Navigation-Aware Adaptive Streaming Strategies for Omnidirectional Video

Silvia Rossi and Laura Toni
Department of Electronic & Electrical Engineering, University College London, London, UK
{s.rossi, l.toni}@ucl.ac.uk

***Abstract*—Virtual reality (VR) applications target high-quality and zero-latency scene navigation to provide users with a full-immersion sensation within a scene. From a network perspective, this requires transmission of the omnidirectional content in its entirety, at a high resolution, which is not always feasible in bandwidth-limited networks. In this work, we propose an optimal transmission strategy for virtual reality applications able to fulfill the bandwidth requirements, while optimizing the end-user quality experienced in the navigation. In further detail, we consider a tile-based coded content for adaptive streaming systems, and we propose a navigation-aware transmission strategy at the client-side (i.e., adaptation logic), which is able to optimize the rate at which each tile is downloaded. First, we introduce the viewport-quality as metric that reflects the quality of any portion of the sphere displayed by the end-user. Then, we cast the tile-rate optimization as an integer linear programming problem and show that the proposed solution achieves substantial quality gains when compared to state-of-the-art adaptation logic methods.**

I. Introduction

Nowadays, the multimedia format for Virtual Reality (VR) applications is based on omnidirectional content, where a 360° scene is acquired instantaneously by an omnidirectional camera. The immersive sensation typical of VR is provided by placing the user at the center of the sphere and dynamically altering the portion of spherical content on display (*viewport*) according to the head direction of the user. This interaction of the user with the scene has created novel challenges from both coding and transmission perspectives. For instance, in classical video streaming, the entire scene is delivered and displayed at the user side, while in VR applications, users only consume a portion of the content in a highly dynamic way. Such a dynamic behavior has posed novel questions on how to most efficiently utilize the available network resources. In particular, transmission of the entire panorama, even if only a small portion of it is actually displayed, guarantees zero latency for the user when switching viewing direction. However, this comes at the price of a poor quality, being the panorama sent at low quality for poor channel resources. A more efficient usage of bandwidth would be to exclusively send the viewport of interest. However, the viewport needs to be prefetched in advance, when the viewport requested by the user is not known yet but rather predicted. An erroneous prediction of the displayed viewport would require a re-transmission of a new predicted viewport, leading to large switching delays. Therefore, it is essential to seek the correct streaming strategy able to find the optimal trade-off between bandwidth efficiency, quality and latency, since the way in which users consume videos while navigating is highly dynamic and uncertain. In this work, we propose a novel transmission strategy able to directly address this trade-off in the case of HTTP adaptive streaming (HAS) systems i.e., Dynamic Adaptive Streaming over HTTP (DASH) [1].

HAS systems offer users the possibility to adaptively select different versions (i.e. different coding rates and resolutions) of video streams that have been pre-encoded and stored at the content distribution server. Based on the experienced channel, each media client optimizes the appropriate version in order to maximize the video quality experienced, while navigating the scene. Along this direction, initial steps have been made with the study of adaptive streaming strategies for 360° content [2]–[5]. The work in [2] optimizes DASH systems for omnidirectional content, but focuses mainly on the server side of the chain. Namely, the optimal storage strategy is investigated in the case that the panorama representations are encoded over unequal quality levels (i.e. an area with high quality and the rest of the panorama at a low quality). A more formal tile-based DASH system is presented in [3], [5], where the new extension of DASH, defined as Spatial Relationship Description (SRD), is applied to the 360° video sequences. Both works focus on algorithms to generate tiles on the sphere, but crucially, no optimal strategy to select the optimal tile-representation at the client side is proposed. A tile-based adaptive streaming method is proposed in [4], where each user receives only the tiles that overlap with the predicted display viewport. This strategy, while effective from both a bandwidth and quality perspective, strongly depends on the viewport prediction. An erroneous estimate would immediately lead to re-transmissions, and hence, a possible stall or quality reduction effect. In summary, a formal adaptation logic method able to take into account particular individual factors such as scene geometry and user navigation, typical of omnidirectional content, is still missing.

This work proposes a navigation-aware adaptation strategy for 360° video adaptive streaming when sequence delivery is required for interactive users, aiming to provide a solution to the previously outlined challenge. In more details, we consider the scenario of 360° video sequences stored at the main server of the service provider (e.g., Netflix, YouTube). Each acquired

***MMSP'17**, October 16-18, 2017 - Luton (London), UK.*
978-1-5090-3649-3/17/$31.00 2017 European Union

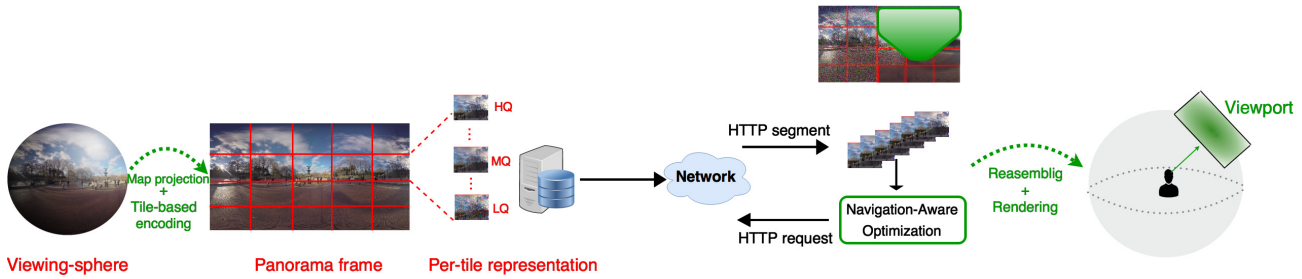


Figure 1. Overview of proposed architecture from viewing sphere to viewport display.

360° video is projected onto a plane called *panorama* and is then processed by a tile-based encoder. Each tile is encoded at a different coding rate and resolution, creating per-tile representations. Each representation is then decomposed into temporal chunks (usually $2s$ long) and stored at the server. Based on his own future navigation path, the client requests the best set of per-tile representations for the entire panorama. From the panorama, the viewport of interest is then rendered and displayed. Figure 1 depicts the considered scenario. The best set of representations downloaded by the user is defined as the one that (*i*) satisfies the channel bandwidth constraints and (*ii*) minimizes the distortion of the most likely displayed viewports, while also reducing the distortion variations along most likely navigation paths.

To achieve this goal, we evaluate the quality metric as a geometry based Mean Square Error (MSE) to consider not only the content characteristics (i.e., coding artifacts on the panorama) but also the scene geometry (i.e., the projection of portions of the panorama on the sphere). Next, we provide a formal problem formulation of the client adaptation logic and cast the problem as an integer linear programming (ILP) framework, which can be easily solved using the CPLEX solver. We further compare our adaptation logic strategy with the case of non-tile based coding. Simulation results show significant gains (in terms of navigation quality and smoothness) under different streaming scenarios. This reflects a more effective adaptation of the available network resources and furthermore, higher satisfaction experienced by the end-users. Finally, we compare the impact of different tile sizes on the final quality perceived by the user, showing that the optimal size depends on both the content characteristics as well as on the users interaction.

II. System Model

We now provide an overview of the navigation-aware adaptive streaming system proposed in this work. We first describe the structure of adaptive streaming systems, and then outline the key features of the omnidirectional content.

A. Adaptive Streaming over HTTP

In adaptive streaming systems, one video sequence is divided into chunks of fixed duration (typically $2s$), and the number of chunks into which each sequence is decomposed is denoted by K. In the case of omnidirectional sequences, each chunk consists of T panoramic frames. Each panoramic frame F_t with $t = 1, ..., T$ is decomposed into N regular blocks (or tiles). Each tile is encoded into Q per-tile representations, with coding rates defined by the following set of rates $\mathcal{R} = \{R_1, R_2, ..., R_Q\}$[1]. Without loss of generality, we assume the system at regime (no rump-up or re-buffering phase) in which one chunk is periodically downloaded every chunk duration. Therefore, while displaying a chunk, the client downloads the following one, asking for the set of representation resulting from the adaptation logic optimization. The adaptation logic optimizes the representation vector $\mathbf{r} = [r_1, r_2, .., r_N]$, where $r_n \in \mathcal{R}$ represents the coding rate for the n^{th} tile of chunk to download. The optimization resulting in the best set of representations to download for each chunk is what we propose in this paper. At each downloading opportunity, the user knows the popularity of each viewport to be displayed (*heatmap*) as well as the rate-distortion function for each tile-representation for the chunk of interest. This information can be periodically delivered to clients through the media presentation description, and it can reflect the information for each chunk or be averaged over a set of chunks (i.e., a trade-off between communication overhead and optimization accuracy). Equipped with this information, the optimization proposed in the following sections is invoked and the optimal chunk is requested for downloading.

B. Omnidirectional Video

We consider an acquired spherical video projected into rectangular panoramic frames (*map projection*) via an equirectangular projection,[2] since it is the simplest and most popular map projection [6]. In particular, a point on the viewing sphere can be mapped onto the panorama through longitude ($0 \leq \theta \leq 2\pi$) and latitude ($0 \leq \phi \leq \pi$) values. Then, each panoramic frame is processed by a tile-based encoder with uniform tiles. Therefore, panoramic frames are decomposed into blocks of area $S_b = \Delta\theta_b \Delta\phi_b$ where $\Delta\theta_b$ and $\Delta\phi_b$ are the longitudinal and latitudinal dimensions of each block. Because of the map projection, these blocks on the viewing sphere have unequal sizes. Each block can be seen as the sum of

[1] In this paper, we do not vary the encoded resolution across representations. However, our optimization problem can be directly extended to provide a solution in a scenario considering multiple resolutions.

[2] Note that the optimization problem proposed in this work is general enough to be extended to any other map projection method.

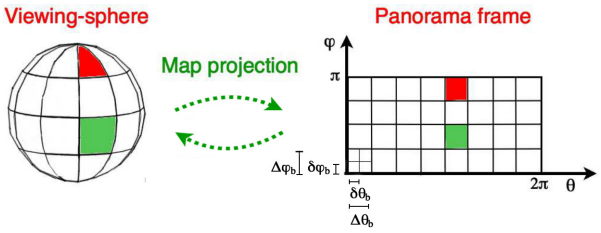


Figure 2. Map projection from the viewing sphere to the panorama image with $B = 2$.

infinitesimal elements with dimension $\delta\theta_b$ and $\delta\phi_b$. Therefore each block centred in (θ_b, ϕ_b) has an area on the sphere given by $\mathcal{S}_b = l^2 B^2 sin\phi_b \delta\theta_b \delta\phi_b$, where l is the radius of the sphere and B the number of infinitesimal elements per block [3]. In particular, the blocks near the poles are smaller than those in the equatorial zone, as can be inferred from Figure 2.

At the client side, any viewers, equipped with a head mounted device, navigates the 360° video by moving his head and changing the displayed viewport accordingly. The viewport is a plane tangent at the viewing sphere in the user's view direction (θ_i, ϕ_i), as shown in Figure 1. In particular, its longitudinal and vertical resolutions are imposed by the user's screen and denoted by $\Delta\theta_v$ and $\Delta\phi_v$, respectively. Considering the sphere with unitary ray ($l = 1$), we denote by $\mathcal{VP}_i$ the viewport with centre in (θ_i, ϕ_i) with $i = 1, ..., I$,[4] and its surface on the sphere is equal to:

$$\mathcal{S}_{\mathcal{V}_i} = \int_{\theta_i - \frac{\Delta\theta_v}{2}}^{\theta_i + \frac{\Delta\theta_v}{2}} \int_{\phi_i - \frac{\Delta\phi_v}{2}}^{\phi_i + \frac{\Delta\phi_v}{2}} \sin\phi d\theta d\phi \,. \tag{1}$$

Each viewport consists of a set of blocks (or tiles). Therefore, let us denote by $\mathcal{S}_{b_n}$, a surface on the sphere of the n^{th} block centered in (θ_n, ϕ_n), given by:

$$\mathcal{S}_{b_n} = \int_{\theta_n - \frac{\Delta\theta_b}{2}}^{\theta_n + \frac{\Delta\theta_b}{2}} \int_{\phi_n - \frac{\Delta\phi_b}{2}}^{\phi_n + \frac{\Delta\phi_b}{2}} \sin\phi d\theta d\phi \tag{2}$$

and $\alpha_{n,i}$ is the portion on the sphere of the n^{th} block overlapping with the viewport $\mathcal{VP}_i$.

III. Geometry-based QoE metric

We now define two quality metrics that describes the objective function in our optimization: (*i*) the *popularity-weighted geometry-based distortion*, i.e. the distortion of the different regions of the sphere associated to each possible viewport, weighted by the probability that the user selects that specific viewport, and (*ii*) the *navigation-smoothness*, i.e., the variation of the geometry-based distortion experienced during the navigation.

[3]We denote $\mathcal{S}$ for any given surface value on the sphere, while S represents any surface value on the panoramic image.

[4]We denote the total number of directions that we sample on the sphere y I. Ideally, $I \to \infty$, but in practice the head position is quantized.

A. Popularity-weighted geometry-based distortion

Firstly, we define the distortion experienced by the user while navigating in the 360° video and we highlight the differences with respect to the distortion of the decoded panoramic frame. In term of notation, in the following we adopt $\mathcal{D}$ to indicate any distortion values on the sphere and D to indicate the distortion on the panoramic image.

We assume that the distortion of a given viewport is measured by the distortion of the portion of the sphere which underpins the viewport. Therefore, the distortion of a generic viewport $\mathcal{VP}_i$ with its center in (θ_i, ϕ_i) is evaluated as:

$$\mathcal{D}_i = \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \int_{\theta_i - \frac{\Delta\theta_v}{2}}^{\theta_i + \frac{\Delta\theta_v}{2}} \int_{\phi_i - \frac{\Delta\phi_v}{2}}^{\phi_i + \frac{\Delta\phi_v}{2}} \mathcal{D}(\theta, \phi) \sin\phi d\theta d\phi \tag{3}$$

where $\mathcal{D}(\theta, \phi)$ is the distortion function at any point (θ, ϕ) on the viewing sphere. Decomposing the viewport into the different blocks derived from the tile-based coding, (3) can be reformulated as:

$$\mathcal{D}_i = \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n \in \mathcal{VP}_i} \int_{\tilde{\theta}_{i,n}^-}^{\tilde{\theta}_{i,n}^+} \int_{\tilde{\phi}_{i,n}^-}^{\tilde{\phi}_{i,n}^+} \mathcal{D}(\theta, \phi) \sin\phi d\theta d\phi \tag{4}$$

where $\tilde{\theta}_{i,n}^- = \min(\theta_i - \frac{\Delta\theta_v}{2}, \theta_n - \frac{\Delta\theta_n}{2})$, $\tilde{\theta}_{i,n}^+ = \min(\theta_i + \frac{\Delta\theta_v}{2}, \theta_n + \frac{\Delta\theta_n}{2})$, and similarly $\tilde{\phi}_{i,n}^- = \min(\theta_i - \frac{\Delta\theta_v}{2}, \theta_n - \frac{\Delta\theta_n}{2})$, $\tilde{\phi}_{i,n}^+ = \min(\theta_i + \frac{\Delta\theta_v}{2}, \theta_n + \frac{\Delta\theta_n}{2})$. Recalling that $\alpha_{n,i}$ is the percentage of block n that overlaps with the portion of the sphere underpinning viewport $\mathcal{VP}_i$, the previous equation can be generalized as follows:

$$\mathcal{D}_i = \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n=1}^{N} \alpha_{n,i} \int_{\theta_n - \frac{\Delta\theta_v}{2}}^{\theta_n + \frac{\Delta\theta_v}{2}} \int_{\phi_n - \frac{\Delta\phi_v}{2}}^{\phi_n + \frac{\Delta\phi_v}{2}} \mathcal{D}(\theta, \phi) \sin\phi d\theta d\phi \tag{5}$$

where the summation has been extended to all blocks within the panorama. All pixels on the sphere in the range $\{[\theta_n - \frac{\Delta\theta_v}{2}, \theta_n + \frac{\Delta\theta_v}{2}], [\phi_n - \frac{\Delta\phi_v}{2}, \phi_n + \frac{\Delta\phi_v}{2}]\}$ belong to the block n on the panorama, which has been encoded at the same rate for each representation level. The representation encoded at rate r_n will lead to a distortion averaged over the block denoted by $D_n(r_n)$. From this consideration as well as from (2), the distortion of viewport $\mathcal{VP}_i$ is given by

$$\begin{aligned}
\mathcal{D}_i(\mathbf{r}) &= \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n=1}^{N} D_n(r_n) \alpha_{n,i} \int_{\theta_n - \frac{\Delta\theta_v}{2}}^{\theta_n + \frac{\Delta\theta_v}{2}} \int_{\phi_n - \frac{\Delta\phi_v}{2}}^{\phi_n + \frac{\Delta\phi_v}{2}} \sin\phi d\theta d\phi \\
&= \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n=1}^{N} D_n(r_n) \alpha_{n,i} \mathcal{S}_{b_n} \\
&= \sum_{n=1}^{N} D_n(r_n) \alpha_{n,i} \widehat{\mathcal{S}}_{n,i}
\end{aligned} \tag{6}$$

where $\widehat{\mathcal{S}}_{n,i} = \mathcal{S}_{b_n} / \mathcal{S}_{\mathcal{V}_i}$ is the block surface on the sphere normalized by the area of the viewport, and where we explicitly show the dependency of $\mathcal{D}_i$ on $\mathbf{r}$.

The probability for the user to display viewport $\mathcal{VP}_i$ in the panoramic frame t is denoted by $p_{t,i}$, and hence, the

popularity-weighted distortion of the chunk to be downloaded is:

$$\mathcal{D}(\mathbf{r}) = \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{n=1}^{N} D_n(r_n)\widehat{\mathcal{S}}_{n,i}\alpha_{n,i}p_{t,i}. \quad (7)$$

It is worth noting that the rate-distortion on the panorama block $D_n(r_n)$ does not depend on the time index t since $D_n(r_n)$ reflects the mean distortion of block n encoded at the coding rate r_n for all frames in the chunk.

B. Navigation-smoothness

Beyond the average quality experienced during the navigation, we are interested in evaluating the quality variation, since varying quality while changing viewport can result in an annoying degradation in quality of experience.

We first evaluate the distortion variation between two consecutive viewports displayed at time $t-1$ and t, when displaying viewport $\mathcal{VP}_i$ at time t. This is given by

$$\Delta\mathcal{D}_{t,i}(\mathbf{r}) = \sum_{j\in\mathcal{N}(i)} \left|\mathcal{D}_i(\mathbf{r}) - \mathcal{D}_j(\mathbf{r})\right| p_{t-1,j}$$

where $\mathcal{N}(i)$ is the set of viewports that could have been displayed at time $t-1$ and is defined as the set of viewports with center in (θ_j, ϕ_j) such that

$$\begin{cases} \theta_j \leq \theta_i \pm \theta_{head} \\ \phi_j \leq \phi_i \pm \phi_{head} \end{cases} \quad (8)$$

where θ_{head} and ϕ_{head} are the maximum angular movements of the human head between two consecutive frames.

The navigation-smoothness per chunk can then be evaluated as follows:

$$\Delta\mathcal{D}(\mathbf{r}) = \sum_{t=2}^{T}\sum_{i=1}^{I}\Delta\mathcal{D}_{t,i}(\mathbf{r})p_{t,i} \quad (9)$$

$$= \sum_{t}\sum_{i}\sum_{j\in\mathcal{N}(i)}\left|\sum_{n} D_n(r_n)(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j})\right| p_{t-1,j}p_{t,i}$$

IV. Navigation-Bandwidth Adaptive Logic

Equipped with the above metrics and notations, we can now formulate the optimization problem that needs to be solved at the client side at each downloading opportunity. In the following, we first formalize the optimization problem and we then describe the solving method.

A. Problem formulation

We seek the optimal set of representations for all blocks of the panoramic frames such that the quality experienced in the scene navigation is maximized and yet the bandwidth constraint is respected. We can then express the navigation-aware adaptation logic optimization for each chunk as:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \mathcal{D}_{user}(\mathbf{r}) \\ \text{s.t.} \quad & \sum_{n} r_n \leq C \end{aligned} \quad (10)$$

where C is the estimated channel capacity during the delivery of the chunk of interest and $\mathcal{D}_{user}(\mathbf{r})$ is the metric that takes into account both the geometry-based quality and the navigation-smoothness. In particular,

$$\begin{aligned} \mathcal{D}_{user}(\mathbf{r}) &= \mathcal{D}(\mathbf{r}) + \lambda\Delta\mathcal{D}(\mathbf{r}) \\ &= \sum_{t}\sum_{i}\left[\mathcal{D}_i(\mathbf{r}) + \lambda\Delta\mathcal{D}_{t,i}(\mathbf{r})\right]p_{t,i} \end{aligned} \quad (11)$$

where λ is the multiplier that allows us to assign the appropriate weight to quality in the objective metric. Parametrizing the rate-distortion function of the panorama blocks leads to the following [7]:

$$D_n(r_n) = a_n + \frac{b_n}{r_n + c_n} \quad (12)$$

and hence, the problem formulation in (10) becomes:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \mathcal{D}_{user}(\mathbf{r}) \\ s.t. \quad & \sum_{n}\frac{b_n}{D_n(r_n) - a_n} - c_n \leq C \end{aligned} \quad (13)$$

where a_n, b_n and c_n are constants that depend on the content characteristics of block n.

The above optimization problem is computationally complex to solve being $\mathcal{D}_{user}$ neither a convex nor a linear function. In the following, we show how to cast the problem in (13) in a tractable ILP optimization problem.

B. ILP Optimization Algorithm

We recall that the set of representations available for each block is finite and corresponds to a specific set of coding rates $\mathcal{R}$ used to store the representations at the server. It follows that, in the panoramic frame, the distortion of each block $D_n(r_n)$ can be expressed as:

$$D_n(r_n) = \sum_{q=1}^{Q} D_n(R_q)\beta_{n,q} \quad (14)$$

where $R_q \in \mathcal{R}$, and $\beta_{n,q} = 1$ if $r_n = R_q$, $\beta_{n,q} = 0$ otherwise. This means that rather than seeking the best coding rate $\{r_n\}_n$ for all blocks in the panorama, we seek the best set of binary variables $\{\beta_{n,q}\}_{n,q}$. Adopting a change of variable $x_{n,q} \rightarrow D_n(R_q)$, the objective function becomes:

$$\begin{aligned} \mathcal{D}_{user}(\mathbf{r}) = \sum_{t=1}^{T}\sum_{i=1}^{I}\Bigg[& \sum_{n=1}^{N}\sum_{q=1}^{Q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} + \\ & \sum_{j\in\mathcal{N}(i)}\left|\sum_{n=1}^{N}\sum_{q=1}^{Q} x_{n,q}\beta_{n,q}(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j})\right| p_{t-1,j}\Bigg] p_{t,i} \end{aligned}$$

The previous expression is not linear because of the absolute value in the second term. However, an equivalent objective function linear in $\beta_{n,q}$ can be evaluated as shown in the following. We introduce an auxiliary variable y such that:

$$y = \sum_{n=1}^{N}\sum_{q=1}^{Q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}}_{n,i}(\alpha_{n,i} - \alpha_{n,j}) \quad (15)$$

Integer Linear Programming 1

$$\min_{\beta,\mathbf{y}} \quad \sum_t \sum_i \left[\sum_n \sum_q x_{n,q}\beta_{n,q}\widehat{\mathcal{S}_n}\alpha_{n,i} + \sum_{j\in\mathcal{N}(i)} y_{i,j}p_{t-1,j} \right] p_{t,i} \tag{17a}$$

$$s.t. \quad \sum_q \beta_{n,q} = 1, \forall n \in [1,N], \tag{17b}$$

$$\sum_n \sum_q \left(\frac{b_n}{x_{n,q} - a_n} - c_n \right) \beta_{n,q} \leq C \qquad \forall n \in [1,N] \tag{17c}$$

$$y_{i,j} \geq \sum_n \sum_q x_{n,q}\beta_{n,q}\widehat{\mathcal{S}_n}(\alpha_{n,i} - \alpha_{n,j}) \tag{17d}$$

$$\forall t \in [1,T], \forall i \in [1,I], \forall j \in \mathcal{N}(i), \forall n \in [1,N] \tag{17e}$$

$$y_{i,j} \geq -\left(\sum_n \sum_q x_{n,q}\beta_{n,q}\widehat{\mathcal{S}_n}(\alpha_{n,i} - \alpha_{n,j}) \right) \tag{17f}$$

$$\forall t \in [1,T], \forall i \in [1,I], \forall j \in \mathcal{N}(i), \forall n \in [1,N] \tag{17g}$$

The absolute value in (15) can then be obtained by imposing the two following constraints on the y variable:

$$y_{i,j} \geq \sum_n \sum_q x_{n,q}\beta_{n,q}(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j})$$

$$y_{i,j} \geq -\left(\sum_n \sum_q x_{n,q}\beta_{n,q}(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j}) \right).$$

Finally, the optimization problem in (10) can be casted as an ILP problem shown in (17). The objective function minimizes the expected quality experienced by the user when navigating the scene in the chunk duration. The constraint (17b) guarantees that only one representation is selected for each block in a chunk, while (17c) imposes the bandwidth constraint. Finally, the constraints (17e) and (17g) are the terms of transformation of the absolute value in a linear function.

V. Simulation Results

A. Simulation Setups

We consider two 360° videos, namely *"Rollercoaster"* and *"Timelapse NY"*. Both the sequences have been downloaded in equirectangular format at the maximum spatial resolution and frame rate available on the platform YouTube, i.e. 3840x2048 pixels and 30 fps, respectively [8]. The sequences have been selected due to the different spatial and temporal complexity of their content. In particular, *"Rollercoaster"* is more complex since it has a moving camera and its values of Spatial Information (SI) and Temporal Information (TI) equal to 72 and 45, respectively. On the contrary, *"Timelapse NY"* has a fixed camera that shoots city streets and its corresponding SI and TI values are 44 and 14, respectively.

To simulate a tile-based encoding, sequences have been split temporally and spatially in blocks. This results in a reduced coding efficiency with respect to a standard tile-based encoder. Therefore, the gain provided in following should be considered as lower bound to the actual gains, which can further improve in the case of more efficient tile-based coding strategies. We set a chunk of duration of about $2s$ and squared blocks with three different sizes, $L = [256, 512, 680]$ pixels. We then compare our optimized strategy with a baseline case in which the entire panorama is encoded (without tile-based encoding) at the same average rate. We label this baseline method by "Full Video" in the following results. Each block (as well as the entire panorama) has been encoded with HEVC codec [9] with an overall coding rate ranging between 16 kbps and 150 Mbps. We then consider 15 representations for each blocks ($Q = 15$). These representations are selected as the one corresponding to quality levels (in terms of PSNR) of $[25, 28, 29, 30, 31, 32, 33, 34, 35, 38, 40, 42, 45, 50, 52]$ dB. The rate value associated to each quality score has been derived by the rate-distortion function given in (12), where the parametric values are evaluated by curve fitting.

As input to our ILP problem, the prediction of user's navigation path in the 360° content is required. Using the free software Graph-Based Visual Saliency (GBVS) [10], the position of each focus of attention (FoA) could be computed for each panoramic frame. From this FoA map, we derived the heatmap over time. The two considered videos differ substantially in terms of resulting heatmap over time. The *"Rollercoaster"* sequence has one main FoA, which leads to a nicely predictable behavior of the users. On the contrary, *"Timelapse NY"* has several FoAs, increasing therefore the uncertainty of the interactivity behavior of the users. Finally, the selection of the most suitable set of representations-per-block is optimized with our ILP optimization problem in scenarios characterized by values of C ranging from 2 Mbps to 40 Mbps. Moreover, we assign a unitary weight to quality in the objective function of our problem ($\lambda = 1$). We have used the generic solver IBM ILOG CPLEX [11] to solve the ILP proposed in this work. Results in the following are provided both for the quality (in terms of PSNR) and for the navigation-smoothness (in terms of PSNR difference) and they

have been carried out by over 100 simulated interactive users downloading over a constant channel constraint over time. It is worth noting that our simulation considers some approximations (infinite playback buffers, exact channel estimation, etc.) with respect to real HAS systems. But these do not impact on our objective in this paper, which is to demonstrate the benefit of considering content and interactive information in the optimal representation selection for a HAS client in a stationary regime.

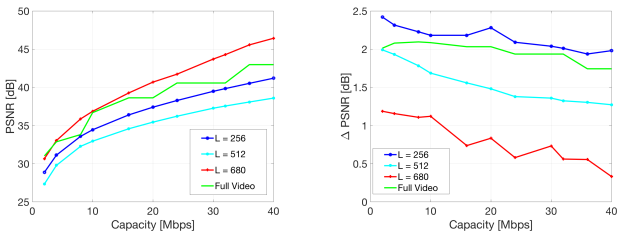
B. Results

In Figure 3, both the quality (in terms of PSNR) defined in (6) and the navigation-smoothness (in terms of PSNR difference) have been provided as a function of the available bandwidth, for the *"Rollercoaster"* video sequence. As expected, the quality increases with the available bandwidth, Figure 3(a). Most importantly, the proposed optimization with tile size $L = 680$ outperforms the Full Video case (with no tiling). This shows the gain of the added degree of freedom in the adaptation logic thanks to the tiling. However, by decreasing the tile size, this quality gain fades away. This is motivated by the fact that tiling leads to a more flexible transmission strategy, but at the price of a reduced coding efficiency. This tradeoff is overall good for $L = 680$ and not for $L = 512$ and $L = 256$. In particular, in this type of sequences in which the FoA is very narrow and uniform across users, there is no need of too much refined tiles (i.e., small values of L). Therefore, the loss in coding efficiency due to small L value is not necessarily balanced by the gain in the adaptation logic. A similar trend is observed for the smoothness-navigation, where $L = 680$ reduces the quality variations experienced during the navigation of the 100 randomly generated users.

A slightly different behaviour is observed in the case of the *"Timelapse NY"* sequence, Figure 4. For values of capacity bigger than 5 Mbps, each tiled solution achieves a better final quality than in the delivery of the entire encoded panorama. This is due to (i) different video characteristics that lead to a different penalty in coding efficiency, (ii) different navigation patterns of the interactive users. The distribution of FoA is far more variable then the case of *"Rollercoaster"* and misses a dominant area of interest. Therefore, higher resolution in optimizing the per-tile representation (small L values) balance the loss in coding efficiency and lead to a quality gain with respect to the no-tile case (Full Video). However, the quality variations observed in Figure 4(b) are highly random. This can be mainly justified by the fact that in the case of multiple FoA predicting the users navigation path only from the heatmap (as we assume in our problem formulation) is not enough reliable. This shows the need for an improved prediction model to be adopted in our representation optimization.

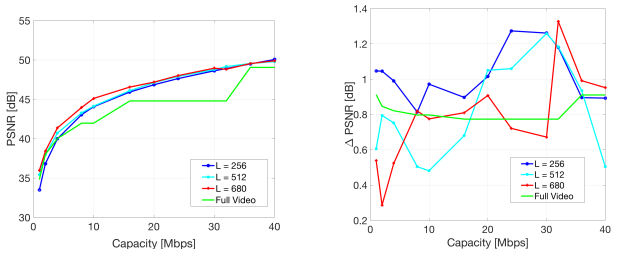
VI. Conclusion

In this paper, we have presented a novel navigation-aware strategy for 360° video adaptive streaming. In particular, we have proposed an adaptation logic at client side able to choose the best set of representations-per-block to download, in order to achieve an optimal final quality. We have evaluated the performance of our algorithm comparing the final quality of different tile sizes with the entire encoded video. Even if a visible gain in terms of navigation quality is provided, the results shows also to be strongly affected by the content of sequences and user navigation.



(a) Viewport Quality (b) Viewport Quality Variation

Figure 3. Analysis of Rollercoaster with $\lambda = 1$ and 100 users.



(a) Viewport Quality (b) Viewport Quality Variation

Figure 4. Analysis of Timelapse NY with $\lambda = 1$ and 100 users.

Acknowledgement

This work has been partially funded by the EPSRC Institutional Sponsorship funding OISTER no 536085.

References

[1] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[2] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," *arXiv preprint arXiv:1609.08042*, 2016.

[3] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using mpeg-dash," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 41.

[4] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[5] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer!" *arXiv preprint arXiv:1609.08729*, 2016.

[6] F. De Simone, P. Wilkins, N. Birkbeck, A. Kokaram, and P. Frossard, "Geometry-driven quantization for omnidirectional image coding," in *32nd Picture Coding Symposium (PCS 2016)*, 2016.

[7] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[8] Youtube. [Online]. Available: https://www.youtube.com

[9] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[10] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, 2007, pp. 545–552.

[11] IBM, "ILOG CPLEX optimization studio," 2013.