A Combined Field-of-View Prediction-assisted Viewport Adaptive Delivery Scheme for 360° Videos

Abid Yaqoob, *Student Member, IEEE*, Gabriel-Miro Muntean, *Senior Member, IEEE*

***Abstract*—Recently, 360° or omnidirectional videos have become increasingly popular for both personal and enterprise use-cases. However, 360° video streaming has very high bandwidth and processing requirements. State-of-the-art viewport-based streaming solutions lower these requirements by performing selective streaming based on long-term Field-of-View (FoV) prediction mechanisms. However, sometimes user movement is extremely unpredictable during some parts of the video, and applying these solutions adversely affects the overall quality of experience (QoE). This paper proposes a novel Combined Field-of-View tile-based adaptive streaming solution (CFOV) that improves end-user QoE for 360° video streaming. CFOV performs interactive tile selection based on more accurate dynamical viewing area identification by combining the results of two FoV prediction mechanisms. It also employs an innovative priority-based bitrate adaptation approach that ensures improved bitrate budget distribution between different tiles. We evaluate the proposed solution with a comprehensive set of experiments involving four immersive videos, diverse tiling patterns (i.e., 4x3, 6x4, and 8x6), different segment lengths (i.e., 1s, 2s, and 3s), and 48 empirical head movement traces under different bandwidth settings. The evaluation employs a newly defined QoE metric specifically introduced to assess the streaming performance of 360° videos objectively. The experimental findings show that, compared to alternative approaches, our proposed solution can achieve a higher viewport match and can significantly improve the user QoE for different watching behaviors and content characteristics.**

***Index terms*— 360° video streaming, tile-based adaptation, HTTP adaptive streaming, FoV prediction, QoE**

I. Introduction

OMNIDIRECTIONAL 360° video is rapidly moving towards the mainstream mainly due to the recent developments in computing, display, and networking technologies. Major commercial video streaming vendors (e.g., YouTube, Facebook, and Vimeo) promote panoramic nature content. With the increasing adoption of new and interactive 360° videos in virtual reality, gaming, and sports industry [1], mobile video traffic is projected to account for about 82% of global cellular traffic by 2022 [2]. 360° cameras are available for producing high-resolution video content. The stitching or post-processing software ensures the best content preparation. Modern head-mounted display (HMD) devices are equipped with powerful sensors and processing components for efficient display of 360° videos. However, this type of video transmission over existing IP networks is still very challenging, which stems from their larger size [3]. Moreover, the real-time handling of 360° content is highly time-sensitive because all the requested content has to be displayed in less than 20ms [4], [5] in response to the viewer head movements.

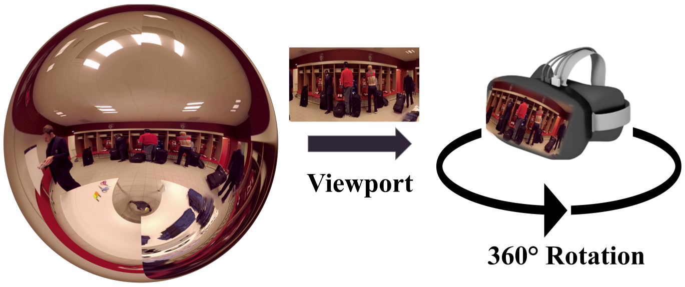


Fig. 1: Viewport-based 360° viewing arrangements for a spherical image retrieved from a sports video.

360° videos are similar to interactive applications, enabling its audience a look around the environment [6]. Fig. 1[1] illustrates the viewport-based visualization for virtual 360° environment. The usefulness of interactive video services is strictly dependent on managing bandwidth resources during playback time. Adaptive video transmission help support the user's appetite for improved streaming experience by dealing with both the content and network objectives, e.g., visual quality [7], [8], navigation [9], Region-of-Interest (RoI) [10], [11], energy consumption [12], [13], load balancing [14], etc. on mobile and fixed networks [15]. Compared to traditional adaptive streaming, 360° client needs to switch among different viewing regions according to the consumer's head movements and playback rate adaptation.

360° video streaming has progressed from viewport-independent streaming to viewport-dependent or tile-based streaming. Viewport-independent streaming is similar to traditional video streaming, where the playback adaptation for the whole segment is performed based on the estimated network capacity. However, a user is able to watch only a little part of the transmitted stream (e.g., 20%-30%) [16], known as the viewport, due to the visual limitations of both the human

A. Yaqoob is with the Performance Engineering Lab and Insight SFI Centre for Data Analytics, School of Electronic Engineering, Dublin City University, Ireland (e-mail: abid.yaqoob2@mail.dcu.ie).

G.-M. Muntean is with the Performance Engineering Lab, School of Electronic Engineering, Dublin City University (e-mail: gabriel.muntean@dcu.ie).

This work is funded by the European Regional Development Fund through the Science Foundation Ireland (SFI) Research Centres Programme under Grant Number SFI/12/RC/2289_P2 (Insight Centre for Data Analytics) and 16/SP/3804 (ENABLE). (Corresponding author: G.-M. Muntean.)

[1] The spherical image is retrieved from the *LOSC Football* video available at https://www.youtube.com/watch?v=lvH89OkkKQ8.

and display devices. Such streaming solutions are simple to implement; however, they result in substantial wastage of bandwidth resources [17]. Instead of transmitting the whole frames in higher quality, viewport-dependent streaming solutions allow certain frame areas' selective transmissions in accordance with the user's viewing orientation. Such solutions have lower bandwidth requirements; however, they are associated with very high storage and processing overheads [18]. Tile-based streaming is an extension of viewport-dependent streaming, where 360° frames are spatially cut into several rectangular grids, known as tiles [19]–[21]. The video tiles are then temporally split into several equal-duration segments to facilitate adaptive streaming. Both spatial and temporal adaptation is performed by leveraging human viewing behavior information. The client selects the visible and non-visible tiles and their quality levels based on the predicted viewport and the next segment's available connection speed. The tiles requests are performed in advance to ensure synchronous and timely playback of the content. Some straightforward tile-based solutions [22]–[24] adjust the quality based on the available viewport data. However, this is impractical in a real-time streaming scenario, as user actual and predicted viewing positions could be different [25]. Some solutions [26]–[29] have been proposed to stream non-visible tiles to avoid the black dots in the viewport, including some solutions [24], [30], [31] that assign the lowest resolution to the invisible tiles to save network bandwidth.

Although beneficial, tile-based adaptive solutions struggle to perform good viewport identification, synchronization with user head movements, bitrate adjustments, etc. Long-term accurate Field-of-View (FoV) prediction for the upcoming segments can support high-quality future media services [32]. However, it is highly error-prone and adds tremendous pressure on the prediction mechanisms. Delivering viewport tiles following inaccurate viewport prediction may significantly deteriorate user-perceived quality and reduce their satisfaction with the 360° video streaming service. Unfortunately, conventional bitrate adaptation heuristics [33]–[38] are not able to perform accurate content adjustment during tile-based streaming in the presence of highly variable and diverse factors (e.g., available bandwidth, user movement, segment sizes, etc.) or to make the best selection as the video segments are prepared in numerous tiles and encoding bitrates. Several existing tile-based adaptive streaming solutions either increase the viewport quality aggressively [27], [29], [39] or use a conservative approach [26], [30], [40] to maintain continuous video playback. However, this is not acceptable because the former approach will result in playback interruptions and wastage of the bandwidth, while the later policy will result in a poor streaming experience. Therefore, it is indispensable to maintain an important balance between bandwidth utilization and user-perceived quality. Moreover, considering visual quality only as a key assessment metric cannot ensure high-performance streaming. The multiple objective metrics such as maximizing viewport quality and minimizing background quality while also maintaining the inter- and intra-viewport smoothness play a significant role in optimizing the adaptive 360° video distribution.

In order to overcome the limitations of existing solutions, this paper introduces ***a combined FoV prediction-assisted 360° video streaming approach (CFOV)***. In contrast to existing schemes, CFOV employs two FoV prediction mechanisms to reduce the impact of unpredictable user movements noted on different videos. The proposed solution is capable to dynamically perform tile selection and bitrate adjustment for each adaptation interval during 360° video streaming. To be more specific, the CFOV client systematically decides best-fit tiles for each segment, considering the fact that the viewer can change the view at any time during the playback. Then, CFOV renders the selected tiles at the best possible quality to reach the optimization goal. Unlike previous solutions, CFOV implements an aggressive priority-related weighted quality adjustment for the tiles belonging to different regions based on exploration and exploitation of environmental variables such as viewing areas, tiles distribution, and connection speed.

The main contributions of this paper are as follows:

1) It introduces a practical-oriented tile selection method for 360° videos, which lowers the impact of fast head movement. This method defines the user viewport in terms of a 110° viewing space in both horizontal and vertical directions and employs a combination of two viewport prediction mechanisms.
2) It proposes a new adaptation algorithm, which actively allocates the video bitrate budget to different video frame areas to maximize the VR perception levels. The benefit of the proposed method is assessed using videos with different levels of motion content and with different segment durations.
3) It presents extensive trace-driven simulations using real head motion traces of 48 VR users, with different content types, tiling patterns, segment durations, and dynamic bandwidth variations. Experimental results reveal that the proposed CFOV solution significantly improves the streaming performance compared to existing tile-based streaming approaches. For instance, CFOV provides an improvement between 12.74%-21.5% in terms of average QoE under different testing settings.

Paper Organization: The paper is organized as follows: several existing field-of-view prediction solutions are described next, along with 360° tile-based streaming solutions. An illustration of the CFOV architecture follows the presentation of the proposed system design. The experimental testing setup and comparative evaluation with different other approaches are presented next. The last Section includes conclusive remarks and indicates possibilities for some potential future avenues.

II. Related Works

The latest unprecedented demand for 360° video content is mostly due to the associated immersive user experience. An integral part of current multimedia applications, 360° video distribution has drawn many researchers' attention. This Section discusses the latest related works in terms of streaming technologies, proposed frameworks, their main innovations, and possible limitations.

A. FoV Prediction

FoV prediction is considered a key player in the optimized streaming of 360° video. The latest wearable headsets allow the clients to refresh their scenes matching to their viewing positions. The FoV prediction approaches can be categorized as *content-dependent solutions* that make predictions based on the video content data, and *content-independent solutions* that require only the historical positions to anticipate the future viewing positions. Several existing prediction approaches predict future viewing positions using average [30], linear regression (LR) [25], [27], [30], motion-based [41]–[43], user-clustering [44]–[47], or straightforward machine learning (ML) [48]–[50] methods.

A short-term (0.5s-3s) viewpoint generator model has been proposed by Qian et al. [30] based on average, linear regression, and weighted linear regression methods. The authors entirely streamed the viewport tiles in higher resolution based on the derived future coordinates. Bao et al. [48] employed an LR-based neural network model to best fit the variations in the head tracking dataset. Azuma et al. [43] proposed a frequency-driven prediction model based on the viewing position, velocity, and acceleration. Likewise, Mavlankar et al. [41] characterized the user's viewing movements as motion vectors, i.e., speed and acceleration, for a zoomable panoramic framework. La Fuente et al. [42] tracked the future head position based on the angular velocity and angular acceleration of the user head movements. Petrangeli et al. [51] extrapolated the 100ms orientation data of the user to drive the viewing behaviors for the upcoming segment.

Linear regression and motion-based prediction approaches result in relatively lower prediction accuracy, especially for outdoor fast motion content [52]. Jiang et al. [53] established a Long-Short Term Memory (LSTM) model to analyze the user viewing behavior using an open-source dataset recorded with five videos watched by 59 users [54]. The authors found that most users have swift yaw movements than the movements in the pitch direction. They compared the proposed model with the LR and average approaches and showed that the LSTM-based viewport predictor outperforms the others for both yaw and pitch angle predictions. Qian et al. [25] proposed a practical view-based streaming system for commodity devices named Flare. They compared the performance of naive, LR, ridge regression (RR), and support vector regression (SVR) methods on 1300 head motion datasets collected from 130 users. They suggested using the LR (for <1s prediction window) and RR (≥1s prediction window) methods for the Flare to make it more robust and lightweight.

The cross-user learning-based systems can reduce the mismatch between predicted and ground truth data. Liu et al. [45] employed a data fusion approach that considers several exciting features, such as the behavior of the current and previous users, their engagement levels for a single or multiple videos, streaming device, and mobility-level among others, to predict the future viewing coordinates. Xie et al. [47] proposed a cooperative client-server view prediction model that can improve the prediction precision by up to 15% compared with LR. At the server-side, the users are grouped based on their watching interest for each video using the DBSCAN [55] clustering. On the client-side, the viewport prediction module decides the viewing group of the current user. Ban et al. [56] took advantage of users' attention distribution in 360° video to improve the view prediction performance. They analyzed the current user's watching behavior using the LR method and then combined it with other users' similar ROI using the K-Nearest-Neighbors (KNN) algorithm to fetch the viewport tiles for the next segment. Experimental evaluation on real datasets reveals that 20% improved prediction accuracy can result in up to 30% more quality gain than the LR-based streaming approach.

B. Tile-based Adaptive Streaming

Recently, tile-based adaptive streaming is a hot research direction that enables the client to optimize the spatial random bitrate allocation based on the user's interest and network constraints. Rossi et al. [22] undeviatingly designed a tile-based streaming algorithm to maximize user expectations for the known set of tiles. Similarly, based on the given viewport data, Ghosh et al. [23] encoded the visible tiles in higher resolution while the rest tiles in lower resolution according to the time-varying network constraints. The authors showed that streaming variable quality levels for the visible and non-visible regions can boost the performance in terms of formulated Quality of Experience (QoE) metric by up to 20%. Graf et al. [24] analyzed the performance of five tiling patterns, such as 1x1, 3x2, 5x3, 6x4, and 8x5, in comparison to the straight-forward monolithic streaming. The authors showed that a 6x4 tiling pattern could provide a worthwhile trade-off between coding performance and bandwidth consumption for different content types. Besides, they showed that a bandwidth saving of more than 60% could be achieved by employing a full delivery basic streaming strategy for a given viewport data. Chao et al. [57] proposed a clustering-based tiles selection mechanism, named ClusTile, to lower the bandwidth and computation overheads. ClusTile dynamically performs the tiles selection and bitrate adjustments for each segment. It could achieve a bandwidth saving of around 52% in comparison to the best-performing tiling method, as demonstrated by the experiments.

To minimize the impact of spatial quality variance and viewport quality distortion, Xie et al. [58] proposed a tile-based streaming framework that employs a QoE-driven target-buffer based rate optimization. Trace-driven experiments reveal that the proposed probability-based tiles-selection mechanism can enhance the visible quality levels by up to 39% and alleviate the spatial quality variance by 45% in comparison to other approaches. Hosseini et al. [29] proposed a priority-based adaptation algorithm for the central, surrounding, and outside tiles. The proposed algorithm firstly assigns the lowest quality to all the tiles. Then, it adjusts the central tile's quality to the maximum level and repeats the same procedure for the surrounding and outer tiles while respecting the available bandwidth budget. Hooft et al. [39] proposed two variants of bitrate adaptation for 360° videos, named as uniform viewport quality (UVP) and center tile first (CTF). As the name suggests, UVP allocates a uniform quality to all the viewport tiles, while the CTF mainly focuses on the viewport center similar to the [29]. The tiles are selected based on a *spherical walk* approach that extrapolates the 3D trajectory of the user movements on the spherical surface to predict the next viewing points. Petrangeli et al. [51] proposed a priority-aware HTTP/2 based segment transmission scheme to facilitate 360° video streaming. The urgent transmission of high-priority tiles based on the user's interest can improve the throughput performance compared to HTTP/1.1 under variable network delay conditions. Instead of solely relying on viewport and bandwidth data for quality adjustments, Nguyen et al. [59] proposed to select the viewport bitrate by also taking into account the viewport prediction errors during each segment duration. He et al. [60] performed a network delay-based joint selection of the viewport coverage and bitrate. The simulation outcomes confirm that adaptable viewport coverage offers improved quality streaming under different delay settings.

The algorithms discussed have set a stable background by considering user-specific viewing preferences for 360° videos. However, the space and time separation of such videos makes it challenging to develop a successful VR streaming framework. Most proposed schemes including [20], [23], [61], [62] use different quality levels for the viewport and background tiles. This approach can assist in bandwidth-efficient streaming. However, following inaccurate FoV predictions, the different quality tiles can dramatically lower user-perceived video quality. Moreover, all existing schemes transmit the tiles based on a single prediction mechanism and then expand the viewport either in all directions [24], [29], [51] or towards some specific sides [59]. Compared to previous works, this paper's novelty lies in dynamically deciding the coverage of different viewing regions based on the combined output of two FoV prediction mechanisms to achieve higher viewport matching performance. In contrast to [29], [39], [51], the proposed CFOV streams either the viewport only tiles or all the tiles at certain quality levels by learning tile distribution and real-time network transmission capacity.

III. Proposed Solution

A. System Architecture

The proposed end-to-end 360° video streaming solution aims to improve the viewport overlap by requesting extra tiles in higher resolution while reducing the bandwidth utilization for background tiles. Fig. 2 illustrates the end-to-end 360° video streaming framework. The server is responsible for storing and pre-processing video content. The 360° sphere representation is transformed into an equirectangular projection format [63] following capturing and stitching steps. The equirectangular projected video is temporally split into S equal duration segments, and each segment is prepared in M spatial tiles, and each tile encoded into N bitrate levels. Let $\mathcal{L}_j^k(i)$ represents the quality level $j \in [1, N]$ of tile $k \in [1, M]$ in segment $i \in [1, S]$. Let $x^k(i)$ be a decision variable representing that the kth tile for (i)th segment is selected for streaming (i.e., $x^k(i) = 1$) or not (i.e., $x^k(i) = 0$).

At the client-side, the *FoVs Prediction* module predicts the future FoVs coordinates based on the user's watching history. Accordingly, the *Tiles Selection* module selects the viewport, external, and background tiles sets for the (i)th segment, i.e., $\mathcal{T}^v(i)$, $\mathcal{T}^e(i)$, $\mathcal{T}^b(i)$, from the tiles set, $\mathcal{T}(i)$. Based on the output of the *Tiles Selection* module, the *Bitrate Adaptation* unit selects suitable bitrates for each tile according to the associated region and the estimated network throughput. Once the segments are received, the client performs decoding and stitching of the requested tiles to reconstruct the 360° video. It then performs the rendering and starts playing the requested content. Table I includes the mathematical symbols and their meanings used in the following discussion.

TABLE I: Notations used in the paper

Symbol	**Meaning**
S, N, M	Number of segments, bitrates, and tiles
i, j, k	Index of segment, bitrate, and tiles
τ	Segment duration
$\mathcal{T}^{\hat{v}}(i)$	Set containing the actual viewport tiles
$\mathcal{T}^{\hat{b}}(i)$	Set containing the actual background tiles
$\mathcal{T}(i)$	Set containing all the tiles in a streaming session
$\mathcal{T}^v(i)$	Set containing the predicted viewport tiles
$\mathcal{T}^b(i)$	Set containing the predicted external tiles
$\mathcal{T}^b(i)$	Set containing the predicted background tiles
$\mathcal{L}_j^k(i)$	Video bitrate level j selected for kth tile of (i)th segment
$\widehat{Th}(i)$	Estimated throughput for the (i)th segment
$Th^v(i), Th^e(i)$	Estimated throughput for the viewport and external tiles
$\alpha, \beta, \gamma, \delta$	Weight Coefficients
$TO(i)$	Tiles overlap for (i)th segment

B. Problem Definition

The high-quality expectations of the user mainly depend on the quality of the visible area. The lower rate of visible tiles may not satisfy the user even if the background tiles are played in good quality. Some key challenges to help support the high QoE levels include real-time scene update, accurate FoV prediction, tiles selection, adaptive quality adjustments, and employing efficient delivery protocols, among others [52]. The client seeks optimal bitrates for each segment, intending to optimize the user's long-term QoE reward, subject to the constraints (2-7). Mathematically, the optimization problem can be formulated as the following problem:

Problem:

$$arg \max_{i \in [1,S]} QoE(i) \tag{1}$$

Constraints:

$$\sum_{k \in \mathcal{T}(i)} \mathcal{L}_j^k(i) * x^k(i) \le \widehat{Th}(i), \forall j \in [1, N] \tag{2}$$

$$\mathcal{L}_j^k(i) * x^k(i) = \mathcal{L}_j^{k'}(i) * x^{k'}(i), \forall k, k' \in \mathcal{T}^v(i), \forall j \in [1, N] \tag{3}$$

$$\mathcal{L}_j^k(i) = \mathcal{L}_1^k(i) * x^k(i), \forall k \in \mathcal{T}^b(i), \forall j \in [1, N] \tag{4}$$

$$\sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) * x^k(i) \le Th^v(i), \forall j \in [1, N] \tag{5}$$

$$\sum_{k \in \mathcal{T}^e(i)} \mathcal{L}_j^k(i) * x^k(i) \le Th^e(i), \forall j \in [1, N] \tag{6}$$

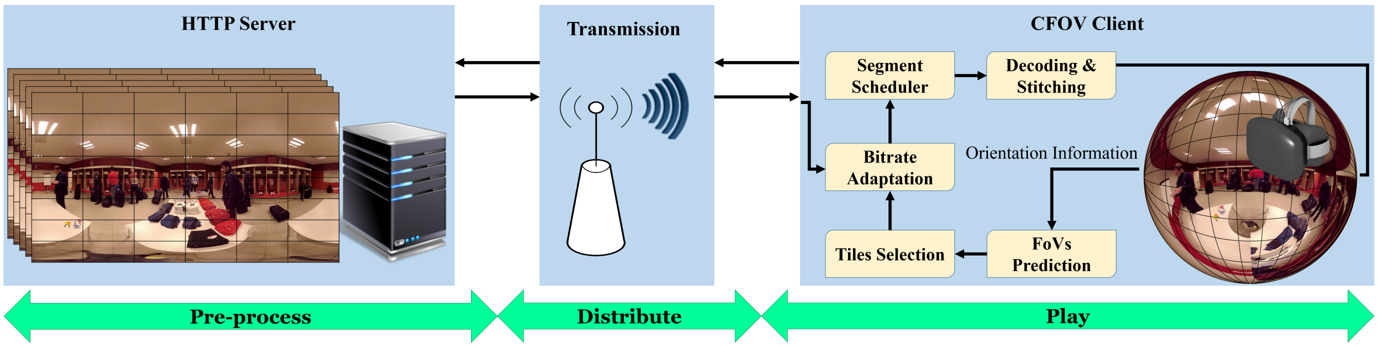


Fig. 2: The proposed end-to-end 360° video streaming framework.

$$\sum_{k\in\mathcal{T}^{v}(i)} \mathcal{L}_j^k(i)*x^k(i) \geq \sum_{k\in\mathcal{T}^{e}(i)} \mathcal{L}_j^k(i)*x^k(i), \forall j \in [1,N] \quad (7)$$

The constraints in Eq. (2) limit the selected bitrate of all the tiles in *(i)th* segment. Constraints in Eq. (3) state that all the viewport tiles should have the same selected video bitrate. Eq. (4) restricts the bitrate of all the background tiles to the lowest quality level (i.e., $\mathcal{L}_1^k(i)$). The constraints in Eq. (5) and Eq. (6) make sure that the bitrate of the viewport and external tiles is not higher than the throughput of the viewport and external tiles, respectively. Particularly, the viewport tiles are downloaded in higher bitrates compared to the external tiles. The throughput calculation based on the importance of each region is described in section IV.D. Finally, the constraints in Eq. (7) ensure that the bitrates of external tiles would not be higher than the viewport tiles. The following three steps solve the above problem:

1) defines a user QoE metric that assesses the perceived quality not solely based on the visual quality.
2) employs a content-agnostic FoVs prediction-based tiles selection approach that dynamically performs the viewing area selection to improve the overlap between real and predicted viewport tiles.
3) selects optimal quality levels by assigning priority-related weights to each tile of different regions.

The following subsection elaborates on these aspects.

IV. Proposed Architecture and Algorithms

A. CFOV QoE

With the adaptive transmission of omnidirectional video, it is imperative to recognize the unique quality aspects of the consumer due to its highly prevalent nature. How long a user feels immersion in a VR video dictates the level of experience perceived by users. Accurate QoE assessment is a key factor in optimizing the adaptive video streaming [64]. However, calculating visual quality alone is not adequate for a complete VR QoE framework. In evaluating the user's QoE, it is also essential to define the effect of other parameters, e.g., bandwidth savings, spatial fluctuations, and temporal quality variations, etc.

- **Viewport Quality:** By averaging the quality of the viewport tiles based on the real viewport traces, we get the viewport quality in segment (i) as follow [65], [66]:

$$f_1(i) = \frac{\sum_{k\in\mathcal{T}^{\hat{v}}(i)} \sum_{j\in[1,N]} Q(\mathcal{L}_j^k(i))}{|\mathcal{T}^{\hat{v}}(i)|} \quad (8)$$

where $\mathcal{T}^{\hat{v}}(i)$ represents the actual viewport tiles set and $|\mathcal{T}^{\hat{v}}(i)|$ indicates the number of tiles in viewport tiles set. $Q(\mathcal{L}_j^k(i))$ maps the video bitrate to the relevant quality level for the (i)th segment.

- **Background Quality:** Ideally, the 360° client should only stream the viewport tiles at best possible quality with no background tiles. But several solutions stream the background tiles to lower the impact of viewport anomalies due to the limited precision of prediction mechanisms. This metric explicitly indicates the average quality of the background tile in (i)th segment and is given as follow:

$$f_2(i) = \frac{\sum_{k\in\mathcal{T}^{\hat{b}}(i)} \sum_{j\in[1,N]} Q(\mathcal{L}_j^k(i))}{|\mathcal{T}^{\hat{b}}(i)|} \quad (9)$$

where $\mathcal{T}^{\hat{b}}(i)$ represents the background tiles set which contains tiles not visible to the user based on the ground truth viewport traces during the (i)th segment. The term in the denominator $|\mathcal{T}^{\hat{b}}(i)|$ represents the number of tiles in background tiles set.

- **Temporal Quality Oscillations:** The efficiency of tile-based streaming schemes can be impaired by the disparity in quality levels between two viewports of consecutive segments. Therefore, the temporal quality fluctuations need not be drastic and can be calculated by [65]:

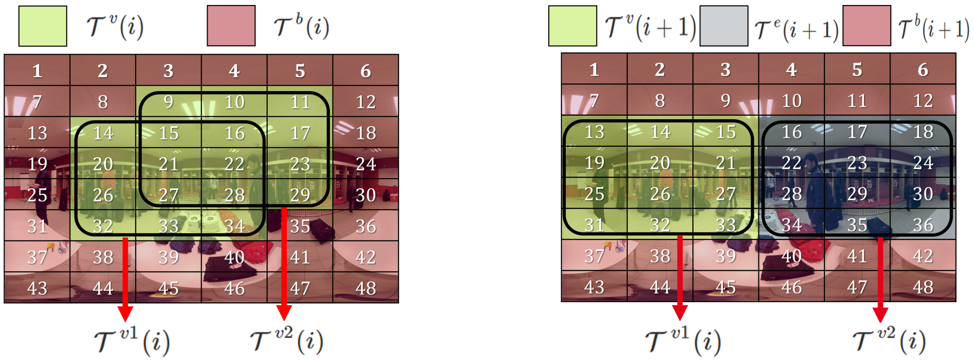
$$f_3(i) = |\, f_1(i) - f_1(i-1)| \quad (10)$$

- **Spatial Quality Oscillations:** Cybersickness, viewing irritation, and other physiological effects, such as nausea, fatigue, and aversion [67], can be driven by variable quality levels within the viewport. That leads, therefore, to lower QoE levels. Following [53], we measure this value according to the coefficient of variation (CV) of the viewport quality levels.

$$f_4(i) = \frac{\sigma(Q(\mathcal{L}_j^k(i)))}{\mu(Q(\mathcal{L}_j^k(i)))}, \quad \forall k \in \mathcal{T}^{\hat{v}}(i), \forall j \in [1,N] \quad (11)$$

The term in the numerator represents the standard deviation of the viewport quality samples, while the denominator represents the mean of the samples.

Following the principle behind the QoE metric for traditional video [35], we define a QoE metric for 360° video.



(a) Extended FoV for the (i)th segment

(b) Fixed FoV for the $(i+1)$th segment

Fig. 3: Tiles selection cases for two consecutive segments in CFOV framework.

The proposed metric considers the weighted summation of the above mentioned components and is given as follows:

$$QoE(i) = \alpha * f_1(i) - \beta * f_2(i) - \gamma * f_3(i) - \delta * f_4(i) \qquad (12)$$

where α, β, γ, and δ are the non-negative weight coefficients corresponding to the background quality and temporal and spatial quality oscillations, respectively. We want to minimize the $f_2(i)$, $f_3(i)$, and $f_4(i)$, therefore, these functions are negative.

B. CFOV Tiles Selection

360° video has become an integral part of popular multimedia applications, as the consumer is interested in an increasingly interactive and immersive streaming experience. One of the key features of VR devices is to update the scene according to the viewer's head movement. When a user changes his viewpoint, the end terminal processes the feedback signals and can render the relevant FoV so that a view is accessible from a regular visual angle. Typically, a user will access only a small portion of the stream being transmitted. The high-quality transmission of the entire frames results in the waste of a large amount of bandwidth used for the unseen portion of the content. In addition, the viewing experience of a user depends on how efficiently the client can select the visible tiles for the next segment. For instance, if video tiles are requested based on an incorrect prediction, the user's actual viewport may be covered by black tiles for which no content was requested.

Viewport prediction is analogous to a sequence prediction problem, which focuses on forecasting future viewing positions based on past head movement trajectories. It has become an essential part of 360° video streaming. However, the latest FoV prediction models result in a low long-term prediction accuracy [68]. Rondon et al. [69] reported that existing neural network models used for both content-based and content-independent viewport prediction perform worse than a basic (last known) approach that simply uses the last viewing position for the next segment. Due to the extremely unpredictable viewing nature of the user, the basic idea is to stream more tiles than necessary to cover the actual viewing area. This work considers two viewpoint/viewport prediction mechanisms to perform the interactive tiles selection during each adaptation interval. The current viewing point is used as the first predicted viewpoint for the next segment. A *spherical walk* approach proposed in [39] is adopted for the second viewpoint prediction that considers the user's motion as a walk on a sphere and predicts the future position based on the spherical movement from one point to another point. Based on the two predicted viewpoints and the FoV of the headset (usually in the range of 90°-110°), the tiles for both viewports are selected by calculating the spherical distance between the predicted viewpoints and the center of each of the tiles. The tiles whose centers are less than half of the FoV size apart from the viewpoint will belong to the viewport region. In this way, both the first (*last known*) and second (*spherical walk*) viewport sets represented by et $\mathcal{T}^{v1}(i)$, $\mathcal{T}^{v2}(i)$, respectively, are computed for (i)th segment.

Algorithm 1: CFOV Tiles Selection Algorithm

```
Input :
    $\mathcal{T}(i) \leftarrow$ Tiles set in the streaming session
    $\mathcal{T}^{v1}(i) \leftarrow$ Tiles set for the first predicted FoV
    $\mathcal{T}^{v2}(i) \leftarrow$ Tiles set for the second predicted FoV
Result :
    $\mathcal{T}^v(i), \mathcal{T}^e(i), \mathcal{T}^b(i) \leftarrow$ Estimated viewport, external,
    and background tiles sets for the $(i)$th segment
if $(\mathcal{T}^{v1}(i) \cap \mathcal{T}^{v2}(i) \neq \emptyset)$ then
    $\mathcal{T}^v(i) = \mathcal{T}^{v1}(i) \cup (\mathcal{T}^{v2}(i) - \mathcal{T}^{v1}(i))$
    $\mathcal{T}^e(i) = \emptyset$
    $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$
else
    $\mathcal{T}^v(i) = \mathcal{T}^{v1}(i)$
    $\mathcal{T}^e(i) = \mathcal{T}^{v2}(i)$
    $\mathcal{T}^b(i) = \mathcal{T}(i) - (\mathcal{T}^v(i) \cup \mathcal{T}^e(i))$
```

For each video segment, the client classifies a 360° video frame into the viewport, external, and background regions. We consider *Extended FoV* and *Fixed FoV* cases for the innovative tiles selections in CFOV. Fig. 3a illustrates the *Extended FoV* case in an equirectangular space where both the predicted FoVs partially overlap during the (i)th segment. The goal here is to extend the FoV coverage by adding non-overlapping tiles of the second FoV tiles set to the first FoV tiles set to deal with possible head movement prediction errors. With

Algorithm 2: CFOV Bitrate Allocation Algorithm

```
Input :
  ${\mathcal{L}_1(i), ..., \mathcal{L}_j(i), ..., \mathcal{L}_N(i)}$← Video bitrate-levels set for each tile in $(i)th$ segment
  $\mathcal{T}(i)$← Tiles set containing M tiles for the $(i)$th segment
  $\mathcal{T}^v(i), \mathcal{T}^e(i)$← Tiles sets for the viewport and external regions computed from Algorithm 1
  $|\mathcal{T}^v(i)|, |\mathcal{T}^e(i)|$← Number of tiles in viewport and external regions
  $\widehat{Th}(i)$← Estimated throughput for $(i)$th segment
Result :
  $w^{\mathcal{T}^v}(i), w^{\mathcal{T}^e}(i)$← Priority related weights for viewport and external tiles
  $Th^v(i), Th^e(i)$← Estimated throughput for the viewport and external tiles
  $\mathcal{L}^{\mathcal{T}}(i), \mathcal{L}^{\mathcal{T}^v}(i), \mathcal{L}^{\mathcal{T}^e}(i)$← Video bitrates selected for the tiles of $(i)$th segment
1  if $(1 + \Delta * \widehat{Th}(i) \leq \sum_{k \in \mathcal{T}(i)} \mathcal{L}_1^k)$ then
2      $\mathcal{L}^{\mathcal{T}^v}(i) = \max_{j \in [1,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq \widehat{Th}(i)\}$
3  else
4      $\mathcal{L}^{\mathcal{T}}(i) = \mathcal{L}_1^k(i), \forall k \in \mathcal{T}(i)$
5      $Th(i) = \widehat{Th}(i) - \sum_{k \in \mathcal{T}(i)} \mathcal{L}_1^k(i)$
6      if $(\mathcal{T}^e(i) = \emptyset)$ then
7          $Th^v(i) = Th(i)$
8          $\mathcal{L}^{\mathcal{T}^v}(i) = \max_{j \in [2,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq Th^v(i)\}$
9      else
10         $w^{\mathcal{T}^e}(i) = (|\mathcal{T}^e(i)| / (2 * |\mathcal{T}^v(i)| + |\mathcal{T}^e(i)|))$
11         $w^{\mathcal{T}^v}(i) = 1 - w^{\mathcal{T}^e}(i)$
12         $Th^v(i) = Th(i) * w^{\mathcal{T}^v}(i)$
13         $Th^e(i) = Th(i) * w^{\mathcal{T}^e}(i)$
14         $\mathcal{L}^{\mathcal{T}^v}(i) = \max_{j \in [2,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq Th^v(i)\}$
15         $\mathcal{L}^{\mathcal{T}^e}(i) = \max_{j \in [2,N]} \{\mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^e(i)} \mathcal{L}_j^k(i) \leq Th^e(i)\}$
```

no external tiles in *Extended FoV* case, the rest of the tiles belong to the background region. Due to the abrupt user movements, different mechanism's predicted viewpoints can be far from each other. In this case, we can stream both FoVs by executing priority-based bitrate budget distribution to facilitate differentiated quality streaming. Fig. 3b represents the *Fixed FoV* case, where the two FoVs do not have common tiles for the $(i+1)$th segment. The first FoV tiles are classified as viewport tiles, while the second FoV tiles set includes external tiles for the $(i+1)$ segment. The external tiles can be streamed in higher resolution than the background tiles, which are always streamed with the lowest resolution.

Algorithm 1 describes the *Tiles Selection* mechanism in CFOV. For each segment, the tiles belonging to the different regions are chosen dynamically based on the performance of the prediction mechanisms. Algorithm 1 begins by finding the intersection of two predicted FoVs and then selects the tiles for each of the three regions. The viewport tiles set ($\mathcal{T}^v(i)$) is determined by adding all the unique tiles of two predicted tiles sets if the intersection of two predicted sets is not empty for the (i)th segment (lines 1-2). In *Extended FoV* case, the set of external tiles ($\mathcal{T}^e(i)$) does not contain any tiles (line 3). All the remaining tiles are classified as background tiles (line 4). If the tiles in both FoVs are identical, then similar tiles of both FoVs, referred to as viewport tiles, along with the background tiles, are inputted to the bitrate allocation unit. For the *Fixed FoV* case, where the two predicted FoVs do not overlap, the tiles belonging to the first and second FoVs are labeled as viewport and external tiles, respectively (lines 6-7). The range of background tiles set is then computed for the (i)th segment (line 8).

C. CFOV Bitrate Allocation

In adaptive streaming, a key challenging aspect is to predict the network throughput correctly [70]. An under-estimation of the actual throughput may lead to requests for lower quality segments, while an over-estimation may result in significant rebuffering events. The HTTP clients infer the network throughput from prior measurements [71]. The calculation of the throughput for the (i)th segment is defined in Eq. (13).

$$\widehat{Th}(i) = \frac{\sum_{k \in \mathcal{T}(i)} \mathcal{L}_j^k(i-1) * \tau}{\mathcal{F}(i-1)} \tag{13}$$

where $\mathcal{L}_j^k(i-1)$ represents the bitrate of all the tiles, τ is the playback duration of the segment, and $\mathcal{F}(i-1)$ represents the total fetching time of the $(i-1)$th segment.

Algorithm 2 allocates the bitrate to the outputted tiles of the *Tiles Selection* module to achieve the optimization aims described in Eq. (1) and Eq. (12). The adaptation for the (i)th segment is performed after completely fetching the $(i-1)$th

TABLE II: Average video bitrates for the *Boxing*, *Conan*, *Football*, and *Spotlight* videos [Mbps]

		1s			**2s**			**3s**		
Video	**QP**	**4x3**	**6x4**	**8x6**	**4x3**	**6x4**	**8x6**	**4x3**	**6x4**	**8x6**
Boxing	22.00	24.81	25.39	26.13	49.63	50.80	52.29	74.44	76.18	78.41
	27.00	13.38	13.75	14.28	26.78	27.53	28.60	40.16	41.27	42.87
	32.00	7.21	7.45	7.83	14.44	14.92	15.69	21.65	22.37	23.50
	37.00	3.95	4.12	4.39	7.92	8.26	8.81	11.88	12.37	13.19
	42.00	2.19	2.30	2.52	4.39	4.63	5.07	6.58	6.93	7.57
Conan	22.00	10.60	10.68	10.88	21.22	21.37	21.77	31.63	31.85	32.44
	27.00	5.05	5.12	5.30	10.10	10.25	10.63	15.05	15.26	15.83
	32.00	2.43	2.51	2.68	4.87	5.02	5.39	7.26	7.48	8.01
	37.00	1.24	1.32	1.50	2.49	2.66	3.03	3.71	3.95	4.49
	42.00	0.72	0.80	0.98	1.44	1.62	1.99	2.14	2.40	2.94
Football	22.00	6.90	7.02	7.19	13.88	14.12	14.48	20.69	21.04	21.56
	27.00	3.55	3.64	3.81	7.14	7.34	7.68	10.64	10.92	11.41
	32.00	1.97	2.07	2.23	3.98	4.16	4.50	5.92	6.19	6.68
	37.00	1.15	1.24	1.40	2.31	2.49	2.82	3.44	3.70	4.18
	42.00	0.69	0.77	0.93	1.39	1.56	1.88	2.06	2.31	2.78
Spotlight	22.00	13.63	13.93	14.31	27.18	27.77	28.55	40.76	41.64	42.80
	27.00	7.21	7.44	7.77	14.37	14.84	15.50	21.54	22.25	23.23
	32.00	4.06	4.24	4.50	8.11	8.47	9.01	12.15	12.69	13.48
	37.00	2.36	2.49	2.72	4.70	4.98	5.44	7.04	7.46	8.13
	42.00	1.35	1.46	1.66	2.69	2.93	3.35	4.03	4.38	4.99

segment. Algorithm 2 ensures that the segment size does not exceed the available bandwidth budget in fulfilling the constraints from Eq. (2). Suppose the lowest available video bitrate for the entire 360° segment is $(1 + \mathbf{\Delta})$ times greater than the estimated network throughput. In that case, only the viewport tiles with the highest permitted video bitrate are streamed to ensure a seamless video playback corresponding to Eq. (3) for the actual spatial smoothness defined in Eq. (11) (lines 1-2). Otherwise, the bitrate allocation is carried out for the entire frame by firstly assigning the lowest bitrate to all the tiles (Eq. (4)) to achieve a lower background penalty for the actual background tiles, defined in Eq. (9) (line 4). The bandwidth budget is revised then (line 5). Next, the viewport throughput is determined to select the best possible video bitrate for the viewport tiles if there are no external tiles (lines 6-7). All the viewport tiles are streamed with the same selected rate to improve the perceived visual quality levels mentioned in Eq. (8) (line 8). Next, if the external tiles set is non-empty, the proposed algorithm ensures that similar to the constraints in Eq. (5) and Eq. (6), the bitrate of the viewport and external tiles is not higher than the throughput of the viewport and external tiles, respectively. The bitrate allocation is performed for viewport and external tiles after calculating their priority-related weights. The weights are determined depending on the number of tiles in the viewport and external regions (lines 10-11). As the viewer is more interested in watching the viewport content at higher quality levels; therefore, viewport tiles are assigned with double weights compared to the external tiles to fulfill the constraints in Eq. (7). After that, the throughput for the viewport and external tiles are computed based on computed weights (lines 12-13). Finally, the video bitrate levels for the viewport and the external tiles are calculated. The maximum available video bitrates not exceeding each region's corresponding throughput budget are allocated to each viewport and external tiles (lines 14-15). Noteworthy is to state that Algorithm 2 provides a solution to achieve an important balance between different quality objectives defined in Eq. (8)-Eq. (11) under constraints in Eq. (2)-Eq. (7) and maximize the optimization goal defined in Eq. (12).

TABLE III: Experimental Settings

Parameter	**Configuration**
Video, Duration, & Category	Boxing -2′52″ (Sport)
	Conan-2′44″ (Performance)
	Football- 2′44″ (Sport)
	Spotlight- 4′53″ (Film)
Resolution & FPS	Boxing (3840x1920)-29FPS
	Conan (3840x2160)-29FPS
	Football (3840x2160)-25FPS
	Spotlight (3840x2160)-30FPS
Tiling patterns	4x3, 6x4, 8x6
Encoding Versions	QPs (22, 27, 32, 37, 42)
Segment Duration (τ)	1s, 2s, 3s
Viewport Size	110°
Simulation Length	Video duration
QoE Coefficients	$(\alpha = 1, \beta = 0.3, \gamma = 0.1, \delta = 0.1)$
	$(\alpha = 1, \beta = 0.4, \gamma = 0.2, \delta = 0.2)$
	$(\alpha = 1, \beta = 0.5, \gamma = 0.3, \delta = 0.3)$

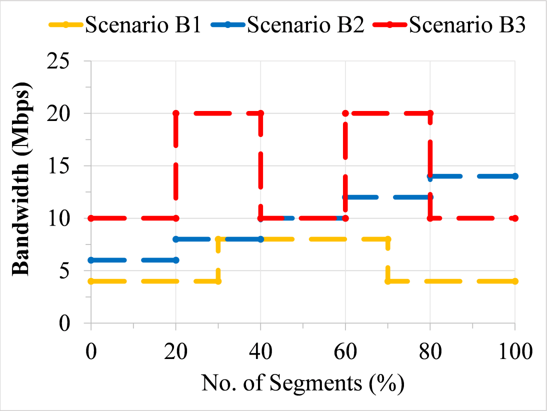


Fig. 4: Bandwidth scenarios used in experiments.

V. Experimental Evaluation

This section introduces the trace-driven evaluation of the proposed solution compared to the existing tile-based solutions under a wide variety of content and network characteristics. Next, we present the experimental setup and the comparison schemes. Then, for each of the streaming solutions, we show the experimental results and their analysis.

TABLE IV: Viewport Streaming Approaches for Tile-based Adaptive 360° Video

Works	Prediction Mechanism		FoV Selection		Streaming Strategy		
	Single	Combined	Fixed FoV	Extended FoV	Viewport Only	Viewport and Background	Viewport, External, and Background
UVP [39]	✓	✗	✓	✗	✗	✓	✗
CTF [39]	✓	✗	✗	✓	✓	✗	✗
Hos [29]	✓	✗	✓	✗	✗	✓	✗
Pet [51]	✓	✗	✓	✗	✗	✗	✓
CFOV	✗	✓	✓	✓	✓	✓	✓

A. Experimental Setup

1) 360° Video Player: The modeling and evaluation of the proposed system were conducted by employing an enhanced version of a VR player[2] running on an Ubuntu 16.04 machine with a 64-bit Intel Core i7-7500U CPU 2.7 GHz quad-core and 16 GB memory. The 360° player requested video segments from the HTTP server based on the available bandwidth and estimated viewpoint coordinates.

2) 360° Videos and Head Movement Traces: The experimental evaluation was performed using a trace-driven simulation involving real viewport traces of 48 VR users from an open-source dataset [72]. This dataset is widely used, including in [44], [47], [66], [73]. From this dataset, we chose four videos which include a wide range of motion content: *SHOWTIME Boxing*[3], *Conan360-Sandwich*[4], *LOSC Football*[5], and *Google Spotlight-HELP*[6]. The content category also differed across the selected clips; the first and third streams belong to the sports category, while the second and fourth clips belong to artistic performance and action film, respectively. These videos are referred to as *Boxing*, *Conan*, *Football*, and *Spotlight* throughout this paper. All the videos were rescaled to the 4K resolution using FFmpeg[7] software. The videos were spatially split into 4x3, 6x4, and 8x6 tiling patterns. The tiles were encoded using an open-source Kvazaar encoder [74] considering five different quantization parameters (QPs) (i.e., 22, 27, 32, 37, and 42). Subsequently, DASH video segments were generated using GPAC MP4Box[8] with a duration of 1s, 2s, and 3s, respectively. The average segment sizes for each video and encoding rate are illustrated in Table II. The value of Δ was set to 0.5, and the viewport coverage was set to 110°, as used in the head movement collection by Wu et al. [72]. The length of each simulation was equal to the duration of the video employed. Table III presents the content characteristics and experimental settings.

3) Bandwidth Scenarios: The experiments were performed using the following dynamic bandwidth scenarios, also shown in Fig. 4:

1) Scenario B1: The bandwidth of the link between the HTTP client and server was varied for each video as follows: 4 Mbps for the first 30% of the segments, 8 Mbps for the following 40% segments, and then back to 4 Mbps until the end of the video playback. Scenario B1 was used for the experiments performed for 1s segment duration.
2) Scenario B2: A stepwise switch-up connection was considered for the 2s video segment durations where the bandwidth was set to 6 Mbps for the first 20% of the segments, and then increased with 2 Mbps after every 20% of the playback.
3) Scenario B3: The bandwidth of the link between the client and HTTP server varied in an on-off between 10 Mbps and 20 Mbps after each 20% of the playback for all the videos. Scenario B3 was employed for the 3s video segment duration.

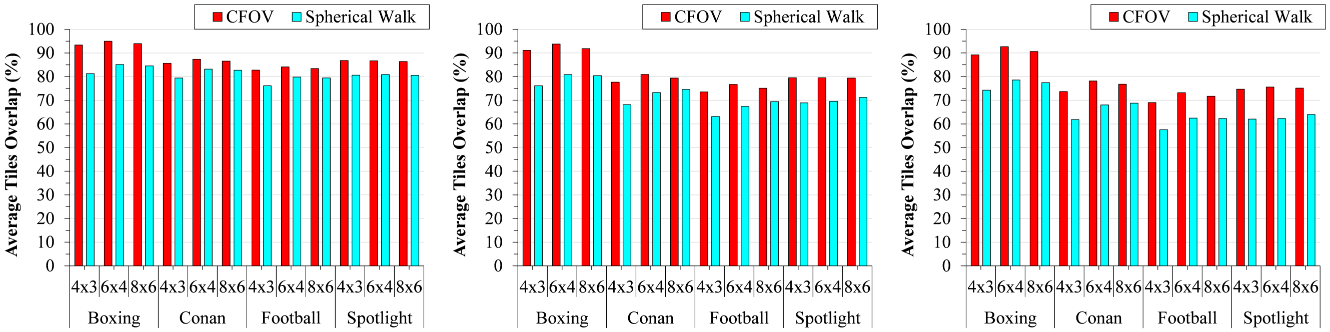
4) QoE Weight Coefficients: To verify the effectiveness of the proposed solution, the following sets of QoE coefficients are chosen:

1) Coefficients C1: ($\alpha = 1, \beta = 0.3, \gamma = 0.1, \delta = 0.1$)
2) Coefficients C2: ($\alpha = 1, \beta = 0.4, \gamma = 0.2, \delta = 0.2$)
3) Coefficients C3: ($\alpha = 1, \beta = 0.5, \gamma = 0.3, \delta = 0.3$)

In practice, the QoE weight coefficients can be selected in order to emphasize different QoE objectives such as to maximize the viewport quality, minimize the background content quality, and reduce the spatial and temporal quality variations or their combination.

5) Baseline Algorithms: We compare the performance of CFOV with four tile-based streaming solutions. All of the reference tile-based delivery solutions incorporate viewer head movements for adaptive bitrate selection. The first approach, denoted as UVP [39], classifies the tiles into the viewport and non-viewport regions; no external region is considered here. It initially selects the lowest resolution for all the tiles; then, it uniformly increases the quality of viewport and outer tiles while respecting the available bandwidth budget. The second method, referred to as CTF [39], increases the quality, starting from the viewport center to the last tile. The third approach denoted as Hos [29], performs priority-based bitrate adaptation for tiles belonging to three zones, Z_1 (viewpoint tile), Z_2 (viewpoint surrounding tiles), and Z_3 (background tiles). The fourth approach, denoted as Pet [51], divides the 360° frames into the viewport, adjacent, and outside regions. Different from the previous works, the external area in our method is a special case. It could be adjacent to the viewport or can reside at a distance from the viewport depending on the difference of the prediction mechanisms. Table IV illustrates the significant differences between the proposed and the comparative schemes in terms of prediction mechanism, FoV selection, and streaming strategy.

[2] https://github.com/jvdrhoof/VRClient
[3] https://www.youtube.com/watch?v=raKh0OIERew
[4] https://www.youtube.com/watch?v=FiClYLgxJ5s
[5] https://www.youtube.com/watch?v=lvH89OkkKQ8
[6] https://www.youtube.com/watch?v=G-XZhKqQAHU
[7] https://ffmpeg.org/
[8] https://gpac.wp.imt.fr/mp4box/



(a) Segment duration: 1s

(b) Segment duration: 2s

(c) Segment duration: 3s

Fig. 5: Average tiles overlap achieved by CFOV and Spherical Walk methods for 1s, 2s, and 3s prediction horizons; results from four videos watched by 48 users.

B. Experimental Results

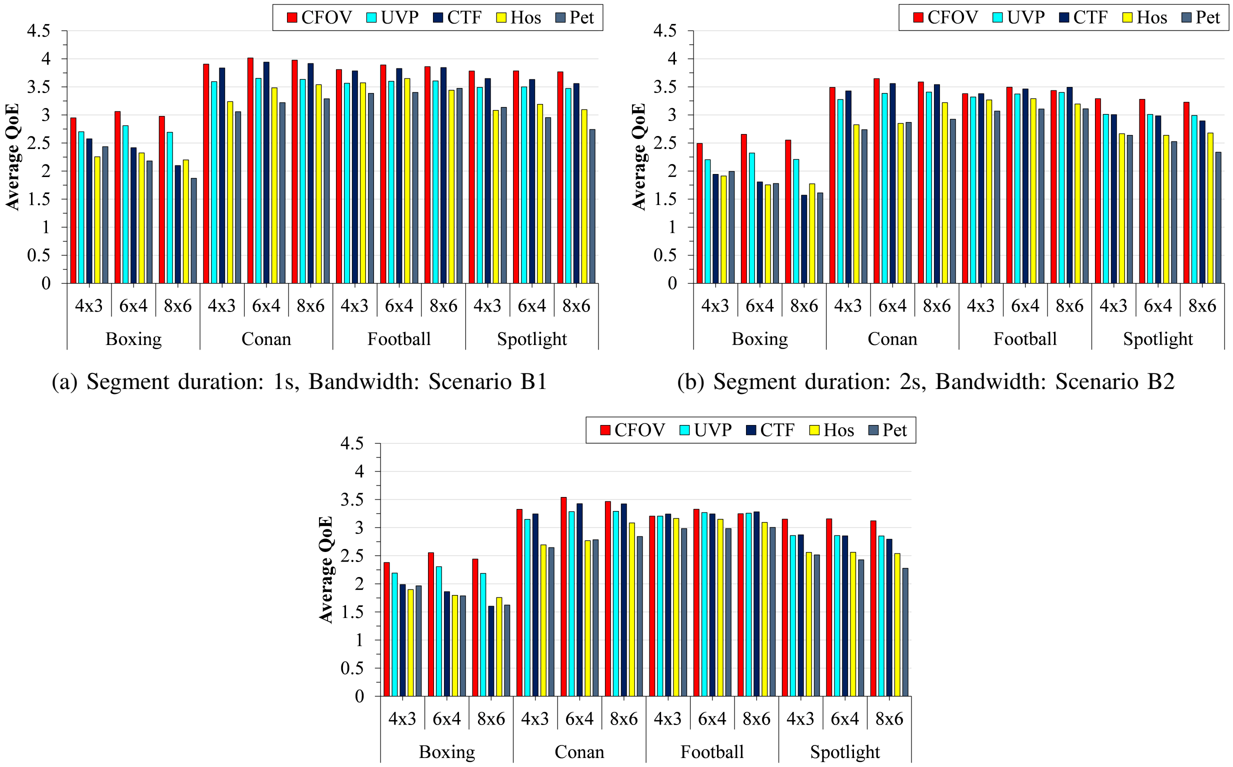
1) Tiles Overlap: This metric directly calculates the fraction of the actual viewport tiles $\mathcal{T}^{\hat{v}}(i)$ covered by the predicted viewport tiles $\mathcal{T}^{v}(i)$. For the (i)th segment, the tile overlap is given as follows [75]:

$$TO(i) = \frac{|\mathcal{T}^{\hat{v}}(i) \cap \mathcal{T}^{v}(i)|}{|\mathcal{T}^{\hat{v}}(i)|} \tag{14}$$

We compare CFOV with a *spherical walk* approach proposed in [39]. Fig. 5 illustrates the average tiles overlap for four videos prepared with three segment durations and three tiling patterns across the 48 head movement traces. The *spherical walk* method is adopted in UVP, CTF, Hos, and Pet streaming algorithms. A relatively low tile overlap is observed for the 4x3 tiling pattern since both methods arrange the tiles considering arc distance between the viewpoint and each tile's center. The viewers have relatively fast head movements when watching the *Football* video since it is an outdoor sports video and contains several fast-moving objects. A high average tile overlap is observed for the *Boxing* video since the users have more regular and stable head movements for this video, which are easy to forecast. It is notable that proactive tile selection in CFOV mainly yields high matching performance and outperforms the *spherical walk* approach for different user behaviors. In particular, for all four videos, CFOV experiences an average tile overlap of more than 80% (Fig. 5a). The *spherical walk* approach observes a low average tile overlap because the actual and predicted viewport positions are far from each other even when the head movements are stable. As seen, for the *Boxing* video, CFOV outperforms the *spherical walk* method by 10.46% for 1s (Fig. 5a), by 13.09% for 2s (Fig. 5b), and by 14.05% for the 3s prediction horizon (Fig. 5c). Similarly, CFOV shows its superior capability in increasing viewport match for the *Spotlight* video and outperforms the *spherical walk* method for the 4x3 tiling pattern by 6.22% for 1s (Fig. 5a), by 10.6% for 2s (Fig. 5b), and by 12.62% for the 3s prediction horizon (Fig. 5c). Fig. 5a shows that for the 1s prediction window, CFOV gets very close to a perfect viewport match (e.g., 94.99% for the *Boxing* video, 87.35% for the *Conan* video, 84% for the *Football* video, and 86.82% for the *Spotlight* video) across all tiling patterns. At the same time, a very small percentage of viewport mismatch is observed when the prediction horizon is set to 2s (e.g., 7.78% for the *Boxing* video, 20.69% for the *Conan* video, 24.9% for the *Football* video, and 20.52% for the *Spotlight* video) (Fig. 5b). The evaluation results show that our dynamic tiles selection method ensures stable visual angles to provide users with a favorable QoE. It is also notable from Fig. 5c that CFOV outperforms the *spherical walk* approach by up to 14.89% for the *Boxing* video, up to 11.84% for the *Conan* video, up to 11.46% for the *Football* video, and up to 13.31% for the *Spotlight* video. This is because the tiles selection cases in CFOV adapt better to the varying user behaviors for different video characteristics. As a result, it can be concluded that CFOV exploits user-watching information better than the *spherical walk* method and reduces the mismatch between the actual and predicted FoV tiles.

2) Average QoE with Coefficient Set C1: We computed the average quality score based on the QoE metric defined in Section IV-A. Fig. 6 presents the average QoE scores achieved by each streaming algorithm under the three dynamic bandwidth scenarios for the *Boxing*, *Conan*, *Football*, and *Spotlight* videos, which are spatially and temporally split into three tiling patterns and three segment durations, respectively. The performance results are depicted for the QoE weight coefficient set C1. The experimental findings reveal that CFOV attains an optimal trade-off when selecting the streaming tiles quality and the highest QoE among the approaches compared. The algorithms' performance decreases accordingly with the increase of segment length. The *Boxing* video requires higher bitrates for achieving a particular quality score compared to those of the other videos (as can be seen in Table II). Hence, it is more challenging to achieve higher QoE with limited dynamic bandwidth. The *Conan* video has higher visual quality levels than the *Football* and *Spotlight* videos since it has higher average tiles overlap. This indicates that content features and user interaction strongly influence the streaming performance of 360° videos. It can be noticed that all of the streaming methods achieve slightly higher performance for the 6x4 tiling pattern followed by the 8x6 tiling pattern for all four videos. This is because the higher viewport overlap and feasible segment sizes yield higher QoE levels.

The streaming results for 1s segment duration in dynamic bandwidth Scenario B1 illustrated in Fig. 6a show that CFOV achieves about 8.71%, 21.05%, 24.55%, and 27.74% higher



(a) Segment duration: 1s, Bandwidth: Scenario B1

(b) Segment duration: 2s, Bandwidth: Scenario B2

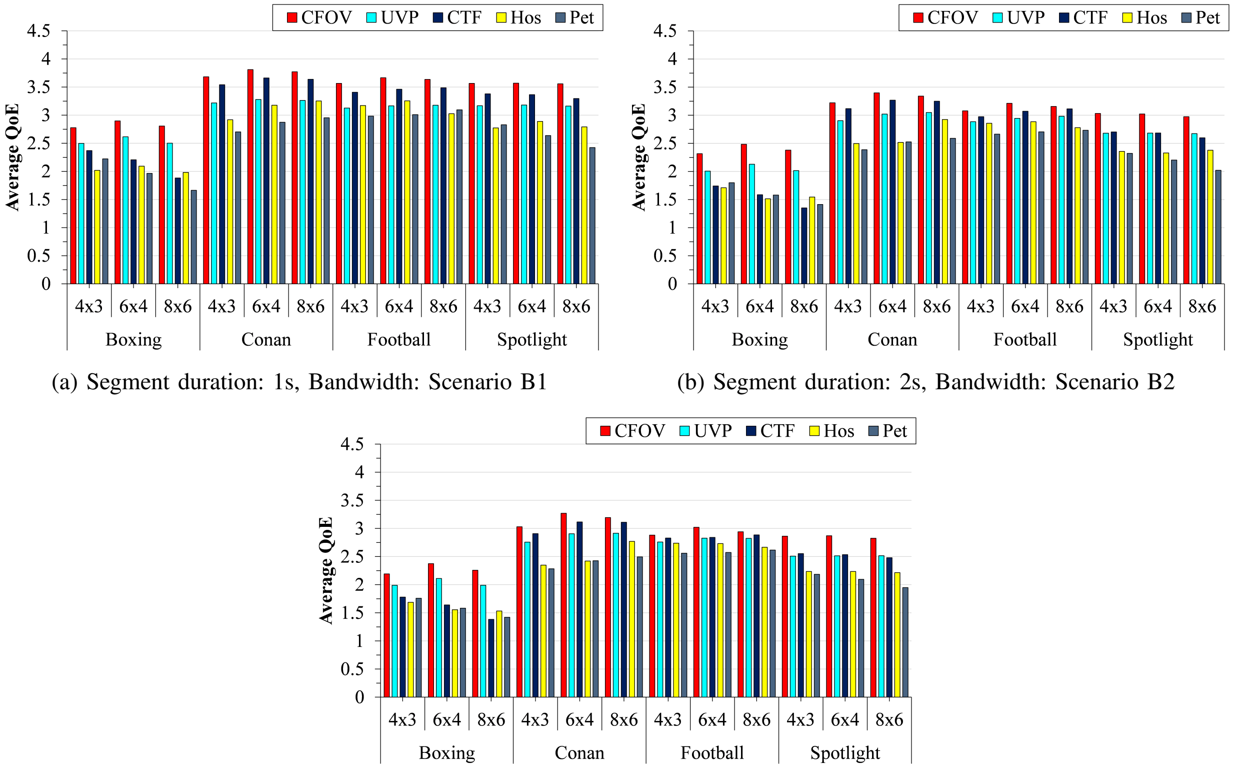
(c) Segment duration: 3s, Bandwidth: Scenario B3

Fig. 6: Average QoE results achieved by five streaming clients for 48 VR users with QoE weight coefficients set to C1; results from four videos prepared in three tiling patterns and three segment duration.

quality scores than UVP, CTF, Hos, and Pet algorithms, respectively, for the *Boxing* video. For *Conan*, the average gain over all the tiling patterns of UVP, CTF, Hos, and Pet is about 8.53%, 1.68%, 13.75%, and 19.60%, respectively. For different motion contents, viewport mismatch leads to high-quality degradation for tile-based methods. In CFOV, both the *Fixed FoV* and *Extended FoV* cases favor high-quality perception of the viewing area. Accordingly, CFOV achieves up to 7.86%, 5.51%, 18.51%, and 27.22% higher QoE for the *Spotlight* video compared to UVP 8x6, CTF 8x6, Hos 4x3, and Pet 8x6, respectively. The reason is that, instead of sending all the tiles at the lowest quality, CFOV implements an aggressive strategy by streaming only the viewport tiles at the maximum possible quality when the available bandwidth budget is limited.

Fig. 6b displays the average QoE values of the proposed solution and four reference methods for the 2s segment duration and stepwise switch-up bandwidth scenario B2. It is interesting to note that CFOV always has better performance than the reference methods. This is because our approach favors high viewport quality. Compared to UVP and CTF methods, CFOV can improve the average QoE by up to 12.57% and 30.83%, respectively, for the *Boxing* video. The average improvements in QoE over the Hos and Pet methods are 18.50% and 23.46% for the *Spotlight* video. For the *Conan* video with a 6x4 tiling pattern, CFOV achieves 3.64 on average QoE compared to the scores of 3.38, 3.55, 2.84, and 2.86 achieved by UVP, CTF, Hos, and Pet algorithms, respectively (Fig. 6b). Similar to the 1s and 2s cases, CFOV mostly achieves the highest average QoE value for all three tiling patterns when the segment duration is set to 3s (Fig. 6c). The *Football* video has the smallest average segment sizes; however, the head movement traces for this video contain significant variations in viewing directions. Therefore, the viewport-based methods achieve slightly lower QoE values than the *Conan* video for the bandwidth fluctuation scenario B3. The proposed method achieves up to 25.97% and 27% on average higher QoE values for the *Boxing* than Hos and Pet algorithms, respectively (Fig. 6c). The under-performance of the Hos and Pet methods, even under stable head movements, is mainly because they needlessly increase the quality of the adjacent tiles.

3) Average QoE with Coefficient Set C2: To better understand streaming approaches' performance, we increase the weights of the background quality and spatial and temporal quality oscillations penalties. Fig. 7 compares the average gains on QoE for different content types with the QoE weight coefficient set C2. Fig. 7a indicates that CFOV outperforms UVP, CTF, Hos, and Pet methods by 11.87%, 9.59%, 19.78%, and 24.45%, respectively, for all four videos across all three tiling patterns for 1s segment duration. CFOV is highly vulnerable to imperfect viewport prediction because only the viewport tiles can be streamed to the client when the network bandwidth is low. However, streaming a lower amount of data provides significant benefits for CFOV. Fig. 7b shows that for the 6x4 *Football* video, the average QoE values of CFOV, UVP, Hos, and Pet methods are 3.21, 2.94, 3.07, 2.88,



(a) Segment duration: 1s, Bandwidth: Scenario B1

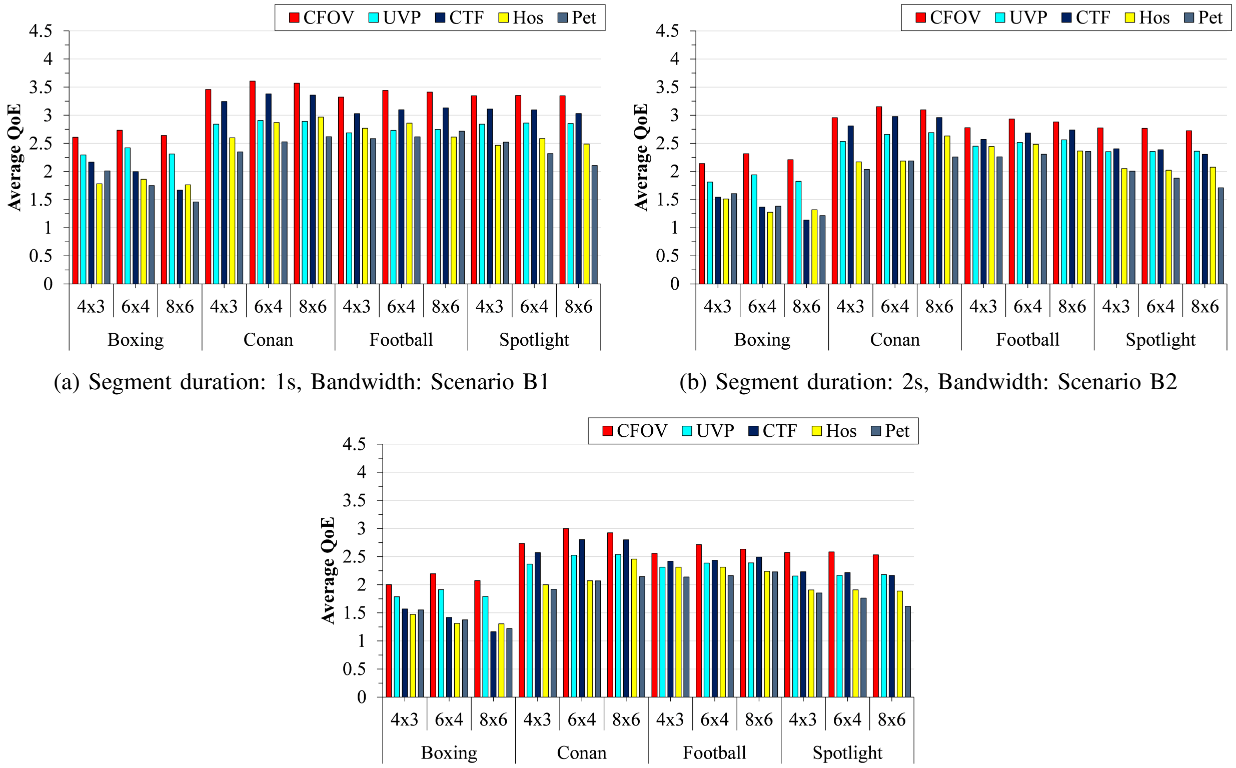
(b) Segment duration: 2s, Bandwidth: Scenario B2

(c) Segment duration: 3s, Bandwidth: Scenario B3

Fig. 7: Average QoE results achieved by five streaming clients for 48 VR users with QoE weight coefficients set to C2; results from four videos prepared in three tiling patterns and three segment duration.

and 2.7, respectively. CFOV improves QoE over UVP and CTF due to its higher prediction performance. CFOV provides improved performance over Hos and Pet methods because it reduces the amount of data to send and contributes to lower background quality. The average gain on QoE achieved by CFOV for all four videos with 4x3, 6x4, and 8x6 tiling patterns is about 28.88% (*Boxing*), 14.55% (*Conan*), 8.44% (*Football*), and 17.95% (*Spotlight*) higher than the others when the segment duration is set to 2s (Fig. 7b). It can be noticed that with the more considerable difference between the actual and predicted viewports, the average QoE in Fig. 7c is lower than what is depicted in Fig. 7a and Fig. 7b. Interestingly, all the comparative methods tend to download the highest quality levels when bandwidth is higher than the available quality levels. This leads to favor the QoE metric by lowering spatial and temporal quality variations. CFOV's average QoE is the highest for *Boxing* and *Spotlight* videos, followed by the UVP and CTF methods. This is due to the fact that CFOV preserves the highest visual quality by dealing effectively with the abrupt view switching during each adaptation interval. CTF leads to better results than UVP for the *Conan* and *Football* videos since it directly assigns the highest quality to the viewpoint tile based on the available bandwidth budget. Instead of completely relying on bandwidth, CFOV adjusts the viewport quality by dynamically deciding the coverage of the FoV. Moreover, CFOV incurs a lower bandwidth consumption without noticeable quality degradation by streaming background tiles at the lowest quality.

4) Average QoE with Coefficient Set C3: Next, the performance of CFOV and those of the other approaches are tested by setting the QoE weight coefficients to set C3. Fig. 8 illustrates the video quality experienced, averaged across the 48 users for the different videos and tile patterns. It can be noted that with the increase of background quality and spatial and temporal quality penalties, the average quality score in Fig. 8 is lower than what is depicted in Fig. 6 and Fig. 7. Fig. 8a shows that CFOV outperforms the existing streaming approaches by achieving 26.41%, 18.78%, 17.48%, and 19.64% on average QoE improvements for the *Boxing*, *Conan*, *Football*, and *Spotlight*l videos, respectively. In particular, CFOV improves the average QoE by up to 16.30% compared to the UVP, up to 12.51% compared to the CTF, up to 24.22% compared to the Hos, and up to 29.29% compared to the Pet method for the entire test dataset. Fig. 8b shows the average QoE comparison under Scenario B2 when the segment duration is set to 2s. CFOV experiences only 6.25% and 19.10% viewport deviation for the *Boxing* and *Conan* videos with a 6x4 tiling pattern (Fig. 5b); therefore, it efficiently utilizes the available bandwidth budget and achieves average QoE improvements of up to 16.25%, 41%, 44.95%, and 40.28% for the *Boxing* video, and up to 15.67%, 5.56%, 30.68%, and 30.64% for the *Conan* video, in comparison to the other four methods. Similarly, Fig. 8c shows that for the *Football* video with a segment duration of 3s, CFOV achieves with 10.29%, 7.01%, 13.11%, and 17.35% higher QoE in comparison to the UVP, CTF, Hos, and Pet methods, respectively. Similarly, for the *Spotlight* video,



(a) Segment duration: 1s, Bandwidth: Scenario B1

(b) Segment duration: 2s, Bandwidth: Scenario B2

(c) Segment duration: 3s, Bandwidth: Scenario B3

Fig. 8: Average QoE results achieved by five streaming clients for 48 VR users with QoE weight coefficients set to C3; results from four videos prepared in three tiling patterns and three segment duration.

CFOV has an effective QoE improvement between 14% and 32% in comparison with the other approaches.

C. Discussion

Most existing algorithms strive to balance different QoE objectives, i.e., viewport quality, background quality, spatial and temporal quality variations. For CTF and Hos algorithms, the primary factor leading to performance degradation is the per tile quality allocation starting from the center tile while sacrificing the quality of the remaining tiles. As a result, these algorithms struggle with the user-perceived quality and visual smoothness objectives. UVP allocates bitrate for tiles belonging to the same classification based on the estimated bandwidth to reduce the spatial and temporal quality objectives. However, for segment duration >2s, the visible tiles' rate is reduced substantially due to the limited prediction accuracy (58~78%), leading to inefficient bandwidth utilization. The Pet method has significantly lower QoE values than the other solutions under stable and drastic head rotations. This is because the invisible tiles consume an essential share of the bandwidth. Contrary, our proposed solution always results in a higher QoE than the alternative methods for all VR users. CFOV sends much less data for the background tiles than the other algorithms; therefore, it results in a lower background quality penalty for different viewport prediction results. Under variable head movement traces, the *Extended FoV* or external tiles of the proposed method provide improved QoE for different videos across all tiling patterns. In conclusion, the CFOV delivery of 360° videos is better than when the other benchmark methods are employed.

VI. Conclusions

This paper presents CFOV, an innovative adaptive 360° video streaming solution which improves end-user QoE. In the context of quality-efficient 360° remote video services, CFOV reduces the complexity of tile selection by adopting two FoV prediction mechanisms to better accommodate the user's viewing region in response to the different head movements. In addition, CFOV performs active and improved region-wise bitrate allocations for selected tiles without incurring unnecessary bandwidth consumption. An extensive experimental assessment was performed using four video streams prepared in three tiling patterns and three segment durations under three dynamic bandwidth scenarios. The experimental results show that CFOV achieves with 9.28% higher average viewport overlap and between 12.74% and 21.5% higher average QoE than the other solutions under different testing scenarios.

References

[1] M. Postgate, "BBC announces live Ultra HD and VR trials for World Cup," 2018. [Online]. Available: https://www.bbc.co.uk/mediacentre/latestnews/2018/uhd-vr-world-cup

[2] Cisco Systems Inc., "Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper," *Cisco Forecast and Methodology*, 2019.

[3] T. Biatek, W. Hamidouche, P. Cabarat, J. Travers, and O. Déforges, "Scalable Video Coding for Backward-Compatible 360° Video Delivery Over Broadcast Networks," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 322–332, 2020.

[4] M. Fiedler, H.-J. Zepernick, and V. Kelkkanen, "Network-Induced Temporal Disturbances in Virtual Reality Applications," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2019, pp. 1–3.

[5] M. Abrash, "What VR could, should, and almost certainly will be within two years," *Steam Dev Days, Seattle*, vol. 4, p. 2014, 2014.

[6] C. Ozcinar, J. Cabrera, and A. Smolic, "Visual Attention-Aware Omnidirectional Video Streaming using Optimal Tiles for Virtual Reality," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 217–230, March 2019.

[7] S. Jia, C. Xu, J. Guan, H. Zhang, and G. Muntean, "A Novel Cooperative Content Fetching-Based Strategy to Increase the Quality of Video Delivery to Mobile Users in Wireless Networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 370–384, 2014.

[8] A. Yaqoob, T. Bi, and G.-M. Muntean, "A DASH-based Efficient Throughput and Buffer Occupancy-based Adaptation Algorithm for Smooth Multimedia Streaming," in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2019, pp. 643–649.

[9] C. Yao, J. Xiao, Y. Zhao, and A. Ming, "Video Streaming Adaptation Strategy for Multiview Navigation Over DASH," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 521–533, 2019.

[10] G.-M. Muntean, G. Ghinea, and T. N. Sheehan, "Region of Interest-based Adaptive Multimedia Streaming Scheme," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 296–303, 2008.

[11] B. Ciubotaru, G. Ghinea, and G.-M. Muntean, "Subjective Assessment of Region of Interest-Aware Adaptive Multimedia Streaming Quality," *IEEE Trans. on Broadcasting*, vol. 60, no. 1, pp. 50–60, Mar. 2014.

[12] S. Chen, Z. Yuan, and G. Muntean, "An Energy-Aware Routing Algorithm for Quality-Oriented Wireless Video Delivery," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 55–68, 2016.

[13] R. Trestian, O. Ormond, and G. Muntean, "Energy–Quality–Cost Tradeoff in a Multimedia-Based Heterogeneous Wireless Network Environment," *IEEE Transactions on Broadcasting*, vol. 59, no. 2, pp. 340–357, 2013.

[14] A. Hava, Y. Ghamri-Doudane, J. Murphy, and G.-M. Muntean, "A Load Balancing Solution for Improving Video Quality in Loaded Wireless Network Conditions," *IEEE Transactions on Broadcasting*, vol. 65, no. 4, pp. 742–754, 2019.

[15] G.-M. Muntean and N. Cranley, "Resource Efficient Quality-Oriented Wireless Broadcasting of Adaptive Multimedia Content," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 362–368, March 2007.

[16] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, and D. Eager, "The prefetch aggressiveness tradeoff in 360 video streaming," in *Proc. 9th ACM Multimedia Systems Conference*, 2018, pp. 258–269.

[17] T. C. Nguyen and J.-H. Yun, "Predictive Tile Selection for 360-Degree VR Video Streaming in Bandwidth-limited Networks," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1858–1861, 2018.

[18] C. Zhou, Z. Li, and Y. Liu, "A measurement Study of Oculus 360 Degree Video Streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 2017, pp. 27–37.

[19] Y. Sánchez, R. Skupin, and T. Schierl, "Compressed Domain Video Processing for Tile based Panoramic Streaming using HEVC," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2244–2248.

[20] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 601–605.

[21] R. Aksu, J. Chakareski, and V. Swaminathan, "Viewport-driven Rate-distortion Optimized Scalable Live 360° Video Network Multicast," in *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2018, pp. 1–6.

[22] S. Rossi and L. Toni, "Navigation-aware Adaptive Streaming Strategies for Omnidirectional Video," *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2017.

[23] A. Ghosh, V. Aggarwal, and F. Qian, "A Rate Adaptation Algorithm for Tile-based 360-degree Video Streaming," *arXiv e-prints*, p. arXiv:1704.08215, Apr. 2017.

[24] M. Graf, C. Timmerer, and C. Mueller, "Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP: Design, Implementation, and Evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 261–271. [Online]. Available: http://doi.acm.org/10.1145/3083187.3084016

[25] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 99–114.

[26] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware Adaptive 360 Video Streaming using Tiles for Virtual Reality," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2174–2178.

[27] Y. Hu, Y. Liu, and Y. Wang, "VAS360: QoE-Driven Viewport Adaptive Streaming for 360 Video," in *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2019, pp. 324–329.

[28] D. V. Nguyen, H. T. Tran, A. T. Pham, and T. C. Thang, "An Optimal Tile-Based Approach for Viewport-Adaptive 360-Degree Video Streamings," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 29–42, 2019.

[29] M. Hosseini and V. Swaminathan, "Adaptive 360 VR Video Streaming: Divide and Conquer," in *Multimedia (ISM), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 107–110.

[30] 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.

[31] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive Navigable 360-degree Video Delivery," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[32] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very Long Term Field of View Prediction for 360-degree Video Streaming," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 297–302.

[33] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal Bitrate Adaptation for Online Videos," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[34] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados, and P. Chatzimisios, "FDASH: A Fuzzy-based MPEG/DASH Adaptation Algorithm," *IEEE Systems Journal*, vol. 10, no. 2, pp. 859–868, 2016.

[35] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 197–210.

[36] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[37] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.

[38] S. Q. Jabbar, D. J. Kadhim, and Y. Li, "Proposed an Adaptive Bitrate Algorithm based on Measuring Bandwidth and Video Buffer Occupancy for Providing Smoothly Video Streaming."

[39] J. V. d. Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. D. Turck, "Tile-based Adaptive Streaming for Virtual Reality Video," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–24, 2019.

[40] S.-C. Yen, C.-L. Fan, and C.-H. Hsu, "Streaming 360° Videos to Head-mounted Virtual Reality using DASH over QUIC Transport Protocol," in *Proceedings of the 24th ACM Workshop on Packet Video*. ACM, 2019, pp. 7–12.

[41] A. Mavlankar and B. Girod, "Video Streaming with Interactive Pan/Tilt/Zoom," in *High-Quality Visual Experience*. Springer, 2010, pp. 431–455.

[42] Y. Sanchez, G. S. Bhullar, R. Skupin, C. Hellge, and T. Schierl, "Delay Impact on MPEG OMAF's Tile-based Viewport-dependent 360° Video Streaming," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2019.

[43] R. Azuma and G. Bishop, "A Frequency-domain Analysis of Head-motion Prediction," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 401–408.

[44] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "Cub360: Exploiting Cross-users Behaviors for Viewport Prediction in 360 Video Adaptive Streaming," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[45] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello, "360° Innovations for Panoramic Video Streaming," in *ACM Workshop on Hot Topics in Networks*. ACM, 2017, pp. 50–56.

[46] S. Petrangeli, G. Simon, and V. Swaminathan, "Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos," in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2018, pp. 157–160.

[47] L. Xie, X. Zhang, and Z. Guo, "CLS: A Cross-User Learning Based System for Improving QoE in 360-Degree Video Adaptive Streaming," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 564–572. [Online]. Available: https://doi.org/10.1145/3240508.3240556

[48] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a Moving Target: Motion-prediction-based Transmission for 360-degree Videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[49] M. Jamali, S. Coulombe, A. Vakili, and C. Vazquez, "LSTM-based Viewpoint Prediction for Multi-Quality Tiled Video Coding in Virtual Reality Streaming," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[50] F. Duanmu, E. Kurdoglu, S. A. Hosseini, Y. Liu, and Y. Wang, "Prioritized Buffer Control in Two-tier 360 Video Streaming," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2017, pp. 13–18.

[51] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An HTTP/2-based Adaptive Streaming Framework for 360 Virtual Reality Videos," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 306–314.

[52] A. Yaqoob, T. Bi, and G. M. Muntean, "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[53] X. Jiang, Y.-H. Chiang, Y. Zhao, and Y. Ji, "Plato: Learning-based adaptive streaming of 360-degree videos," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. IEEE, 2018, pp. 393–400.

[54] X. Corbillon, F. De Simone, and G. Simon, "360-degree Video Head Movement Dataset," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 199–204.

[55] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[56] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[57] C. Zhou, M. Xiao, and Y. Liu, "ClusTile: Toward Minimizing Bandwidth in 360-degree Video Streaming," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018, pp. 962–970.

[58] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 Video Streaming using Tile-based HTTP Adaptive Streaming," *Proceedings of the ACM Multimedia Conference*, pp. 315–323, 2017. [Online]. Available: 10.1145/3123266.3123291

[59] D. V. Nguyen, H. T. T. Tran, A. T. Pham, and T. C. Thang, "An Optimal Tile-Based Approach for Viewport-Adaptive 360-Degree Video Streaming," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 29–42, 2019.

[60] D. He, C. Westphal, and J. Garcia-Luna-Aceves, "Joint Rate and FoV Adaptation in Immersive Video Streaming," in *ACM Sigcomm workshop on AR/VR Networks*, 2018.

[61] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "HEVC Tile based Streaming to Head Mounted Displays," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 613–615.

[62] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree Video Streaming using Scalable Video Coding," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1689–1697.

[63] D. Salomon, *Transformations and projections in computer graphics*. Springer Science & Business Media, 2007.

[64] N. Eswara, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "Perceptual QoE-Optimal Resource Allocation for Adaptive Video Streaming," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 346–358, 2020.

[65] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Rubiks: Practical 360-Degree Streaming for Smartphones," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. New York, NY, USA: ACM, 2018, pp. 482–494.

[66] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "DRL360: 360-degree Video Streaming with Deep Reinforcement Learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1252–1260.

[67] A. S. Fernandes and S. K. Feiner, "Combating VR Sickness Through Subtle Dynamic Field-of-View Modification," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2016, pp. 201–210.

[68] A. Yaqoob and G.-M. Muntean, "A Weighted Tile-based Approach for Viewport Adaptive 360° Video Streaming," in *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2020, pp. 1–7.

[69] M. Fabian Romero Rondon, L. Sassatelli, R. Aparicio Pardo, and F. Precioso, "Revisiting Deep Architectures for Head Motion Prediction in 360° Videos," *arXiv e-prints*, p. arXiv:1911.11702, Nov. 2019.

[70] V. H. Muntean and G.-M. Muntean, "A Novel Adaptive Multimedia Delivery Algorithm for Increasing User Quality of Experience during Wireless and Mobile E-learning," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, May 2009, pp. 1–6.

[71] T. C. Thang, Q.-D. Ho, J. W. Kang, and A. T. Pham, "Adaptive Streaming of Audiovisual Content using MPEG DASH," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78–85, 2012.

[72] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 193–198.

[73] A. Nguyen, Z. Yan, and K. Nahrstedt, "Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 1190–1198.

[74] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Hämäläinen, "Kvazaar: Open-Source HEVC/H.265 Encoder," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016. [Online]. Available: http://doi.acm.org/10.1145/2964284.2973796

[75] A. T. Nasrabadi, A. Samiei, and R. Prakash, "Viewport Prediction for 360° Videos: A Clustering Approach," in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2020, p. 34–39.

Abid Yaqoob (S'20) received the B.Sc. degree in computer systems engineering from the Islamia University of Bahawalpur, Pakistan, in 2014, and the M.Sc. degree in network and information security from Northwestern Polytechnical University, Xi'an, China in 2018. He is currently pursuing a Ph.D. degree with the Performance Engineering Laboratory and the Insight Centre for Data Analytics, School of Electronic Engineering, Dublin City University, Ireland. His research interests include mobile wireless communication, priority-aware multiview video streaming, as well as immersive multimedia processing and delivery solutions.

Gabriel-Miro Muntean (M'04, SM'17) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and Co-Director of the DCU Performance Engineering Lab. Prof. Muntean was awarded the Ph.D. degree by DCU for research on adaptive multimedia delivery in 2004. He has published over 450 papers in top-level international journals and conferences, authored four books and 25 book chapters, and edited six additional books. His research interests include quality, performance, and energy-saving issues related to multimedia and multiple sensorial media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. Prof. Muntean is an Associate Editor of the IEEE Transactions on Broadcasting, the Multimedia Communications Area Editor of the IEEE Communications Surveys and Tutorials.