previous navigation action (*j*), as introduced earlier. We use our model to generate synthetic navigation traces.

D. Navigation Likelihoods

For various HMD used in VR applications, the viewport size experienced by the user varies. In this paper, we assume a viewport of 110° horizontal and 90° vertical fields of view. For every navigation trace for a given 360° video, we compute the fraction of the surface area of tile k occupied by the user

viewport V_i at time instance t, denoted as $w_{k,t}$. To account for the unequal surface area occupied by different viewports, when mapped to a 2D rectangle used to encode the data, depending on their latitude (polar angle θ) on the 360° view sphere (see Figure 10 and Figure 11, and the related discussion therein), each tile k is assigned a normalized weight $\bar{w}_{k,t}$, computed as $\bar{w}_{k,t} = w_{k,t} / \sum_k w_{k,t}$. We can then aggregate these weights over different time durations, to compute the likelihoods of navigating different tiles of the respective 360° video during those time periods, e.g., $p_k = \sum_{t=t_1}^{t_2} \bar{w}_{k,t}/(t_2 - t_1 + 1)$. In our analysis, we are interested in exploiting these navigation likelihoods over the duration of individual Group of Picture (GOP)s comprising the encoded 360° video content. In particular, for each GOP, we assign *dynamic navigation likelihoods* to each tile by aggregating navigation likelihoods of similar user navigation traces, i.e. sharing the GOP's initial tile for their viewport center. Since users are following the contents of the video, it is expected to for them to have similar short-term navigations given they started at the same point.

For illustration, Figure 7 shows the average (over the entire video) navigation likelihoods of different tiles comprising the selected 6×4 tiling applied to the 360° video Roller Coaster [37] used in our experiments. We can see that corner tiles appear rarely in a viewport navigated by the user, as their navigation likelihoods are close to zero. Conversely, it appears that the user often navigated through tiles 15 and 16, for instance, as they have much higher navigation likelihoods.

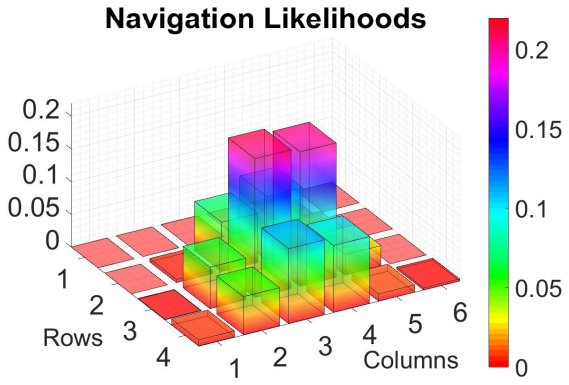


Fig. 7: Navigation likelihoods of tiles for Roller Coaster. [30]

Figure 8 shows the corresponding tile navigation likelihoods for the second 360° video, Wingsuit, used in our experiments. It appears that in this case the viewport navigated by the user is mostly closer to the south pole, as the corresponding tiles have much higher likelihoods now, due to the specific nature of this video (more interesting content is spatially located there).

In Figure 9, we explore the navigation likelihoods of the small tiling setting for the 360° video Roller Coaster. We can see that here there are more tiles in the user viewport with similar/more uniform navigation likelihoods, compared to the case observed in Figure 7. This is expected and is due to the smaller size of tiles that is used here. The finer tiling will enable delineating the user viewport more accurately, which in turn will enable a more precise rate-distortion optimized

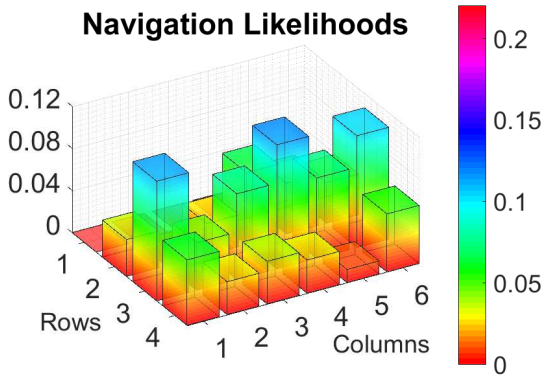


Fig. 8: Navigation likelihoods of tiles for Wingsuit. [30]

allocation of resources across the 360° video panorama. This, in turn, will enable a higher quality viewport for the user and will augment his or her quality of experience. On the other hand, using a bigger number of tiles to partition the 360° video panorama will increase the processing complexity for the video encoder at the streaming server and will reduce the compression efficiency. The former may not be such a challenge for 360° video on demand applications, as considered in this paper, as it can be carried out off-line (ahead of time). The latter, on the other hand, can increase the requirements for network streaming bandwidth, though there are some recent industry studies that provide evidence of only a small penalty in compression efficiency, even when high number of tiles (100) is used [32]. Regardless, the benefits versus drawbacks of using tiling and the selection of the tile size, need to be carefully evaluated for the target application in mind.

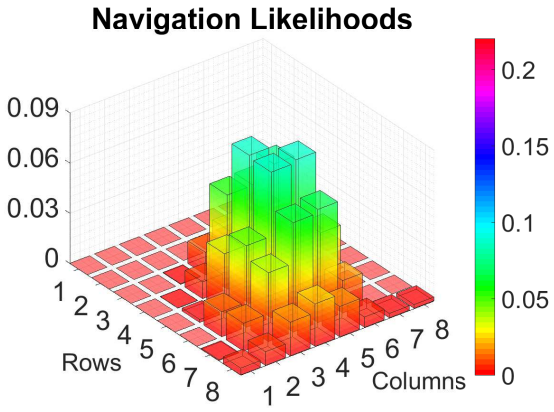


Fig. 9: Navigation likelihoods of small tiles for Roller Coaster.

A visualization of two representative viewports on the 360° panorama is shown in Figures 10 and 11. Since mapping a 3D sphere to an equirectangular 2D plane causes a stretching distortion, the shape of a viewport will also change depending on its spatial location, in particular its latitude on the 360° view sphere as specified by its polar angle θ (see Figure 5 and Section III-B). In equatorial regions, a viewport is smaller and more compact (see Figure 10), while in polar regions of the 360° panorama a viewport is spread over all polar tiles (see Figure 11). This distinction visually clarifies and demonstrates

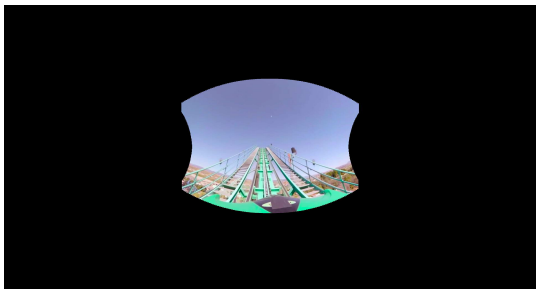


Fig. 10: Viewport at $(\varphi, \theta) = (0°, 0°)$.

the need for the integrated normalization of the tile-viewport intersection weights $w_{k,t}$ carried out as part of the formulation of the tile navigation likelihoods described in this section. Moreover, Figures 3 and 4 can be referenced to understand the spatial locations of these two viewports relative to the underlying tiling of the respective 360° video.

Fig. 11: Viewport at $(\varphi, \theta) = (120°, -60°)$.

E. Rate-Distortion Models

Changing the quality of tiles is a useful method to control the bitrate of a 360° video. The Quantization Parameter (QP) employed by the HEVC codec is a convenient tool for tile quality adaptation. We explore two prospective characterizations of the dependency between the parameter QP and the resulting bitrate R of the encoded tile. That is, we investigate modeling $R = f_1(\text{QP})$ via an exponential or power law function for f_1 as follows

$$R = a_1 e^{-b_1 \text{QP}} \quad \text{or} \quad R = a_2 \text{QP}^{b_2}. \tag{1}$$

We will validate these relationships by comparing the bitrate and QP for an encoded 360° tile in Section VI-B. Since we have a function between the bitrate and QP, we can define bounds for our optimization problem with the highest and lowest QP values that can be selected. And after calculating the optimal bandwidth, going back to QP value and encoding the tiles accordingly can be done easily at the server side.

Similarly, we investigate two prospective characterizations of the dependency between the encoded tile bitrate R and the induced reconstruction error or distortion D for a tile, where the latter can be calculated as the Mean Square Error (MSE) between the encoded tile video data and the corresponding raw video data for the tile. In essence, the distortion D captures the average deviation of encoded tile pixels from their raw data counterparts. In a raw 360° YUV 4:2:0 video, for every pixel sample of the color (chrominance) components U and V there are 4 pixel samples of the (monochromatic) intensity (luminance) component Y. Thus, the luminance distortion dominates the encoding distortion for the two color components. Therefore, we used the luminance component distortion as the representative of the encoding distortion D for a tile, measured for every 360° tile luminance video frame. We investigate modeling the dependency $D = f_2(R)$ via an exponential or power law function for f_2 as follows

$$D = c_1 e^{-d_1 R} \quad \text{or} \quad D = c_2 R^{d_2}. \tag{2}$$

We will also validate these relationships by comparing the encoding bitrate and distortion for an encoded 360° tile in Section VI-B. The characterizations $R = f_1(\text{QP})$ and $D = f_2(R)$ will allow us to formulate the aggregate 360° video encoding quality and pursue related optimizations, as explained in the next section.

IV. Optimization Framework

Given the analytical modeling of the relevant problem variables, we now set out to find the optimal bitrate for each tile. We integrate two new constraints into the problem formulation. These are the aggregate available network bandwidth C and the allowed QP range per tile.

A. Problem Setup

Given the limited network bandwidth, tiles should be transmitted with a data rate corresponding to their navigation likelihoods and rate-distortion characteristics such that we can minimize the distortion (and maximize the delivered aggregate quality) of the respective 360° video. Let $R_i(\text{QP}_i)$ denote the bitrate of the i$^{\text{th}}$ tile where QP is the encoding quantization parameter, as introduced earlier. This gives us the following inequality to maintain:

$$\sum_i R_i(\text{QP}_i) \leq C, \; i = 1, \ldots, M \times N. \tag{3}$$

For practical reasons, for every tile i we set a range of QP values that can be considered, defined by the upper and lower bounds $\text{QP}_{\min}$ and $\text{QP}_{\max}$. This therefore induces constraints on the minimum and maximum data rates that can be assigned to a tile, given the monotonic relationship between QP and R_i, as captured by the function $R_i(\text{QP})$. Formally, these two constraints can be written as

$$R_i(\text{QP}_{\max}) \leq R_i(\text{QP}_i) \leq R_i(\text{QP}_{\min}). \tag{4}$$

Finally, we formulate the expected 360° quality of experience that a user observes while navigating the scene, as the navigation likelihood weighted sum of video qualities of all tiles comprising the 360° video content streamed to the user. This can be formally written as $\sum_i p(i|v) D_i(R_i)$, where $p(i|v)$ denotes the navigation likelihood of tile i given that viewport v is requested initially. To be precise, note that we formulated our objective as the expected 360° video distortion, due to the one-to-one correspondence between video quality and reconstruction error (distortion). Therefore, we aim to minimize our objective function, as it will lead to the same goal (maximum 360° quality of experience).

B. Optimization Formulation

Leveraging the problem setup described earlier, we can now formulate the optimization problem of interest as

$$\begin{aligned} \min_{\{R_i\}} \quad & \sum_i p(i|v) D_i(R_i), \\ \text{subject to:} \quad & \sum_i R_i(\text{QP}_i) \leq C,\ i = 1, \ldots, M \times N, \\ & R_i(\text{QP}_{\max}) \leq R_i(\text{QP}_i) \leq R_i(\text{QP}_{\min}),\ \forall i. \end{aligned} \tag{5}$$

Note that (5) represents a convex optimization problem, due to the nature of the constraints involved and the objective function under consideration. Therefore, it can be efficiently solved using Lagrange multipliers [38]. In our experiments, we carry out the optimization in (5) for every GOP, facilitating the dynamic navigation likelihood assignment described in Section III-D to compute the navigation likelihoods $p(i|v)$. In particular, after the optimization completes, the QP vs R dependency for each tile i in a GOP is used to obtain the explicit optimal QP_i value that corresponds to the optimal data rate R_i^* produced by (5). Note that for illustration we included the average navigation likelihoods $p(i|v)$ across the applied 360° video tilling for the duration of the entire video in Figures 7 and 8, for the 360° content used in our experiments.

We recall that the analytical dependencies between R and D, and between R and QP are not explicitly denoted in (5). As explained earlier, we explore two models for each dependency $D = f_2(R)$ and $R = f_1(\text{QP})$, an exponential one and a power-law one. And the parameters that comprise each model are extracted uniquely for each tile, before we carry out the optimization in (5). In our experiments, we first validate each of these models, for each dependency, and select the one that is more accurate, to carry out the remaining performance evaluation analysis.

V. Implementation Aspects

We note that in our setup, just as in any other on-demand video streaming systems, multiple versions of a given 360° video are constructed/encoded at different data rates off-line, with a preprocessing step of tiling involved first, and then streamed from to a user requesting that content on-demand online. Thus, the setup we consider falls closely under the general DASH-umbrella for on-demand video streaming. The following aspects of our framework can be implemented and integrated with a DASH-compliant system as follows. The rate-distortion analysis and optimization can be carried out respectively off-line, at the server (the former) and online at streaming time, at the client (the latter), within an actual DASH system. Moreover, the necessary rate-distortion models, necessary to run the optimization can be shared by the server with the client at the start of the streaming media session. The DASH-based client will use its estimates of the available download data rate that are typically carried out on its end within DASH to supplement the optimization with the right value of the available network streaming bandwidth C in (5).

VI. Experimentation

A. System Setup

We used five popular 4K 360° videos from Youtube to evaluate the performance of our framework. 48 VR users watched the first two videos (Roller Coaster [37] and Wingsuit [39]) with an Oculus Rift HMD device while their head movements have been tracked using the OpenTrack software [40]. The other three videos (Elephant [41], Timelapse [42], and Diving [43]) are from the Corbillon et al. dataset and each of them has between 40-58 head movement traces. We generated 500 synthetic navigation traces for each video as discussed in Section III-C. Then, we calculated the tile navigation likelihoods for each GOP, as formulated earlier, across all available navigation traces, for a given 360° video, to use them in our optimization and experiments later on.

Each 360° video is preprocessed and compressed off-line using 2 different equirectangular tile sizes: large 6×4 tiles and smaller 8×8 tiles. Each tile is encoded into GOPs of size 32 frames using HEVC. Each video consist of 10 GOPs, which corresponds to 320 frames and 10.6 seconds temporal duration, with a frame rate of 30 fps. Each GOP is encoded using 5 QP values (22, 27, 32, 37, 42). Using the compressed 360° video tiles for these QP values, we extracted the $R-D$ and $QP-R$ parameters for our related analytical modeling, across all tiles and GOPs, to explore and validate the proposed rate-distortion modeling from Section III-E.

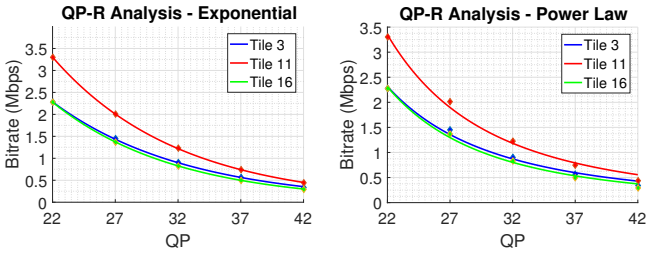
Two reference methods are examined to compare against our optimization framework denoted as *Proposed*. The first one is *Monolithic* where an entire monolithic 360° video is encoded using the following 5 QP values (32, 34, 36, 39, 42). In each case, the induced average data rates for every GOP are used as the network bandwidth constraint C in our own optimization in Section IV-A. The second one is *Speed-based*, a state-of-the-art method proposed by Petrangeli et al. [16]. It predicts future viewports accessed by the user, based on the speed and the position of the current viewport center. Tiles within the current/future predicted viewports in a GOP are encoded with the highest possible QP value. The remaining tiles are encoded with the lowest possible QP value given the remaining bandwidth budget. Finally, *Proposed* indicates the framework introduced in this paper.

We note that space limits precluded us from including the entire corpus of results present in the paper, for every single 360° video used in our experiments. Moreover, we observed that the relative performance of the reference methods and the proposed framework remains consistent across the entire 360° video corpus, for each experimental evaluation we considered. For these two reasons, we decided to vary the specific one or two 360° videos used in a given evaluation considered in Section VI, to increase the breadth of results presented in the paper across the entire video corpus we used.

B. Rate-Distortion Model Validation

We formulated two prospective models for the dependencies $D = f_2(R)$ and $R = f_1(\text{QP})$, described in Section III-E. Here, we explore their accuracy in characterizing the encoded 360° video content we considered in our experiments.

Examination of the employed QP versus induced bitrate relationship for different tiles shows that exponential model fits better the actual data points. In Figure 12, we examine these data points, shown as markers, and the fitted analytical dependencies according to the two formulated models, for three representative tiles, with diverse rate-distortion characteristics, from the Roller Coaster video. Referencing the tile indexing from Figure 3, we can see that while tiles 3 and 16 show lower bitrate requirements due to their relatively static nature, tile 11 requires a higher bitrate as it corresponds to a more dynamic 360° region.

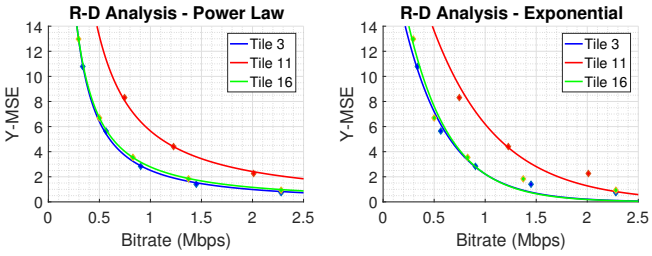


(a) QP vs. bitrate dependency using an exponential model. [30]

(b) QP vs. bitrate dependency using a power law model. [30]

Fig. 12: QP vs. bitrate dependency for different tiles. Actual data points shown as markers.

Figure 13 shows the advantage of the power law model in describing the observed D versus R dependency, denoted with markers, across the 360° video tiles. In particular, for lower bitrates, the impact of higher distortion dominates for tiles with more dynamic content (Tile 11), while for higher bitrates the difference across different tiles in this regard becomes smaller, as seen from Figure 13.



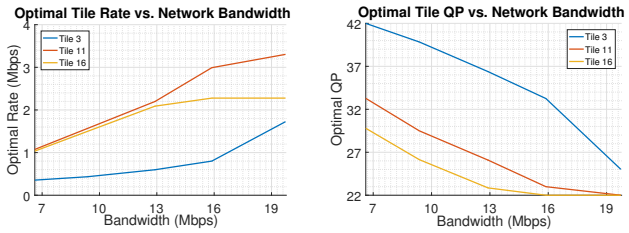
(a) Bitrate vs. distortion dependency using a power law model. [30]

(b) Bitrate vs. distortion dependency using an exponential model. [30]

Fig. 13: Bitrate vs. distortion dependency for different tiles. Actual data points shown as markers.

C. Optimal QP and Bitrates vs. Available Bandwidth

We examine how the optimal data rates R_i and the corresponding QP_i values, produced by the optimization in (5) for every tile i, vary, as the available network bandwidth C is varied. Figure 14a shows the optimal rates produced by (5) for three tiles from the Roller Coaster video, for the GOP number 57 in the 360° video, selected as a representative example. For this GOP, tile 3 has a small navigation likelihood, while tile 16 has the highest among the three tiles considered. Still, it is interesting to note that although tile 16 has a higher navigation likelihood relative to tile 11 and is assigned a smaller QP earlier (as seen from Figure 14b), encoding tile 11 leads to a higher data rate in the second half of the graph in Figure 14a, due to its more dynamic content, which makes encoding it more challenging.

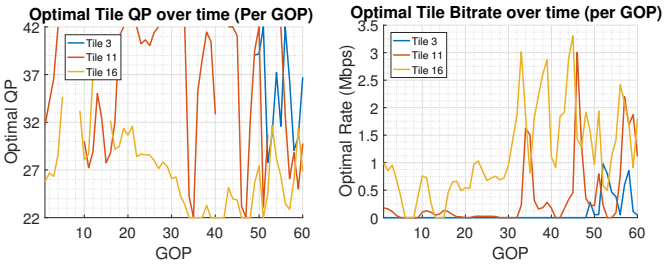


(a) Change of bitrate in bandwidth. [30]

(b) Change of QP in bandwidth. [30]

Fig. 14: Optimal values in various bandwidths

Figure 15 shows the temporal evolution of the optimal QP and bitrate values for these three tiles over the GOPs comprising the 360° content while using the corresponding bandwidth value C for each GOP. We can see that tile 16 typically has a lower QP value relative to the other two tiles, due to its frequently accessed spatial location, while tile 3 is often navigated only for a brief period of time towards the end of the video. Discontinuities in Figure 15a indicate that a tile has not been assigned any rate (skip encoding mode) by the optimization in (5), as indicated by the corresponding graphs in Figure 15b.



(a) QP over time. [30]

(b) Bitrate over time. [30]

Fig. 15: Optimal values in various GOPs

D. Expected 360° Video Quality

For the three 360° video streaming systems under comparison, we measured the video quality per viewport experienced by a user navigating the 360° content, as the Luminance Peak Signal to Noise Ratio (Y-PSNR) of the MSE of the pixels displayed in the viewport. Figure 16 shows the viewport Y-PSNR over time for the three competing systems in the case of the Timelapse video. We can see from Figure 16 that *Speed-based* and *Monolithic* exhibit the same temporal pattern in viewport Y-PSNR variations, as the dynamic 360° content evolves, with our framework outperforming the both method consistently and considerably. We also observed that *Speed-based* offers an improved performance over *Monolithic*, when viewport prediction succeeds. Though there are minor variations for some frames, we observed that on average *Proposed* provides a 4.5 dB gain over *Monolithic*.

Speed-based framework utilizes the user head navigation to predict expected viewport and aims for a uniform QP distribution in the expected viewport. This allows reaching higher qualities in viewport. *Proposed* framework, on the other hand, exploits the rate-distortion characteristics of video tiles

to reach a uniform expected quality. The non-linear nature of a rate-distortion curve enables reaching higher viewport quality for the same network bandwidth budget, for our framework. This results in a more consistent and uniform quality and QoE for the user viewport.

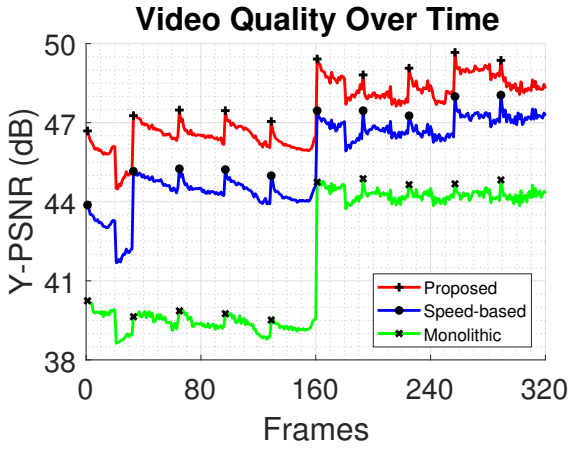


Fig. 16: 360° viewport video quality: Timelapse.

In Figure 16, the markers indicate the initial frames of GOPs. Since we encoded the frames of each GOP using uniform QP values, the I frames have higher Y-PSNR values, for all three methods under comparison. We observed in this evaluation, as expected, that using lower QP values lead to higher Y-PSNR values overall and cause a higher gap between the I frames and the subsequent P/B frames. Figure 17 shows the average Y-PSNR values of each frame in a GOP, averaged over 50 users. Here, we observe similar Y-PSNR trends for all three methods under comparison, with decaying intensity over the GOP. Especially for higher QP values, we observed a maximum of 1.5 dB difference across the duration/length of a GOP for *Proposed*, and slightly smaller values of 1 dB for *Speed-based*, and 0.5 dB for *Monolithic*.

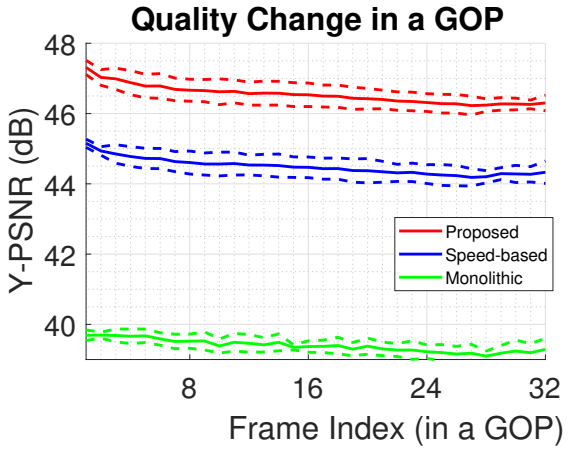


Fig. 17: Intra-GOP (per frame) PSNR trend: Timelapse.

Next, we examine the average (over time) viewport 360° video quality (Y-PSNR) delivered by the three competing systems, as the available network bandwidth C is varied. Figure 18 show these results for average of 5 users in the case of Timelapse. We can see that again *Proposed* outperforms *Speed-based* and *Monolithic*, with a consistent gain more than 4 dB, in C values higher than 2 Mbps. On the other hand, the reconstruction error can vary more spatially across pixels in viewports delivered by *Proposed* and *Speed-based*, due to the applied tiling, especially as the number of tiles that comprise a viewport increases.

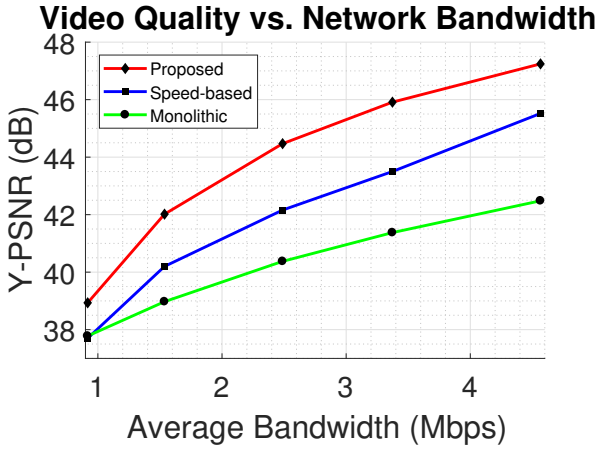


Fig. 18: Average 360° viewport video quality: Timelapse.

The cumulative density function (CDF) of the Y-PSNR values of Timelapse video are compared in figure 19. *Monolithic* approach has the steepest CDF curve since its Y-PSNR values has a very small overall variance. Conversely, *Speed-based* has more gradual curve where it drops below the *Monolithic* case due to occasional mispredictions. Misprediction in *Speed-based* results in very low viewport quality since only expected viewport tiles are encoded with the high quality. *Proposed* approach has the highest overall Y-PSNR values as a result of efficient rate-allocation. However it has the most gradual shape especially due to increasing Y-PSNR differences in higher qualities.

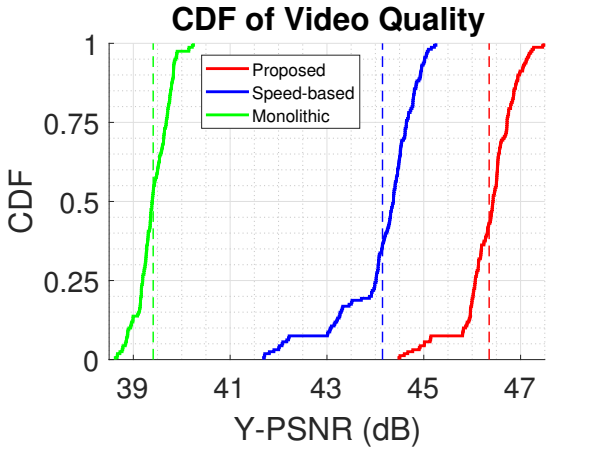


Fig. 19: CDF of Y-PSNR for all frames: Timelapse.

Finally, in Table I, we examine the viewport quality performance of all three methods under comparison and the performance gains of our *Proposed* framework relative to *Monolithic*, across all five 360° videos used in our experiments, for three different network bandwidth values (*high, medium, low*) used to compress and stream each 360° video content to a VR user. First, we can observe that our framework

considerably outperforms the two competing methods across all 360° videos and network bandwidth values considered, due to its integrated advances spanning rate-distortion analysis, system resource optimization, and navigation modeling. The gains over *Monolithic*, a method that is most dominantly used in current industry practices, are the highest and most consistent in the cases of the Timelapse, Wingsuit, and Diving videos. These benefits are promising and merit further investigation and pursuit of practical implementations of our system. We note that we observed more significant correlation in the navigation patterns of users, for the Timelapse and Diving 360° videos. This can enable a more accurate viewport prediction and thus a better utilization of the allocated network resources. Moreover, we also note that given its dynamic content nature, the 360° video Wingsuit required much higher network streaming bandwidth to be compressed and delivered at comparable viewport quality relative to the other four videos used in our experiments. Similarly, we further note that different network streaming bandwidth values were considered to compress and deliver each 360° video, in order to enable comparable encoding qualities in each case (*high, medium, low*), across the entire 360° video corpus we considered. This need arises from the heterogeneous spatiotemporal rate-distortion characteristics and dynamics that each 360° video content features.

Video name	**Network Bandwidth (Mbps)**	**Proposed (dB)**	**Monolithic (dB)**	**Gain** (wrt **Mono.**) **(dB)**
Roller Coaster	4.28	47.50	45.34	2.16
	2.59	44.03	42.78	1.25
	1.23	38.95	38.99	-0.04
Elephant	4.02	49.70	46.23	3.57
	2.20	46.84	44.19	2.65
	0.76	41.40	41.45	-0.05
Timelapse	4.57	47.19	42.58	4.61
	2.49	44.37	40.48	3.89
	0.92	38.97	37.87	1.10
Wingsuit	12.42	52.91	48.66	4.25
	6.38	50.49	46.88	3.61
	2.12	47.67	44.37	3.30
Diving	6.17	49.43	45.68	3.75
	3.42	46.61	43.57	3.04
	1.28	42.37	40.98	1.39

TABLE I: Viewport quality performance and gains over *Monolithic*.

In closing this section, we note the following aspects about the performance evaluation carried out herein. We introduced the speed-based system as another reference method to illustrate that our proposed framework can broadly outperform state-of-the-art as well as commonly used methods. In particular, we have provided five different sets of results here that include both the *Speed-based* and *Monolithic* reference methods. They highlight different research aspects and insights of performance comparison across all the three methods under comparison. Going forward, in the remaining three sections covering evaluation results, we opted to focus more extensively on the comparison to *Monolithic*, as this method represents the method of choice used in practical deployments today, and the investigations in these latter sections cover topics that practical implementations would need to carefully consider, such as the choice of the tiles' size, the impact of the quality evaluation metric, and the variability of quality across the delivered 360° video panorama.

E. Synthetic Navigation Traces

We generated 500 synthetic traces for each 360° video considered in our experiments, using the first order Markov model that we formulated earlier in Section III-C. In order to have similar user navigation patterns and tile navigation likelihoods, across the two cases (synthetic and real traces), we developed our navigation model to capture the related characteristics of the actual data as closely as possible. In this section, we investigate a sample synthetic trace and a sample real trace, for the 360° video Roller Coaster, and compare their resemblance. In particular, Figures 20 and 21 respectively show the synthetic trace and the real trace in the 2D equirectangular plane of the 360° video panorama. It can be observed that both navigation traces exhibit analogous spatial movements save for the model-induced characteristic of the synthetic trace that exhibits more discrete movements that are following the macroblocks of the spatial 360° panorama.

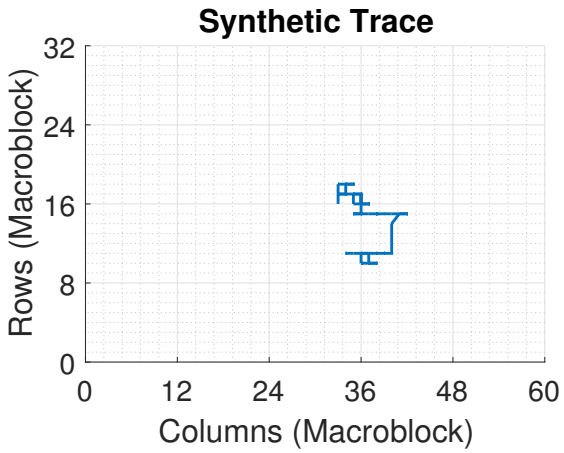


Fig. 20: Synthetic trace for Roller Coaster.

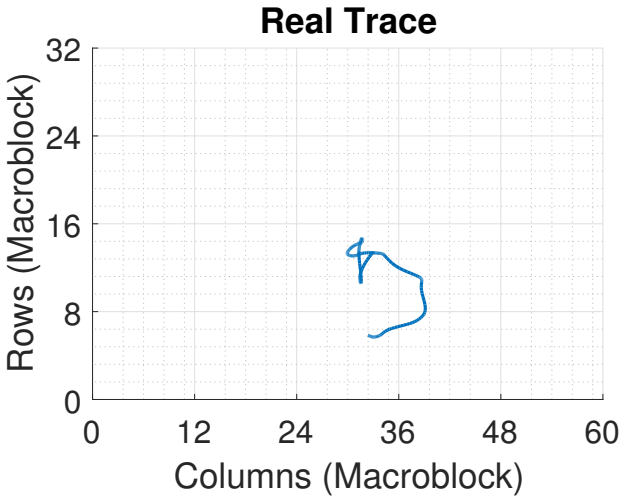


Fig. 21: Real data trace for Roller Coaster.

We also examine the navigation likelihoods for a 360° video generated from synthetic traces. In Figure 22, we

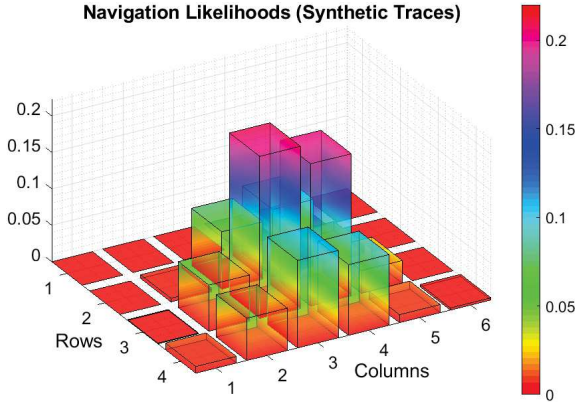


Fig. 22: Navigation likelihoods of tiles for Roller Coaster (synthetic traces).

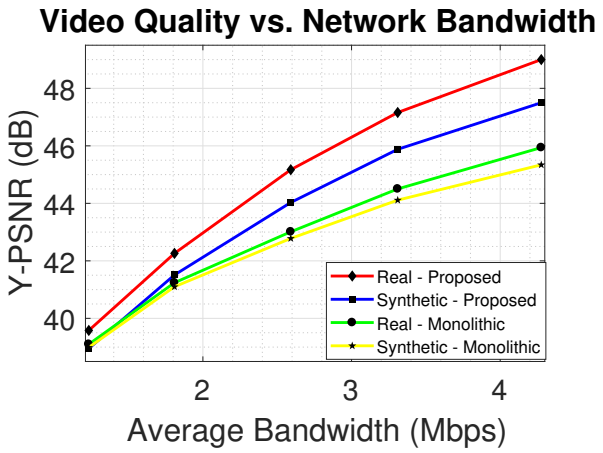


Fig. 23: Delivered viewport quality performance for real and synthetic traces.

examine these quantities in the case of the 360° video Roller Coaster. When we compare these navigation likelihoods to their counterparts from Figure 7, generated based on actual navigation traces captured from VR users, we can observe that they fairly accurately reproduce the latter, in terms of both absolute values as well as relative values that the latter exhibit among them, with some minor differences in the magnitude of the distribution of navigation likelihoods across the 360° video panorama. Thus, using synthetic traces to supplement actual traces can maintain the same navigation behavior probabilistically. These outcomes and observations merit the benefits of using the proposed navigation model and the synthetic navigation traces it can be used to generate, to supplement existing navigation traces with the objective to enhance the efficiency of the proposed optimization framework and 360° video streaming system.

We supplement the above investigation with a brief study of the validity of the Markovian nature of the proposed navigation model. In particular, we compute the expected Kullback–Leibler (KL) distance between two instances of the distribution $P_{m,n}^{j,r}$, computed from actual navigation traces conditioned on two different prior action values, $j = j_1$ and $j = j_2$, respectively. Formally, the KL distance or divergence between two statistical distributions P and Q establishes the degree of their dissimilarity, and is defined as $D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \ln(P(x)/Q(x))$, where $\mathcal{X}$ denotes the set of possible outcomes for the random variable x. Since $D_{KL}(\cdot||\cdot)$ is asymmetric as a measure, we define the distance between $P_{m,n}^{j_1,r}$ and $P_{m,n}^{j_2,r}$ using its symmetrized counterpart, $D_{KL}^*(P_{m,n}^{j_1,r}||P_{m,n}^{j_2,r}) = (D_{KL}(P_{m,n}^{j_1,r}||P_{m,n}^{j_2,r}) + D_{KL}(P_{m,n}^{j_2,r}||P_{m,n}^{j_1,r}))/2$. The expected value of $D_{KL}^*(P_{m,n}^{j_1,r}||P_{m,n}^{j_2,r})$ that we computed in the above case is approximately 0.09, which is considered significant and indicates that the distribution $P_{m,n}^{j,r}$ can be quite distinct depending on the value of the parameter j. In addition, we compute the expected distance D_{KL}^* between the distribution $P_{m,n}^{j,r}$ computed from actual and synthetic navigation traces, respectively. This quantity is only 0.2% and establishes the close statistical nature between the actual and synthetic traces.

Finally, when we investigate the delivered viewport quality performance in each case (real and synthetic trace), for the *Proposed* and *Monolithic* methods, we can see that under a real trace, our framework enables a higher expected performance gain over *Monolitic* across the entire range of network streaming bandwidth values considered on the x-axis of Figure 23. On the other hand, though *Monolithic* exhibits only a marginal performance drop in the case of a synthetic trace that does not exceed 0.5 dB at best, *Proposed* exhibits a somewhat more significant performance drop here which at the high end of network bandwidth values considered in Figure 23 reaches a 1 dB. This can be explained by the observation that more accurate viewport prediction can be carried out based on real traces. In particular, since actual head movements typically follow the dynamics of the 360° video content, navigating users tend to make similar head movements across the same 360° video frames. On the other hand, as each synthetic trace is randomly seeded at start, this aspect is not captured well by synthetic traces (lack of significant correlation across traces at different temporal points), and this can be a point of further investigation in future work, to improve the modeling and the closeness to real life of the synthetic traces generated from the model. For instance, generating multiple traces simultaneously and introducing further statistical dependencies across them as part of our modeling could be one direction to pursue in this regard. Moreover, it should be noted that though the navigation likelihoods associated with synthetic traces achieve a fairly accurate reproduction of those associated with actual traces (see Figure 22), there are still some minor differences between them, especially with respect to the magnitude of some lower valued likelihoods on the 360° video panorama, which may also contribute to a bigger performance difference between the two cases in terms of delivered viewport quality when the available network bandwidth considered by our optimization framework is bigger, as observed from Figure 23. We believe the latter outcome could be improved by having more actual traces to train the proposed first order Markov navigation that is used to generate the synthetic traces. Lastly, and at the same time, the higher performance gains exhibited by *Proposed* in the case of real navigation traces imply that having a bigger real dataset for dynamic tile navigation likelihood assignment will be expected to enable better navigation prediction and thus higher delivered viewport quality and quality of experience for

the user.

We note for the convenience and recall of the reader that only some results presented in this section are based on a synthetic trace. The rest of the results presented throughout Section VI are obtained based on actual navigation traces.

F. Tile Size Choice

Different tile sizes used in partitioning the panorama have a direct effect on optimization since they alter the assigned navigation likelihoods which affect the encoding gains. 2D projected viewport has an arbitrary shape as seen in Figures 10 and 11 and therefore the tiles on the fringe of the viewport end up having a residual area outside the viewport but encoded in high quality. This residual area results in suboptimal use of bandwidth resources especially in the case of larger tiles. Using finer tiles on the other hand, makes it easier to have higher quality around the viewport with minimizing the residual areas that would be out of the viewport. It, on the other hand, can result in tiles with likelihood zero assigned to be closer to the expected viewport and increases the possibility of sudden quality drops.

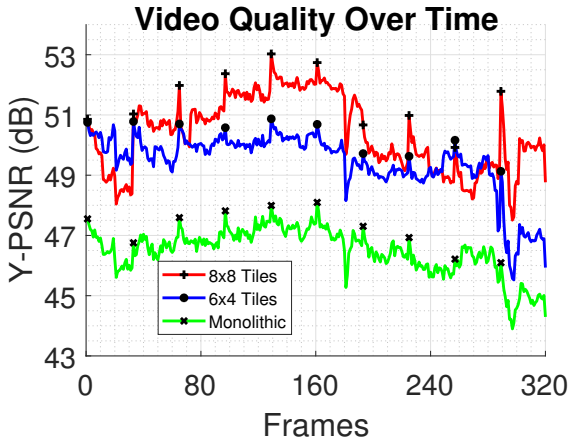


Fig. 24: 360° viewport quality for different tiling: Elephant 360° video.

Figure 24 shows the Y-PSNR differences of the Elephant video using small *8x8 Tiles*, large *6x4 Tiles* and the corresponding *Monolithic* case. *8x8 Tiles* case has higher Y-PSNR values in average due to better utilization of bandwidth in a smaller expected viewport area. One disadvantage of using small tiles is having more sudden and frequent drops. Having a smaller expected viewport area (cf. Figures 7 and 9) allows higher quality levels assigned to the most popular viewport tiles and an average rise in quality, while it has very low or zero quality assigned to some nearby but less popular tiles. As a result there are more occasions of sudden quality drops are observed in smaller tile case.

When we investigate the average quality levels, difference between *6x4 Tiles* and *8x8 Tiles* is still observed even though total Y-PSNR is decreased. In the case of scarce resources, having a larger expected viewport area rapidly decreases expected viewport quality. In minimum bandwidth levels both of the proposed methods are doing worse than *Monolithic* due to poor viewport prediction.

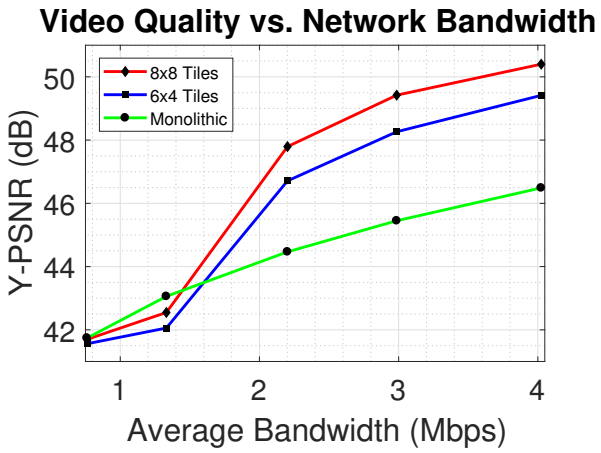


Fig. 25: Average 360° viewport tile comparison: Elephant.

G. 360° Video Quality Evaluation Metrics

In addition to the viewport Y-PSNR video quality evaluation and comparison that we carry out here, we also investigate the Multiscale - Structural Similarity Metric (MS-SSIM) [44] applied to the delivered viewport video signal (its luminance component) by each of the three methods under comparison, in the case of the 360° video Roller Coaster. In particular, in Figure 26, we compare the expected viewport quality delivered to a VR client, measured through both metrics, for each of the *Proposed*, *Speed-based*, and Monolithic methods, as a function of the network streaming bandwidth available to compress and deliver the 360° video content. It should be noted that under all three methods, the viewport's quality measured via MS-SSIM and Y-PSNR exhibits closely similar characteristics. One difference worth noting is that particularly for higher network bandwidth values examined in Figure 26, the viewport quality measured via the Y-PSNR metric shows better improvement and higher gains, since the blurred images used in the evaluation of the MS-SSIM metric exhibit higher similarity when the difference of content details become subtler. We also note that subjective metrics have been widely explored for measuring the QoE. However, although they provide inherently more human-oriented results, they are highly resource-consuming to evaluate. Moreover, according to [45], subjective metrics such as the Mean Opinion Score (MOS) exhibit very similar QoE results and characteristics with the Peak Signal to Noise Ratio (PSNR) metric in traditional wireless video streaming.

H. Spatial Viewport Quality Variation

Spatial quality variation is one of the important metrics in viewport quality. Having inconsistent spatial quality with high and low quality regions in a viewport is not a desired case and severely affect the user QoE. Exploiting unequal rate-distortion characteristics is expected to yield a smoother quality variation over viewport by assigning uniform distortion levels to viewport tiles. Figure 27 compares the CDF of standard deviation of per macroblock Y-PSNR in viewport of Diving video over all frames. *Proposed* approach has an average 2.5 dB standard deviation of Y-PSNR whereas the *Monolithic* approach has 5.2 dB. In addition, steeper curve

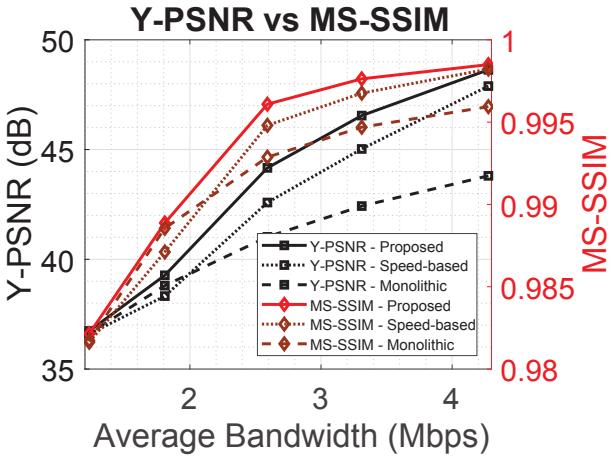


Fig. 26: Viewport video quality: A comparison of the Y-PSNR and Y-MS-SSIM metrics for the Roller Coaster 360° video.

of *Proposed* implies that rate-distortion optimization allows a more steady and limited spatial Y-PSNR distribution.

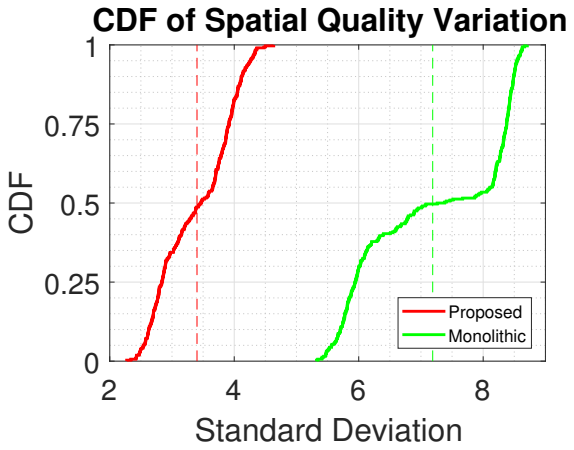


Fig. 27: CDF of viewport spatial quality variation: Diving.

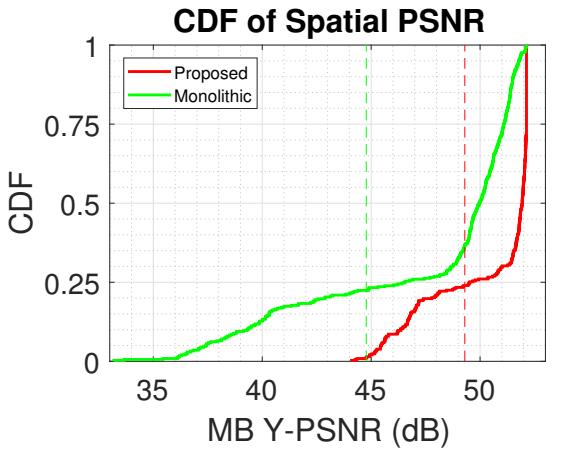


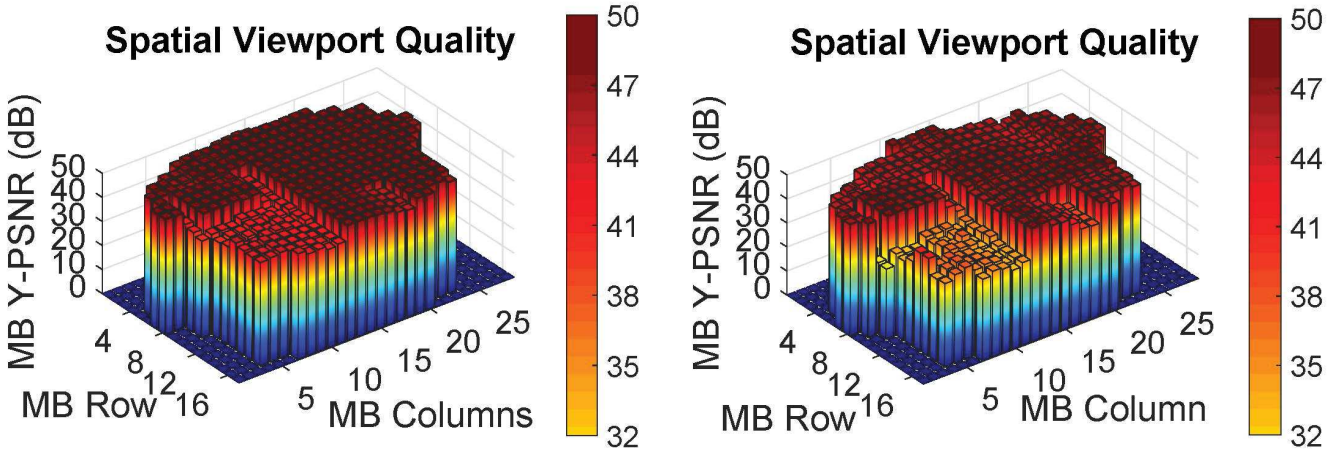
Fig. 28: CDF of viewport PSNR: Diving.

When we examine the distribution of per macroblock Y-PSNR over the user viewport we see that *Proposed* case has a much smaller span (Figure 28). Lower Y-PSNR values are observed less frequently in *Proposed* case and higher Y-PSNR values has a steeper curve. Figure 29 compares the per macroblock Y-PSNR values of viewport in their respective positions. It is apparent that both of the cases have very similar outlines. This outline of the quality distribution is caused by the video content. We observe an overall quality increase over the viewport due to better bandwidth utilization and a smoother Y-PSNR distribution caused by exploiting the rate-distortion characteristics to a uniform distortion level (Figure 29a). High quality regions of *Monolithic* case (upper half and center) are all increased the same 48 dB Y-PSNR value in *Proposed* case. Moreover, seemingly lower quality regions in *Proposed* approach (lower left and lower right) has higher and smoother quality than the *Monolithic* case.

VII. Conclusion

We have formulated a framework for viewport-driven rate optimized 360° video streaming that integrates the user view navigation patterns and the spatiotemporal rate-distortion characteristics of the 360° video content to maximize the delivered user quality of experience for the given network/system resources. Our framework comprises a methodology for computing dynamic navigation likelihoods that capture the user likelihood of navigating different spatial sectors of a 360° video over time, an analysis and characterization of its spatiotemporal rate-distortion characteristics that leverages pre-processed spatial tilling of the 360° video panorama, and an optimization problem formulation that characterizes the delivered expected user viewport video quality, given the user navigation patterns, 360° video encoding decisions, and the available system/network resources. Moreover, we have formulated a user navigation Markov model to analyze the user navigation actions in greater detail and extend our navigation dataset. Our experimental results demonstrate the advantages of our framework over the conventional approach of streaming a monolithic uniformly-encoded 360° video and a state-of-the-art navigation-speed based reference method, enabling considerable video quality of gains up to 5 dB in the case of five popular 4K 360° videos. In addition, the proposed framework achieves a substantially lower spatial video quality variation in the delivered user viewport, compared to monolithic 360° streaming, due to the optimization problem formulation we introduce that implicitly aims for a minimum uniform expected user viewport spatial distortion. We also investigated performance trade-offs associated with selecting the tile size for 360° equirectangular panorama spatial tiling/partitioning. Our experiments show that using finer tiles instead of large tiles utilizes the available network bandwidth more efficiently given enough user navigation history is available. Finally, we explored the impact of two different popular 360° video quality metrics applied to evaluate the streaming performance of our system framework and the two reference methods.

There are multiple directions of future work that we consider. In the present framework, we used two tiling scenarios of the 360° view panorama. Will variable-size 360° tiling provide additional gains, and at what cost, is one question we will aim to investigate. Wireless HMD devices allow users to enjoy VR without the inconvenience of cables yet lack high



(a) *Proposed* (b) *Monolithic*

Fig. 29: Spatial variation of the delivered viewport quality for the Diving 360° video.

computational capacity and require a high throughput wireless connection. Addressing how rate-distortion optimization will affect untethered VR communication is another study we plan to carry out in this context.

REFERENCES

[1] Grand View Research, "Virtual reality market size, share & analysis report, 2020-2027," Jun. 2020. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market

[2] J. G. Apostolopoulos, P. A. Chou, B. Culbertson, T. Kalker, M. D. Trott, and S. Wee, "The road to immersive communication," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 974–990, Apr. 2012.

[3] J. Chakareski, M. Khan, and M. Yuksel, "Towards enabling next generation societal virtual reality applications for virtual human teleportation," *IEEE Signal Processing Magazine*, Sep. 2022.

[4] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–7.

[5] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, Fukuoka, Japan, Sep. 2015, pp. 31–36.

[6] MPEG-DASH-OMAF standard: ISO/IEC FDIS 23090-2, "Omnidirectional Media Format," Apr. 2018.

[7] "Facebook 360: A stunning and captivating way to share immersive stories, places and experiences." [Online]. Available: http://facebook360.fb.com

[8] "YouTube: 360° Videos." [Online]. Available: https://www.youtube.com/

[9] B. Begole, "Why the Internet pipes will burst when virtual reality takes off," Forbes Magazine, Feb. 2016.

[10] E. Knightly, "Scaling Wi-Fi for next generation transformative applications," Keynote Presentation, IEEE International Conference on Computer Communications (INFOCOM), Atlanta, GA, USA, May 2017.

[11] J. D. Moss and E. R. Muth, "Characteristics of headmounted displays and their effects on simulator sickness," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53, no. 3, pp. 308–319, Jun. 2011.

[12] F. D. Simone, P. Frossard, C. Brown, N. Birkbeck, and B. Adsumilli, "Omnidirectional video communications: new challenges for the quality assessment community," *VQEG eLetter*, vol. 3, no. 1, pp. 18–24, Nov 2017.

[13] S. Afzal, J. Chen, and K. K. Ramakrishnan, "Characterization of 360-degree videos," in *Proc. SIGCOMM Workshop on Virtual Reality and Augmented Reality Network*. Los Angeles, CA, USA: ACM, Aug. 2017, pp. 1–6.

[14] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[15] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360° video streaming using tiles for virtual reality," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Beijing, China, Sep. 2017, pp. 2174–2178.

[16] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "Improving virtual reality streaming using HTTP/2," in *Proc. ACM on Multimedia Systems Conference (MMSys)*. Taipei, Taiwan: ACM, Jun. 2017, pp. 225–228.

[17] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. Workshop on All Things Cellular: Operations, Applications and Challenges*. New York City, New York, USA: ACM, Oct. 2016, pp. 1–6.

[18] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proc. IEEE International Symposium on Multimedia (ISM)*, San Jose, CA, USA, Dec. 2016, pp. 107–110.

[19] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "Optile: Toward optimal tiling in 360-degree video streaming," in *Proc. ACM on Multimedia Conference (MM)*, Mountain View, CA, USA, 2017, pp. 708–716.

[20] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation," in *Proc. ACM on Multimedia Systems Conference (MMSys)*, Taipei, Taiwan, Jun. 2017, pp. 261–271.

[21] M. Ben Yahia, Y. Le Louedec, L. Nuyami, and G. Simon, "When HTTP/2 rescues DASH: Video frame multiplexing," in *Proc. IEEE INFOCOM Workshop on Communication and Networking Techniques for Contemporary Video*, Atlanta, GA, USA, May 2017.

[22] 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. ACM on Multimedia Conference (MM)*, Mountain View, CA, USA, 2017, pp. 315–323.

[23] R. Aksu, J. Chakareski, and V. Swaminathan, "Viewport-driven rate-distortion optimized scalable live 360° video network multicast," in *Proc. ICME Int'l Workshop on Hot Topics in 3D (Hot3D)*. San Diego, CA, USA: IEEE, Jul. 2018.

[24] 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, Apr. 2021.

[25] L. Sun, Y. Mao, T. Zong, Y. Liu, and Y. Wang, "Live 360 degree video delivery based on user collaboration in a streaming flock," *IEEE Trans. Multimedia*, Feb. 2022.

[26] O. Eltobgy, O. Arafa, and M. Hefeeda, "Mobile streaming of live 360-degree videos," *IEEE Trans. Multimedia*, vol. 22, no. 12, pp. 3139—-3152, Dec. 2020.

[27] J. Chakareski, "UAV-IoT for next generation virtual reality," *IEEE Trans. Image Processing*, vol. 28, no. 12, pp. 5977–5990, Dec. 2019.

[28] ——, "Viewport-adaptive scalable multi-user virtual reality mobile-edge

streaming," *IEEE Trans. Image Processing*, vol. 29, no. 1, pp. 6330–6342, Dec. 2020.

[29] J. Chakareski, M. Khan, T. Ropitault, and S. Blandino, "Millimeter wave and free-space-optics for future dual-connectivity 6DOF mobile multi-user VR streaming," *ACM Trans. Multimedia Computing Communications and Applications*, May 2022, accepted.

[30] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360° video streaming," in *Proc. IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018.

[31] Y. Sanchez, R. Skupin, and T. Schierl, "Compressed domain video processing for tile based panoramic streaming using HEVC," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 2244–2248.

[32] R. Monnier, R. van Brandenburg, and R. Koenen, "Streaming UHD-quality VR at realistic bitrates: Mission impossible?" White Paper, TiledMedia, May 2017.

[33] M. Khan and J. Chakareski, "Visible light communication for next generation untethered virtual reality systems," in *Proc. Int'l Conf. Communications Workshop on Optical Wireless Communications*. Shanghai, China: IEEE, May 2019.

[34] J. Chakareski, R. Aksu, V. Swaminathan, and M. Zink, "Full UHD 360-degree video dataset and modeling of rate-distortion characteristics and head movement navigation," in *Proc. Multimedia Systems Conf.* Istanbul, Turkey: ACM, Sep. 2021, pp. 267–273.

[35] X. Corbillon, F. De Simone, and G. Simon, "360-degreee video head movement dataset," in *Proc. ACM on Multimedia Systems Conference (MMSys)*, Taipei, Taiwan, Jun. 2017, pp. 199–204.

[36] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How do people explore virtual environments?" *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 2018.

[37] "Mega coaster: Get ready for the drop (360 video)." [Online]. Available: https://youtu.be/-xNN-bJQ4vI

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[39] "Wingsuit 360 degree video over Dubai." [Online]. Available: https://youtu.be/AX4hWfyHr5g

[40] "Opentrack: Head tracking software." [Online]. Available: https://github.com/opentrack/opentrack

[41] "Elephants on the brink (360 video)." [Online]. Available: https://youtu.be/2bpICIClAIg

[42] "NYC 360 time-lapse (360 video)." [Online]. Available: https://youtu.be/CIw8R8thnm8

[43] "Scuba Diving Short Film in 360° Green Island, Taiwan 4K Video Quality." [Online]. Available: https://youtu.be/2OzlksZBTiA

[44] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Pacific Grove, CA, USA, Nov. 2003, pp. 398–1402.

[45] K. Piamrat, C. Viho, J. Bonnin, and A. Ksentini, "Quality of experience measurements for video streaming over wireless networks," in *Proc. Int'l Conf. Information Technology: New Generations*, Las Vegas, NV, USA, Apr. 2009, pp. 1184–1189.