

Received February 11, 2022, accepted March 2, 2022, date of publication March 8, 2022, date of current version March 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3157339

Dynamic Viewport Selection-Based Prioritized Bitrate Adaptation for Tile-Based 360° Video Streaming

ABID YAQOUB^{ID}, (Student Member, IEEE), MOHAMMED AMINE TOGOU^{ID}, (Member, IEEE),
AND GABRIEL-MIRO MUNTEAN^{ID}, (Senior Member, IEEE)

Insight Centre for Data Analytics, Dublin City University, Dublin 9, D09 Y5N0 Ireland

Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, D09 Y5N0 Ireland

Corresponding author: Gabriel-Miro Muntean (gabriel.muntean@dcu.ie)

This work was supported by the European Regional Development Fund through the Science Foundation Ireland (SFI) Research Centres Programme under Grant 12/RC/2289_P2 (Insight SFI Centre for Data Analytics) and Grant 16/SP/3804 (ENABLE).

ABSTRACT Interactive 360° remote video applications have seen booming advancements due to the proliferation of smart display devices that enable a truly immersive experience. Compared to regular monoscopic videos, 360° videos have different requirements related to content preparation, packaging, transmission, specialized viewing equipment, and display factors (e.g., brightness, contrast, delay, frame rate, resolution, image quality, etc). In addition, 360° video requires substantial network and computational resources, which are challenging to achieve with conventional transmission and rendering infrastructure. Viewport-adaptive streaming is a common way to ensure visual quality under limited bandwidth resources. However, identifying, extracting, and rendering the true viewport in response to drastic head rotations can adversely affect user experience. This paper proposes two dynamic viewport selection approaches, which adapt the streamed regions based on content complexity variations and positional information to ensure viewport availability and smooth visual angles for VR users. They incorporate content information as well as user head movement patterns to support tile-based prioritized 360° video streaming. Moreover, a practical, prioritized bitrate adaptation approach, which requests selected tiles at appropriate quality levels, is also proposed to reduce the impact of inefficient bandwidth utilization in the VR scene. Experimental evaluations under real 4G bandwidth logs demonstrate that the proposed solutions outperform the closest state-of-the-art algorithms across multiple performance metrics, i.e., viewport overlap, perceived quality levels, quality fluctuations, and viewport bandwidth utilization.

INDEX TERMS 360° video streaming, viewport prediction, tiles selection, viewport adaptation.

I. INTRODUCTION

Recently, next-generation immersive multimedia applications such as Virtual Reality (VR) and 360° videos have increasingly penetrated various fields, including entertainment, healthcare, education, real estate, manufacturing, retail, transportation, sport, and other consumer-facing services. Specifically, 360° video users can now experience a more personalized viewing perspective by freely navigating through the captured content on portable devices such as the Oculus Rift/Quest, HTC Vive, Sony PlayStation VR, Samsung

Gear VR, etc. Fig. 1¹ illustrates a spherical video viewing environment for 360° video. 360° videos, already famous on popular video-sharing platforms such as YouTube and Facebook, exhibit enormous potential for enterprises and consumers by providing more engaging and unprecedented interaction between viewers and the content [1]. However, to perceive a sense of “presence there,” the displayed information must be consistent during playback, while bandwidth consumption must be low. Therefore, efficient delivery of ultra-high resolution panoramic content is very important.

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Nappi^{ID}.

¹The equirectangular image used is sourced from GoPro VR-Tahiti Surf video available at <https://www.youtube.com/watch?v=MKWWhf8RAV8>.

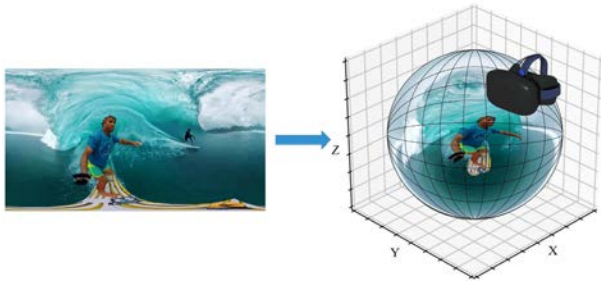


FIGURE 1. 360° video viewing using a head-mounted display device. The current generation three degrees of freedom (3DoF) viewing allows the user to freely navigate across yaw, pitch, and roll directions.

The ever-increasing immersive real-time rendering [2] causes tremendous bandwidth consumption since the entire captured content, including the visible and non-visible scenes, has to be downloaded in higher resolution ($\geq 8k$) [3], [4]. Viewport-adaptive streaming is a straightforward approach to reduce the transmission bandwidth, where the quality of the scenes is dynamically adjusted in real-time depending on the user's Field-of-View (FoV) [5], [6]. Among viewport-based solutions, tile-based adaptive streaming is now a widely embraced solution to support transmitting quality-variable tiles without sacrificing visual quality in response to the user's interaction. The core concept of tile-based adaptive streaming is to perform spatial and temporal adaption by considering the human viewing behavior and network conditions.

The space (tile-level) and time (segment-level) separation of 360° videos makes it challenging to achieve higher QoE levels compared to regular videos. Existing techniques struggle to perform reliable viewport prediction, streaming tiles selection, and spatial and temporal bitrate adaptations. A straightforward technique is to predict viewpoint based on a single prediction mechanism and then calculate the viewing area by mapping the tiles to the horizontal and vertical lengths of the viewport as adopted in [6]–[10]. Such solutions work reasonably well under 1s segment duration with an accuracy of 58~80% [11], [12]. However, unfair visual quality levels and frequent quality fluctuations can be observed when tested for longer segment duration ($> 1s$) due to the higher viewport mismatch ratio [13], [14]. Some solutions [15]–[17] consider prediction angles as well as prediction errors to estimate the viewport. However, they do not involve any neighbor region [16] or viewport extension [15], [17] to truly reflect the proactive streaming. Moreover, including neighboring tiles surrounding viewport in all directions [18] or sub-areas-based streaming [15] can lead to significant bandwidth wastage. Therefore, careful selection of neighboring tiles and dynamic extension of the viewport is important but not simple. Similarly, tiled bitrate adaptation without respecting the tiles' spatial distribution and users' interests can generate results similar to viewport-independent streaming based on conventional bitrate adaptation algorithms [19]–[24].

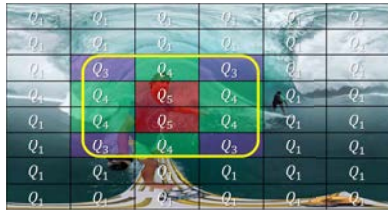
This paper proposes two innovative dynamic viewport selection (DVS) solutions for 360° adaptive video

streaming, referred to as DVS1 and DVS2. The proposed solutions provide support for improved streaming tiles selection and bitrate adaptation based on content complexity and user engagement. Existing works [6]–[10], [15]–[17] only consider the viewing information to perform relevant tiles selection. Unlike these schemes, DVS solutions include content information as well as viewing behaviors to proactively select the streaming regions. Different from [15], [18], DVS solutions do not include the neighboring regions throughout the streaming session (evidenced by Table 5). The extended viewport in DVS1 and DVS2 solutions is more profound than [8], [25], [26] and can accommodate highly variable head rotations. The main contributions of this work are as follows:

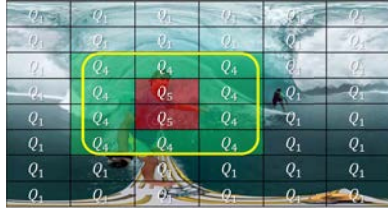
- **Viewport Prediction:** Unlike the existing viewport prediction solutions, which use a single viewpoint prediction mechanism, the DVS1 proposed prediction model employs two low-complex prediction mechanisms to address the accuracy gap, less studied in the research literature. DVS2 surpasses the earlier works by jointly employing in its mechanism prediction angles and prediction errors recorded from experimental results.
- **Streaming Tiles Selection:** Given the viewport prediction results, DVS solutions examine how to best select the tiles to cover the actual viewing window, considering the variations in content complexity and predicted viewing positions. DVS solutions perform advanced streaming tiles selection in the context of three scenarios: (1) *Fixed viewport*; (2) *Neighbor region*; and (3) *Extended viewport* in order to improve the 360° video content visualization.
- **Spatial and Temporal Bitrate Adaptation:** The proposed client-based bitrate adaptation algorithm is designed to maximize the perceived quality levels for various indoor and outdoor recorded streams. Based on the spatial tiles distribution, diverse quality allocation scenarios were considered: (1) *Uniform*; (2) *Weighted*; and (3) *Non-uniform*, to determine the best possible quality levels in variable network and user viewing conditions.

The proposed solutions are evaluated by employing three videos scenes content from performance, sport, and action-film categories, prepared in 4×3 , 6×4 , and 8×6 tiling patterns and 1s and 2s segment duration. The behavior of the proposed and reference approaches is examined under distinct head movement patterns from 48 real-VR users. The effect of tiles resolution and segment duration on algorithms' performance is also explored. Simulations-oriented results show that our proposed solution outperforms existing works by achieving higher bandwidth utilization and higher visual quality levels while reducing the quality fluctuations within the viewport.

The rest of this paper is organized as follows: most recent adaptive streaming solutions involving viewport prediction mechanisms are discussed in Section II. The proposed viewport selection and bitrate adaptation methods are presented



(a) Fixed viewport [6].



(b) Fixed viewport [10].

FIGURE 2. Tiles arrangement and quality adaptation for fixed viewport-based adaptive streaming solutions in an equirectangular space.

(a) Neighbor region [18].



(b) Neighbor region [15].

FIGURE 3. Tiles arrangement and quality adaptation for neighbor region-based adaptive streaming solutions in an equirectangular space.

in Section III. Section IV describes the experimental testing setup and performance analysis performed for different tile-based streaming solutions. Finally, section V includes the paper's conclusive remarks and highlights some potential future research avenues.

II. BACKGROUND AND RELATED WORKS

Viewport prediction, streaming tiles selection, and bitrate adaptation are sequential procedures for adaptive 360° video. Recently, omnidirectional multimedia researchers have focused on selecting the accurate viewport representation and quality levels. Based on the viewport size and division of different streaming tiles, we have classified tile-based adaptive streaming into three main categories: (1) Fixed viewport-based streaming; (2) Neighbor region-based adaptive streaming; (3) Extended viewport-based

adaptive streaming. Next, the most representative schemes in these categories are discussed.

A. FIXED VIEWPORT-BASED STREAMING

In fixed viewport-based adaptive streaming, a fixed-size viewport field is employed to render the high-quality viewport (usually 1/3 portion of the whole stream) regarding the user's dynamic movement patterns. The basic idea is to map the tiles to the actual viewport coverage of the display terminal at higher quality levels. At the same time, the remaining portion of the sphere (2/3 or so) can be streamed at a relatively lower quality. Hooft *et al.* [8] proposed to stream 360° tiles into two groups, i.e., viewport and non-viewport, by using prediction angles from the *spherical walk* prediction method. The authors introduced a uniform viewport (UVP) quality allocation version to stream the uniform viewport and non-viewport tiles in a receiver-centric adaptive streaming architecture. However, this work does not consider prediction angles for viewport selection, and the proposed solution was tested using a single segment duration.

Chen *et al.* [6] introduced different priority levels within the viewport, i.e., the highest priority for the center tile, medium priority for the top, bottom, left, and right tiles, and the lowest priority for the four diagonal neighbor tiles. Different encoding versions or non-uniform quality levels are assigned to the viewport tiles (Fig. 2a) based on their priority levels to reduce the bandwidth and storage capacity. However, this work was evaluated in comparison to a straightforward non-adaptive streaming scheme under fixed network connections only. Similarly, Hosseini and Swaminathan [10] introduced a priority-based bitrate adaptation approach for tiles belonging to three zones, i.e., Z_1 (viewport tile), Z_2 (viewport surrounding tiles), and Z_3 (background tiles), based on user interest. Fig. 2b shows that the tiles in the selected viewport are allocated with different quality versions. The viewport tiles in Z_1 and Z_2 are allocated with higher bitrates (Q_5 and Q_4) than the tiles in Z_3 (Q_1). However, the VR setup was implemented using 2K viewing resolution. Moreover, the videos were encoded using H.264/AVC encoder, which can decrease the coding performance for 360° videos. Xie *et al.* [9] proposed a controlled buffer-based probabilistic bitrate adaptation for the fixed viewport tiles to improve the overall visual quality. The performance was evaluated considering a single low-resolution video stream only.

B. NEIGHBOR REGION-BASED STREAMING

In neighbor region-based adaptive streaming, a spatial extension is defined around the viewport to deal with possible head movement prediction errors. The coverage of the neighbor regions could be in horizontal or vertical directions by defining mapping on the left and right or top and bottom sides of the viewport to manage the horizontal or vertical head movements, respectively. Fig. 3 represents the neighboring tiles selection for three works. Petrangeli *et al.* [18] proposed a region-based adaptive streaming framework involving

TABLE 1. Viewport selection solutions in tile-based adaptive 360° video streaming.

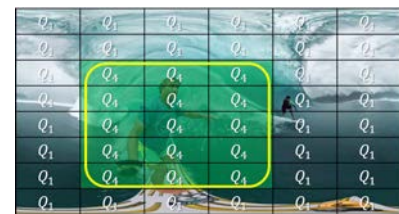
Works	Prediction Angles		Viewport Selection		Streaming Region			Experimental Evaluations					
	Single	Combined	Prediction Angles	Prediction Errors	Fixed VP	NR	Extended VP	Videos	VR Traces	Resolution	Segment Duration	Tiling	Exp. Duration
[27]	✓	×	✓	✓	✓	×	×	1	2	2K	1s	24 tiles	70s
[7]	✓	×	✓	×	✓	×	×	2	8	8K	2s	8, 10, 12 tiles	Video Duration
[6]	✓	×	✓	×	✓	×	×	5	50	2K	1s	3x3, 4x4, 5x5	20s
[8]	✓	×	✓	×	✓	×	✓	3	48	4K	1.067s	1x1, 2x2, 4x2, 4x4, 8x4, 8x6/8, 16x12/16	Video Duration
[9]	✓	×	✓	×	✓	×	×	1	5	2K	1s	6x12	3m
[10]	✓	×	✓	×	✓	×	×	5	1	720p-4K	-	6 tiles	Video Duration
[18]	✓	×	✓	×	✓	×	×	1	10	8K	1s, 2s, 4s	6 tiles	60s
[28]	✓	×	✓	×	✓	×	×	10	50	4K	2s	6x4	1m
[15]	✓	×	✓	✓	✓	✓	×	5	7	4K	1s	8x8	59.7s
[16]	✓	×	✓	✓	✓	×	✓	4	20	-	1s	8x4, 12x4	Video Duration
[17]	✓	×	✓	✓	✓	✓	×	3	153	8K	1s	1024px1024p, 512px512p, 256x256	36s
[29]	✓	×	✓	×	✓	×	✓	1	1	4K	1s	16x8	80s
[30]	✓	×	✓	×	✓	×	✓	-	-	-	-	-	-
DVS1	×	✓	✓	×	✓	✓	✓	3	48	4K	1s, 2s	4x3, 6x4, 8x6	Video Duration
DVS2	×	✓	✓	✓	✓	✓	✓	3	48	4K	1s, 2s	4x3, 6x4, 8x6	Video Duration

viewport prediction modules and tiles quality selection facilitated by a network module to improve the streaming performance of tiled videos in a high latency network environment. Neighbor region includes all the viewport surrounding tiles as shown in Fig. 3a. A uniform quality streaming was implemented for tiles belonging to the viewport, neighbor, and background regions. The viewport traces were collected using the Gear VR framework for one 360° video watched by ten users. The authors considered three encoding levels and three segment duration for tiled and non-tiled versions of the same video. However, this work considered only 60s long single 360° video for the performance evaluation.

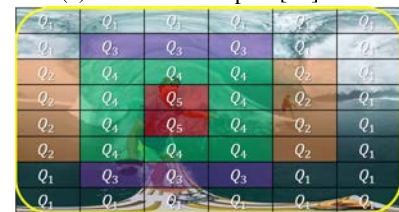
Nguyen *et al.* [15] proposed an adaptation mechanism that dynamically decides the viewport bitrate considering both the prediction angles and prediction errors observed during the previous segment. The proposed system includes some neighboring tiles to limit the viewport quality loss events. The neighboring tiles surrounding the viewport in all directions are divided into different sub-areas (Fig. 3b) depending on the viewport prediction performance. The experimental evaluations with five different motion content demonstrate an increase in the viewport quality without excessive bandwidth utilization for non-viewport regions. However, the experiments were performed for 60s long streaming sessions only.

C. EXTENDED VIEWPORT-BASED STREAMING

The main idea of extended viewport-based streaming is to virtually extend the original viewport coverage to include a larger viewing region to balance the actual and predicted tiles' overlap and visual quality. Instead of adding neighboring tiles, the viewport coverage is extended by 10%-30% or so, depending on the different decision factors, i.e., viewport errors, network conditions, etc. Only the higher-quality viewport tiles or higher-quality viewport and remaining tiles can be rendered remotely in such a scenario. Fig. 4 shows the tiles classification for extended viewport case. From 1° to 30° viewport extension is considered by Hu *et al.* [25] based on observed network conditions (Fig. 4a). In another study [16], the authors considered both predicted coordinates and prediction errors to dynamically adapt the area of viewing scenes. The viewport prediction was performed using the weighted



(a) Extended viewport [25].



(b) Extended viewport [8].

FIGURE 4. Tiles arrangement and quality adaptation for extended viewport-based adaptive streaming solutions in an equirectangular space.

linear regression (WLR) model [31]. A throughput-based bitrate adaptation was carried out to maximize the visual quality levels and bandwidth utilization. The authors employ the open-source dataset [32] to verify the effectiveness of the proposed solution. However, the experiments were not performed for all 48 VR users in the dataset.

Hoofit *et al.* [8] proposed the center tile first (CTF) scheme, which mainly focuses on the quality improvements starting from the center tile similar to [6], [10]. CTF includes all the tiles as the potential viewport tiles set for each adaptation interval as represented in Fig. 4b. He *et al.* [30] performed a network response-based joint adaptation of the viewport coverage and bitrate under dynamic and congested network conditions. The proposed algorithm firstly measures the round trip time (RTT) and then computes the FoV size and allocates the necessary bitrate based on the delay measurements. NS-3 based simulation outcomes confirm that adaptable viewport coverage offers improved quality streaming instead of relying on predicted points and bandwidth for quality adjustments. However, the proposed solution adapts in response to predefined delay settings only, and its performance still needs to be

investigated under complex network conditions. In addition, this work does not provide any information about content characteristics adopted for experimental evaluations.

Despite the extensive research and benefit of those techniques, there is still potential for further enhancing the viewport-based streaming in response to the variable watching behavior of the user. The proposed solutions can improve the personalized viewing experience by dynamically deciding the best-fit viewport versions. Table 1 highlights the significant differences between the proposed and existing viewport selection solutions. Most of the existing works have limitations in terms of performance evaluations, such as the number of videos, segment duration, tiling patterns, etc. We compare the proposed solutions against the UVP, CTF [8], HOS [10], and PET [18] methods using three different motion videos prepared in different tiling patterns and segment lengths, which were shown to outperform others.

III. PROPOSED ARCHITECTURE AND ALGORITHMS

A. SYSTEM ARCHITECTURE

The proposed solutions focus on improving the streaming performance of 360° VR video by requesting more than necessary tiles at the best possible resolutions. Fig. 5 presents the 360° client-server architecture for the DVS client. It includes a server that stores processed tiled video segments, a network that transmits the requested segments and a client which controls the viewport selection and bitrate adaptation processes. The spherical scene is projected into 2D planar format, i.e., equirectangular format [33]. The transformed video content is divided into several segments, i.e., $\mathcal{S} = \{\mathcal{S}(1), \mathcal{S}(2), \dots, \mathcal{S}(i), \dots, \mathcal{S}(I)\}$. The segmented videos are further divided into a number of rectangular grids/tiles, i.e., $\mathcal{T} = \{\mathcal{T}^1(i), \mathcal{T}^2(i), \dots, \mathcal{T}^k(i), \dots, \mathcal{T}^K(i)\}$. The tiled segments are then prepared into multiple quality representations, i.e., $\mathcal{L} = \{\mathcal{L}_1^k(i), \mathcal{L}_2^k(i), \dots, \mathcal{L}_j^k(i), \dots, \mathcal{L}_J^k(i)\}$. Let $\mathcal{L}_j^k(i)$ be the j th bitrate for k th tile in (i) th segment.

The viewport prediction module estimates the future viewing coordinates based on two sophisticated prediction mechanisms. The viewport selection module performs suitable tiles selection based on estimated viewport positions and content information. In this way, tiles belonging to different priority regions, i.e., viewport ($\mathcal{T}^v(i)$), neighboring ($\mathcal{T}^n(i)$), and background ($\mathcal{T}^b(i)$), are selected for each segment. Subsequently, the bitrate adaptation module performs the bitrate choice for tiles based on their distribution and the available network bandwidth. The server transmits the HEVC bitstreams to the VR client, which then decodes and renders the requested tiles to project the 360° content.

B. VIEWPORT PREDICTION AND TILES SELECTION

This section presents streaming tiles selection based on two mechanisms: (i) prediction angles-based tiles selection; (ii) prediction error and prediction angles-based tiles selection. Next, these innovative tiles selection approaches are discussed in details.

1) PREDICTION ANGLES-BASED TILES SELECTION

Accurate viewport prediction plays a significant role in viewport selection, leading to improved streaming performance. Tile-based streaming solutions should be able to select the best-fit tiles in response to the variable head motions of the consumer. The existing prediction models based on a single prediction mechanism, for instance, position-based (i.e., LR, speed-based, machine learning-based, etc.) and complex content-based (i.e., saliency maps, motion maps, etc.), could undergo lower long-term prediction accuracy. Interestingly, the *last known* prediction model that simply outputs the current viewing position for the future segments performs better than the state-of-the-art viewport prediction models [34]. In 360° viewing, the head rotations of the user can be recorded and represented by the Euler angles ($\Phi(i)$, $\Theta(i)$, $\Psi(i)$). Bao et al. [35] showed that these three angles have negligible cross-correlations and strong auto-correlations. Therefore, these angles can be predicted independently.

Due to the limited prediction accuracy of existing solutions and the extremely unpredictable viewing nature of the user, the basic idea is to utilize two prediction mechanisms to refine the tiles selection to cover the actual viewing area. For each video segment, DVS clients classify a 360° video frame into the viewport, neighbor, and background regions. Following this, we consider *Fixed viewport*, *Neighbor region*, and *Extended viewport* cases for tiles selections in DVS1 and DVS2 solutions. The proposed prediction angles-based tiles selection solution (DVS1) performs tiles selection by considering two viewpoint/viewport prediction mechanisms without involving any prediction errors. Different prediction mechanisms can generate different prediction results. Using a combination of these prediction mechanisms can somehow increase the streaming performance. The *last known* prediction mechanism, which outputs the previously actual viewpoint as the future predicted viewpoint, is employed as the primary predicted mechanism. A *spherical walk* approach proposed in [8] is used as the secondary prediction mechanism 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. The prediction angles for primary and secondary prediction mechanisms for (i) th segment are represented as $\Phi^p(i)$, $\Theta^p(i)$ and $\Phi^s(i)$, $\Theta^s(i)$, respectively.

Spatial Information (SI) and Temporal Information (TI), defined by ITU-T P.910 [36], have been extensively used to approximate video scene complexity. SI is based on the Sobel filter, while TI is based upon the difference between successive frames. The research community employs these spatial and temporal complexity measurements to select content for subjective experiments, quality approximation as well as estimating compression levels and bandwidth requirements for a diverse range of streaming applications. More spatially and temporally rich 360° videos result in more drastic viewing patterns. Following this, the proposed solution combines viewing patterns and content complexity measurements to

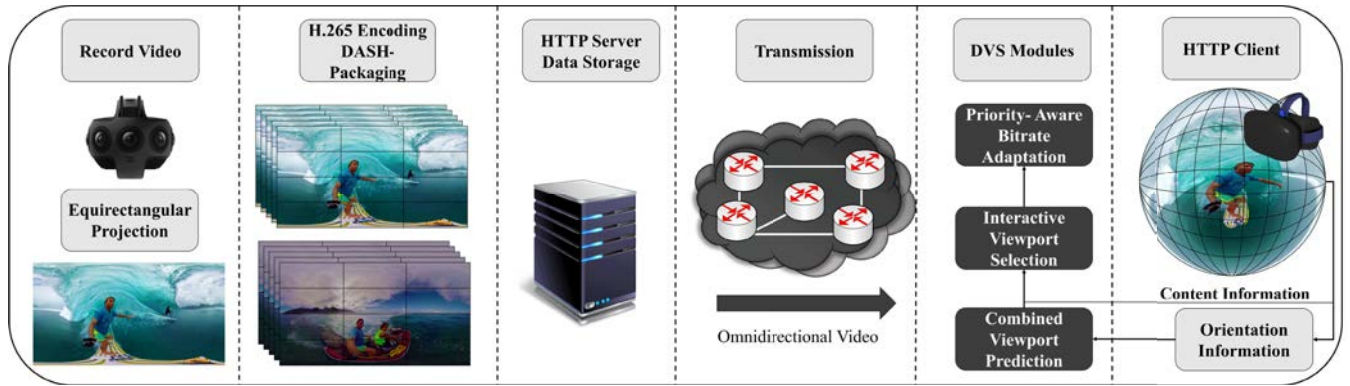


FIGURE 5. Overview of the user-centric DVS framework, enabling tile-based adaptive 360° video. The entire video is spatially and temporally divided into a number of tiles and segments at the server side. The user viewing patterns are leverage at the client side to select suitable tiles for each segment/adaptation interval.

dynamically select the viewing region for 360° video. DVS measures SI and TI values on a frame-by-frame basis for each 360° video. These values are then averaged across each segment. To establish the best spatial and temporal correlation, DVS employs the product of the averaged SI and TI values for each segment. Let $S(i)$ and $T(i)$ be the spatial and temporal scores, respectively, for the (i) th segment. The standard deviation reflects how much dispersion occurs over a while. The variation for the (i) th segment is given as:

$$ST(i) = std(S(i) * T(i) : S(i-1) * T(i-1)) \quad (1)$$

The content complexity for (i) th segment is approximated as follows:

$$CC(i) = \frac{ST(i)}{\max\{ST(i) | \forall i \in [1 : S]\}} \quad (2)$$

Algorithm 1 is based on the difference between two prediction mechanisms and content complexity variations to adapt better to the high dynamics head movements. Algorithm 1 starts by calculating the primary ($\mathcal{T}^p(i)$) and secondary viewport ($\mathcal{T}^s(i)$) tiles sets considering the predicted coordinates and viewport size (\mathcal{VS}). Each viewport contains the tiles whose centers are not located more than half of the viewport size from the center tile. The prediction difference between two prediction mechanisms is adjusted according to the content complexity variations computed in Eq. 2. Let $\mathcal{AD}(i)$ represent the adjusted distance of two prediction mechanisms with respect to the content information. We choose different thresholds to calculate the tiles for different streaming regions. Let Ω_1 and Ω_2 represents the thresholds of great circular distance between two predicted viewpoints, such as $\Omega_1 < \Omega_2$ and $\Omega_1, \Omega_2 \in [0, \mathcal{VS}]$. If the adjusted distance between two predicted viewpoints is less than Ω_1 for the (i) th segment. This refers to *Fixed viewport* scenario, where the viewport tiles set containing all the tiles of the anticipated primary viewport and the background tiles are selected for streaming to the client. The neighboring tiles set is empty and does not contain any tile. If the adjusted prediction difference is in between Ω_1 and Ω_2 , we refer to this case as the *Neighbor*

region scenario, where the goal is to include the neighboring tiles to deal with possible head movement prediction errors. The primary viewport tiles set is considered as the potential viewport set for the (i) th segment. The non-identical tiles of the secondary viewport are considered as the neighboring tiles. The rest of the tiles are added to the background tiles set. Suppose the primary and secondary mechanisms' predicted viewpoints are farther than Ω_2 . This represents the *Extended viewport* scenario, where the viewport size is extended by adding the actual arc distance between primary and secondary predicted viewpoints to the default viewport size. The primary viewport tiles set is recalculated with the updated viewport size. In this scenario, the primary viewport tiles set is labeled as viewport and streamed with the background tiles sets for the (i) th segment.

2) PREDICTION ANGLES AND ERRORS-BASED TILES SELECTION

360° videos require high-resolution for the achievement of better-perceived quality. The proposed tiles selection solution performs tiles selection based on prediction angles and errors as well as content complexity variations to bound the discrepancy between actual and predicted tiles sets. DVS2 selects the tiles for different streaming regions based on a single viewport prediction mechanism. Algorithm 1 mainly considers the adjusted prediction distance between two prediction mechanisms, while Algorithm 2 employs adjusted prediction difference between actual and predicted viewpoints for dynamic viewport selection. Algorithm 2 starts by finding the potential viewpoint angles for the next segments by averaging the predicted points of the primary (*last known*) and secondary (*spherical walk*) prediction mechanisms. Algorithm 2 adjusts the prediction difference measured during the previous segment according to the content complexity variations in order to best select the tiles for the next segment. Let Ω'_1 and Ω'_2 represents the thresholds of great circular distance between original and predicted viewpoints, such as $\Omega'_1 < \Omega'_2$ and $\Omega'_1, \Omega'_2 \in [0, \mathcal{VS}]$. Suppose the adjusted distance between predicted and

Algorithm 1: Viewport Selection Algorithm-DVS1**Input :**

$\mathcal{T}(i) \leftarrow$ Sorted tile set for the streaming session; $\Phi^p(i), \Theta^p(i), \Phi^s(i), \Theta^s(i) \leftarrow$ Predicted angles; $\mathcal{D}(i) \leftarrow$ Arc distance between two predicted viewpoints; $\mathcal{VS} \leftarrow$ Size of the viewport in radians; $\widehat{\mathcal{VS}} = 0 \leftarrow$ Updated size of the viewport initialized to 0; $\mathcal{CC}(i) \leftarrow$ Content complexity variations; $\mathcal{AD}(i) \leftarrow$ Adjusted distance; $\Omega_1, \Omega_2 \leftarrow$ Distance thresholds

Result :

$\mathcal{T}^v(i), \mathcal{T}^n(i), \mathcal{T}^b(i) \leftarrow$ Estimated viewport, neighbor, and background tiles sets for the (i)th segment

```

1 Calculate  $\mathcal{T}^p(i)$  and  $\mathcal{T}^s(i)$  with tiles having distance to viewpoints ( $(\Phi^p(i), \Theta^p(i))$ ) ( $\Phi^s(i), \Theta^s(i)$ ) in range  $[0:\mathcal{VS}/2]$ 
2  $\mathcal{D}(i) = \text{ArcDistance}(\Phi^p(i), \Theta^p(i), \Phi^s(i), \Theta^s(i))$ 
3  $\mathcal{AD}(i) = \mathcal{D}(i) + \mathcal{D}(i) * \mathcal{CC}(i)$ 
4 if  $\mathcal{AD}(i) \geq 0 \&\& \mathcal{AD}(i) \leq \Omega_1$  then
5   |  $\mathcal{T}^v(i) = \mathcal{T}^p(i)$ 
6   |  $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$ 
7 end
8 else if  $(\mathcal{AD}(i) > \Omega_1 \&\& \mathcal{AD}(i) \leq \Omega_2)$  then
9   |  $\mathcal{T}^v(i) = \mathcal{T}^p(i)$ 
10  |  $\mathcal{T}^n(i) = \mathcal{T}^s(i) - \mathcal{T}^p(i)$ 
11  |  $\mathcal{T}^b(i) = \mathcal{T}(i) - (\mathcal{T}^v(i) \cup \mathcal{T}^n(i))$ 
12 end
13 else
14  |  $\widehat{\mathcal{VS}} = \mathcal{VS} + \mathcal{D}(i)$ 
15  | Recalculate  $\mathcal{T}^p(i)$  with tiles having distance to viewpoint  $(\Phi^p(i), \Theta^p(i))$  in range  $[0:\widehat{\mathcal{VS}}/2]$ 
16  |  $\mathcal{T}^v(i) = \mathcal{T}^p(i)$ 
17  |  $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$ 
18 end

```

Algorithm 2: Viewport Selection Algorithm-DVS2**Input :**

$\mathcal{T}(i) \leftarrow$ Sorted tile set for the streaming session; $\Phi^p(i), \Theta^p(i), \Phi^s(i), \Theta^s(i) \leftarrow$ Predicted angles; $\Phi^a(i), \Theta^a(i), \Phi^v(i), \Theta^v(i) \leftarrow$ Actual and Predicted angles; $\mathcal{D}(i-1) \leftarrow$ Arc distance between actual and predicted viewpoints; $\mathcal{VS} \leftarrow$ Size of the viewport in radians; $\widehat{\mathcal{VS}} = 0 \leftarrow$ Updated size of the viewport initialized to 0; $\mathcal{CC}(i) \leftarrow$ Content complexity variations; $\mathcal{AD}(i) \leftarrow$ Adjusted distance; $\Omega'_1, \Omega'_2 \leftarrow$ Distance thresholds

Result :

$\mathcal{T}^v(i), \mathcal{T}^n(i), \mathcal{T}^b(i) \leftarrow$ Estimated viewport, neighbor, and background tiles sets for the (i)th segment

```

1  $\Phi^v(i), \Theta^v(i) = (\Phi^p(i) + \Phi^s(i))/2, (\Theta^p(i) + \Theta^s(i))/2$ 
2  $\mathcal{AD}(i) = \mathcal{D}(i-1) + \mathcal{D}(i-1) * \mathcal{CC}(i)$ 
3 if  $(\mathcal{AD}(i) \geq 0 \&\& \mathcal{AD}(i) \leq \Omega'_1)$  then
4   | Calculate  $\mathcal{T}^v(i)$  with tiles having distance to viewpoint  $(\Phi^v(i), \Theta^v(i))$  in range  $[0:\mathcal{VS}/2]$ 
5   |  $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$ 
6 end
7 else if  $(\mathcal{AD}(i) > \Omega'_1 \&\& \mathcal{AD}(i) \leq \Omega'_2)$  then
8   | Calculate  $\mathcal{T}^v(i)$  with tiles having distance to viewpoint  $(\Phi^v(i), \Theta^v(i))$  in range  $[0:\mathcal{VS}/2]$ 
9   | Calculate  $\mathcal{T}^n(i)$  with tiles having distance to viewpoint  $(\Phi^v(i), \Theta^v(i))$  in range  $(\mathcal{VS}/2:\mathcal{VS}/2 + \mathcal{D}(i-1))$ 
10  |  $\mathcal{T}^b(i) = \mathcal{T}(i) - (\mathcal{T}^v(i) \cup \mathcal{T}^n(i))$ 
11 end
12 else
13  |  $\widehat{\mathcal{VS}} = \mathcal{VS} + \mathcal{D}(i-1)$ 
14  | Calculate  $\mathcal{T}^v(i)$  with tiles having distance to viewpoint  $(\Phi^v(i), \Theta^v(i))$  in range  $[0:\widehat{\mathcal{VS}}/2]$ 
15  |  $\mathcal{T}^b(i) = \mathcal{T}(i) - \mathcal{T}^v(i)$ 
16 end
17  $\mathcal{D}(i) = \text{ArcDistance}(\Phi^a(i), \Theta^a(i), \Phi^v(i), \Theta^v(i))$ 

```


Algorithm 3: Prioritized Bitrate Adaptation Algorithm

Input :
 $\mathcal{L} \leftarrow$ Video bitrates set of 360° segments; $\mathcal{T}^v(i), \mathcal{T}^n(i), \mathcal{T}^b(i) \leftarrow$ Selected tiles sets for streaming; $|\mathcal{T}^v(i)|, |\mathcal{T}^n(i)| \leftarrow$ Cardinality of tiles sets; $\widehat{B}(i) \leftarrow$ Available bandwidth for the (i)th segment; $\widehat{VS} \leftarrow$ Updated size of the viewport;

Result :
 $w^{\mathcal{T}^v}(i), w^{\mathcal{T}^n}(i) \leftarrow$ Weights of tiles sets; $B^v(i), B^n(i), B^b(i) \leftarrow$ Bandwidth for the viewport, neighboring, and background tiles; $\mathcal{L}^{\mathcal{T}^v}(i), \mathcal{L}^{\mathcal{T}^n}(i), \mathcal{L}^{\mathcal{T}^b}(i) \leftarrow$ Video bitrates selected for different tiles sets

```

1  if ( $\widehat{B}(i) \leq \sum_{k \in \mathcal{T}(i)} \mathcal{L}_1^k(i)$ ) then
2    |  $\mathcal{L}^{\mathcal{T}}(i) = \mathcal{L}_1^k(i), \text{for all } k \in \mathcal{T}(i)$ 
3  end
4  else if ( $\widehat{B}(i) \geq \sum_{k \in \mathcal{T}(i)} \mathcal{L}_j^k(i)$ ) then
5    |  $\mathcal{L}^{\mathcal{T}}(i) = \mathcal{L}_j^k(i), \text{for all } k \in \mathcal{T}(i)$ 
6  end
7  else
8    |  $\mathcal{L}^{\mathcal{T}}(i) = \mathcal{L}_1^k(i), \text{for all } k \in \mathcal{T}(i)$ 
9    |  $B(i) = \widehat{B}(i) - \sum_{k \in \mathcal{T}(i)} \mathcal{L}_1^k(i)$ 
10   | if ( $\mathcal{T}^n(i) == \emptyset$ ) then
11     | if ( $\widehat{VS} \neq 0$ ) then
12       |  $\mathcal{L}^{\mathcal{T}}(i) = \max_j \{ \mathcal{L}_j^k(i) | \mathcal{L}_j^k(i) \leq B(i) - \sum_{k \in \mathcal{T}(i)} \sum_{j \in [2:J]} \mathcal{L}_j^k(i) \}$ 
13     | end
14     | else
15       |  $B^v(i) = B(i)$ 
16       |  $\mathcal{L}^{\mathcal{T}^v}(i) = \max_j \{ \mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq B^v(i) \}$ 
17       |  $B^b(i) = B(i) - (\sum_{k \in \mathcal{T}_p^v(i)} \mathcal{L}_{p,j}^k(i))$ 
18       |  $\mathcal{L}^{\mathcal{T}^b}(i) = \max_j \{ \mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^b(i)} \mathcal{L}_j^k(i) \leq B^b(i) \}$ 
19     | end
20   | end
21   | else
22     |  $w^{\mathcal{T}^n}(i) = (|\mathcal{T}^n(i)| / (2 * |\mathcal{T}^v(i)| + |\mathcal{T}^n(i)|))$ 
23     |  $w^{\mathcal{T}^v}(i) = 1 - w^{\mathcal{T}^n}(i)$ 
24     |  $B^v(i) = B(i) * w^{\mathcal{T}^v}(i)$ 
25     |  $B^n(i) = B(i) * w^{\mathcal{T}^n}(i)$ 
26     |  $\mathcal{L}^{\mathcal{T}^v}(i) = \max_j \{ \mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^v(i)} \mathcal{L}_j^k(i) \leq B^v(i) \}$ 
27     |  $\mathcal{L}^{\mathcal{T}^n}(i) = \max_j \{ \mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^n(i)} \mathcal{L}_j^k(i) \leq B^n(i) \}$ 
28     |  $B^b(i) = B(i) - (\sum_{k \in \{\mathcal{T}^v(i), \mathcal{T}^n(i)\}} \mathcal{L}_j^k(i))$ 
29     |  $\mathcal{L}^{\mathcal{T}^b}(i) = \max_j \{ \mathcal{L}_j^k(i) | \sum_{k \in \mathcal{T}^b(i)} \mathcal{L}_j^k(i) \leq B^b(i) \}$ 
30   | end
31 end

```

actual viewpoints during (i - 1)th segment is not greater than the Ω'_1 . This refers to *Fixed viewport* scenario, where only the viewport and background tiles sets are calculated by considering the predicted angles and the actual viewport size. For the *Neighbor region* scenario, the viewport and neighboring tiles are added with tiles having distance to viewpoint centre in range of $[0 : VS/2]$ and $(VS/2 : VS/2 + D(i - 1))$, respectively. The prediction performance could be worse under quick head turns for different motion scenes, and the adjusted distance could be even larger than Ω'_2 . In this

case, an *Extended viewport* streaming approach is adopted by increasing the viewport size with added background tiles to lower the impact of viewport mismatch.

C. PRIORITIZED BITRATE ADAPTATION

In an adaptive streaming environment, the client mainly focuses on using the network resources efficiently to promote the desirable end-user streaming experience [37], [38]. Algorithm 3 determines the appropriate bitrate selection for the tiles belonging to different priority regions.

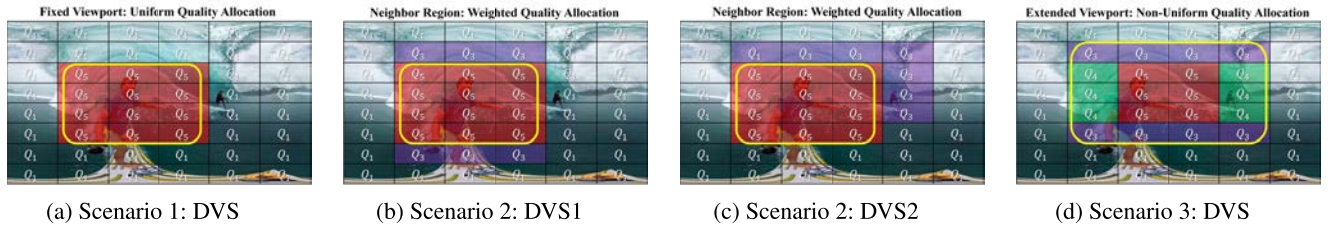


FIGURE 6. Viewport selection and bitrate adaptation scenarios in DVS solutions.

After completely downloading the previous segment, the playback adaptation is performed for each segment. Algorithm 3 requests the whole segment in the lowest quality if the available bandwidth is insufficient. This can prevent viewport deviation but can lead to significant playback interruptions. If the available bandwidth is higher than the largest available quality level, the highest representation for the entire 360° segment is requested to ensure significant visual enhancements.

In all other cases, first, the minimum quality level is assigned to all the tiles, and next, the available bandwidth budget is updated. Supposing there are no neighboring tiles, then for the *Extended viewport* scenario of both Algorithm 1 and Algorithm 2, Algorithm 3 performs a *Non-Uniform* or per-tile quality adaptation by increasing the quality from the viewpoint to the last tile in the frame due to the extended viewport size. A *Uniform* or per-region quality selection is performed in the *Fixed viewport* scenarios of both DVS1 and DVS2 methods. Algorithm 3 selects the same highest possible quality level for viewport tiles to ensure visual quality improvements. Following this step, the background bandwidth budget is updated, and the quality of the background tiles is increased uniformly.

If the neighboring region is non-empty, a *Weighted* quality selection is carried out for viewport and neighboring tiles. The weights association depends on the distribution of the tiles in these regions. Since it is expected that a VR user wants to watch the viewport content at a higher quality than the neighboring tiles, higher priority weights are allocated to the viewport tiles. The viewport and neighboring tiles' bandwidth are computed based on priority-related weights. Next, the quality allocation for the viewport and the neighboring tiles is performed, such as the maximum bitrates possible while not exceeding the bandwidth limit for each region are selected for streaming. The bandwidth for the background tiles is calculated by subtracting the allocated budget to the viewport and neighbor tiles from the updated overall bandwidth budget. Finally, the background tiles' quality is increased as well, respecting the available bandwidth budget. In conclusion, the proposed bitrate adaptation implements a *Uniform*, *Weighted*, and *Non-Uniform* bitrate adjustments for DVS1 and DVS2 methods. Noteworthy is that for different streaming scenarios of Algorithm 1 and Algorithm 2, Algorithm 3 improves the corresponding bitrate choice for each tile that the network can support. Fig. 6 presents the pictorial illustrations of different

TABLE 2. Characteristics of the content employed in the experiments.

Videos	Category	Duration	Resolution	FPS
Conan	Performance	2'44''	3840x2160	29
Spotlight	Film	4'53''	3840x2160	30
Surfing	Sport	3'25''	3840x1920	29

TABLE 3. Average data sizes [Mbps] per-segment for the Conan, Spotlight, and Surfing videos.

Video	QPs	1s			2s		
		4x3	6x4	8x6	4x3	6x4	8x6
Conan	22	10.60	10.68	10.88	21.22	21.37	21.77
	27	5.05	5.12	5.30	10.10	10.25	10.63
	32	2.43	2.51	2.68	4.87	5.02	5.39
	37	1.24	1.32	1.50	2.49	2.66	3.03
Spotlight	42	0.72	0.80	0.98	1.44	1.62	1.99
	22	13.63	13.93	14.31	27.18	27.77	28.55
	27	7.21	7.44	7.77	14.37	14.84	15.50
	32	4.06	4.24	4.50	8.11	8.47	9.01
Surfing	37	2.36	2.49	2.72	4.70	4.98	5.44
	42	1.35	1.46	1.66	2.69	2.93	3.35
	22	22.76	23.07	23.58	45.30	45.93	46.96
	27	12.84	13.07	13.48	25.55	26.03	26.86
	32	7.23	7.41	7.73	14.40	14.76	15.42
	37	4.00	4.15	4.42	7.98	8.27	8.82
	42	2.13	2.26	2.49	4.25	4.51	4.99

TABLE 4. Experimental settings.

Parameter	Configuration
Tiling patterns	4x3, 6x4, 8x6
Encoding versions (QPs)	22, 27, 32, 37, 42
\mathcal{VS}	110°
Ω_1, Ω'_1	$\mathcal{VS}/3$
Ω_2, Ω'_2	$\mathcal{VS}/2$
Segment length (s)	1s, 2s
Buffer levels	2s, 4s
Simulation length	Video duration

viewport selection and bitrate adaptation scenarios for DVS1 and DVS2 methods.

IV. PERFORMANCE EVALUATION

This section assesses the streaming performance of the DVS and popular tile-based adaptive streaming solutions using trace-driven experiments.

A. EXPERIMENTAL SETTINGS

We consider a headless Python-based VR player [8] involving recorded head movement patterns to perform the experiments

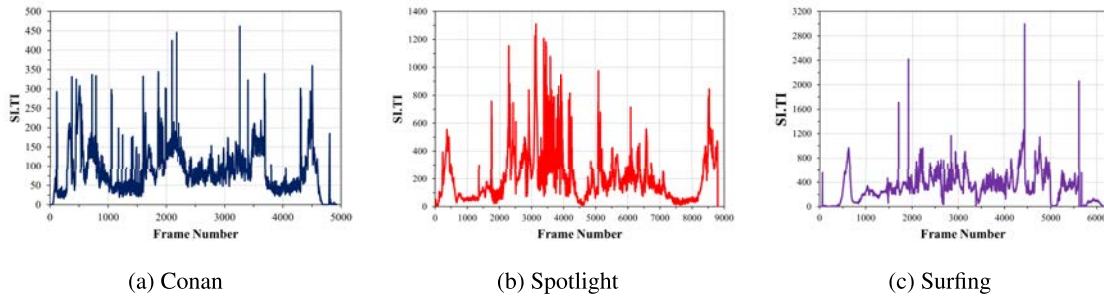


FIGURE 7. The per-frame SI.TI values for Conan, Surfing, and Spotlight videos.

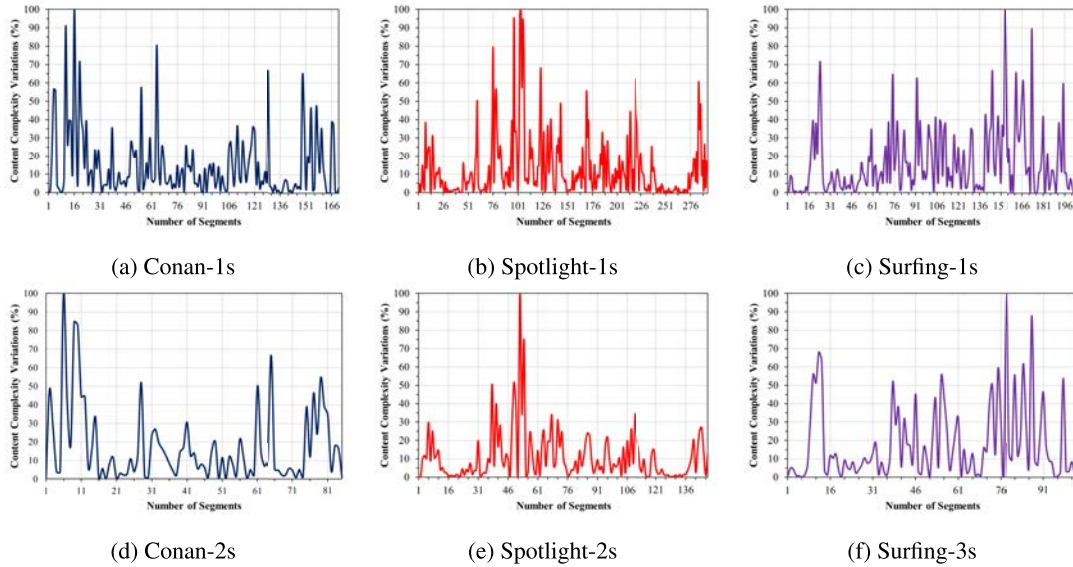


FIGURE 8. Content complexity variations for Conan, Spotlight, and Surfing Videos with 1s and 2s.

on Ubuntu 16.04 machine with a 64-bit Intel Core i7-6600U CPU 2.60 GHz and 16 GB memory. A Python-based HTTP server transmits the requested tiles to the VR player. The VR client is equipped with a playout buffer; however, the proposed bitrate adaptation solution performs adaptation decisions based on throughput and viewing measurements. The head movement patterns from 48 users are employed from a VR dataset [32]. The dataset contains the viewing traces for 18 VR videos recorded in two experiments using an HTC VIVE headset with a viewport size of 110° .

Three different motion 360° videos with at least 4K resolution were sourced from YouTube: (1) **Conan360°-Sandwich**²; (2) **Google Spotlight-HELP**³; (3) **GoPro VR-Tahiti Surf**.⁴ Throughout this paper, these videos are referred to as Conan, Spotlight, and Surfing. These videos belong to three different categories: Performance (Conan), Action Film (Spotlight), and Sport (Surfing). Fig. 7 illustrates the SI.TI values for each video, computed using code from [39]. The

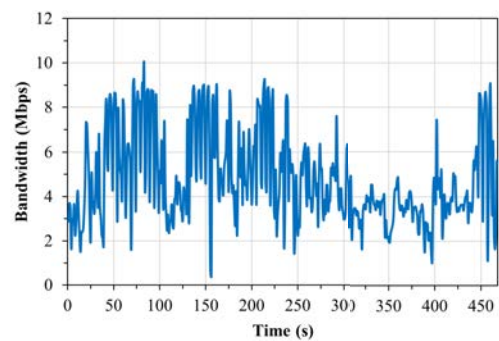


FIGURE 9. 4G bandwidth logs recorded in Bus [41].

content complexity variations computed for each video using Eq. 1 and Eq. 2 are shown in Fig. 8. The content-related features, i.e., category, duration, resolution, and frames per second (FPS), are described in Table 2. The employed video clips are prepared into 4×3 (12), 6×4 (24), and 8×6 (48) tiling patterns using Kvazaar encoder [40]. The tiled clips are encoded with five different quantization parameter (QP) values (i.e., 22, 27, 32, 37, 42). Afterward, the 1s and 2s DASH

²<https://www.youtube.com/watch?v=FiCIYLgxJ5s>

³<https://www.youtube.com/watch?v=G-XZhKqQAHU>

⁴<https://www.youtube.com/watch?v=MKWWhf8RAV8>

video segments are produced using GPAC MP4Box.⁵ Table 3 presents the obtained data sizes in Mbps for three different video contents prepared in 1s and 2s segment durations.

In practice, for 360° video streaming, the buffer should be as small as possible to accommodate the new video chunks in response to the user movements within the immersive video. The buffer level is set to the two video segments for each experiment. The length of each experiment was adjusted following the duration of the employed video. A bandwidth trace from the 4G bandwidth dataset provided by [41] was employed. The bandwidth logs recorded in the Bus are used for the evaluations. The normalized trace to verify the streaming performance under restricted connection speeds is given in Fig. 9. Simulations were performed using different values of Ω_1 , Ω_2 , Ω'_1 , and Ω'_2 . The reported experimental results are obtained by setting Ω_1 , $\Omega'_1 = \mathcal{VS}/3$ and Ω_2 , $\Omega'_2 = \mathcal{VS}/2$. Table 4 exhibits the experimental settings in terms of tiling pattern, encoding versions, segment length, viewport size, and buffer levels.

The proposed solutions were compared with four tile-based adaptive schemes discussed in the related works: (1) **UVP [8]**, a viewport prediction-based scheme, which uniformly distributes the bandwidth budget for tiles belonging to the viewport and background regions based on the spherical walk prediction method; (2) **CTF [8]**, an extended version of the UVP scheme, where the tiles are sorted based on the distance from the viewpoint and the whole frame is transmitted in per-tile quality increment fashion; (3) **HOS [10]**, a viewing probability-based scheme, which considers different quality versions for the center and surrounding tiles; (4) **PET [18]**, a scheme which considers equal neighboring tiles around the viewport in all directions and assigns different priorities to the selected regions.

The following metrics are considered for performance evaluation: (1) **Average Viewport Overlap**: indicating the overlap between actual and predicted viewport tiles; (2) **Average Perceived Quality**: the quality of the visible tiles based on the ground truth head movement traces averaged across all segments and VR users; (3) **Average Quality Fluctuations**: the weighted summation of spatial and temporal quality fluctuations within the viewport region; and (4) **Viewport Bandwidth Utilization**: measured as the ratio of the video bitrates of the viewport and the bitrates of the whole 360° segment.

B. EXPERIMENTAL RESULTS

This subsection presents each solution's experimental results and analysis under different settings.

1) STREAMING TILES SELECTION

Table 5 presents the streaming behavior of DVS1 and DVS2 clients during 360° video streaming. It includes the percentage results of tiles selection cases for each video in these solutions. DVS1 mostly selects *Fixed Viewport*, while the DVS2

TABLE 5. Streaming regions selection [%] by DVS1 and DVS2 clients for Conan (Co), Spotlight (Sp), and Surfing (Su) videos prepared in 4 × 3, 6 × 4, and 8 × 6 tiling patterns and 1s and 2s segment duration. The results are averaged across the 48 VR users.

Video	Tiles	DVS1			DVS2		
		FVP	NR	EVP	FVP	NR	EVP
Co-1s	8x6	92.48	3.90	3.63	83.36	7.21	9.44
	6x4	91.47	4.57	3.96	82.49	7.81	9.71
	4x3	90.58	5.02	4.40	80.88	7.92	11.20
Co-2s	8x6	79.83	6.99	13.19	71.67	8.60	19.73
	6x4	78.60	7.16	14.24	70.96	9.22	19.83
	4x3	77.06	7.57	15.37	71.13	8.87	20.00
Sp-1s	8x6	91.71	4.11	4.18	80.87	7.50	11.63
	6x4	91.14	4.43	4.43	80.47	7.36	12.17
	4x3	90.76	4.69	4.55	80.08	7.55	12.37
Sp-2s	8x6	80.02	7.41	12.57	68.11	10.10	21.78
	6x4	79.12	7.81	13.07	67.06	10.40	22.53
	4x3	78.54	8.04	13.42	66.87	10.63	22.51
Su-1s	8x6	89.16	6.04	4.81	73.39	10.98	15.63
	6x4	88.13	6.53	5.34	72.46	11.20	16.34
	4x3	87.41	7.00	5.59	71.48	11.11	17.41
Su-2s	8x6	71.04	11.04	17.92	56.96	11.87	31.17
	6x4	71.10	11.02	17.88	56.07	12.10	31.84
	4x3	67.21	12.66	20.13	55.50	11.69	32.81

employs *Neighbor Region* and *Extended Viewport* selection more than DVS1. DVS1 employs *Fixed Viewport* for about 83.07% of the streaming session compared to 7% of *Neighbor Region*. DVS2 observes the highest average percentage of 18.73% for the *Extended Viewport* scenario compared to 9.92% of DVS1.

It is interesting to note from Table 5 that with an increase in segment duration the percentage of *Fixed viewport* selection drops and an increase in *Neighbor Region* or *Extended Viewport* is noted. For instance, increasing the segment duration from 1s to 2s for DVS1 client results in 13.01%, 11.97%, and 28.45% decrease in *Fixed Viewport* scenario for Conan, Spotlight, and Surfing videos, respectively. Similarly, for the DVS2 client, an increase of 9.73%, 10.21%, and 15.48% in *Extended Viewport* scenario is observed for three videos with the increase in segment duration. This is because with the increase in segment duration the viewport overlap drops which results in switching from *Fixed Viewport* to *Neighbor Region* or *Extended Viewport* scenarios. The next subsections describe how employing different tiles selection scenarios in proposed solutions result in improved streaming performance measured in terms of tiles overlap, QoE, and bandwidth utilization in comparison to other algorithms.

2) AVERAGE VIEWPORT OVERLAP

We measure the average viewport overlap values (per video, across 48 viewport traces) for DVS1, DVS2, and *spherical walk* streaming methods. DVS1 considers a combination of two prediction mechanisms, i.e., *last known* and *spherical walk* while DVS2 employs the average of these two prediction mechanisms. The *spherical walk* method is used for UVP, CTF, HOS, and PET methods; therefore, the viewport overlap results for UVP are mentioned here. Fig. 10 shows that users have relatively the same head movements for all three videos. Notably, DVS1 and DVS2 achieve significantly higher viewport overlap for all three videos. For all 48 VR users, DVS1

⁵<https://gpac.wp.imt.fr/mp4box/>

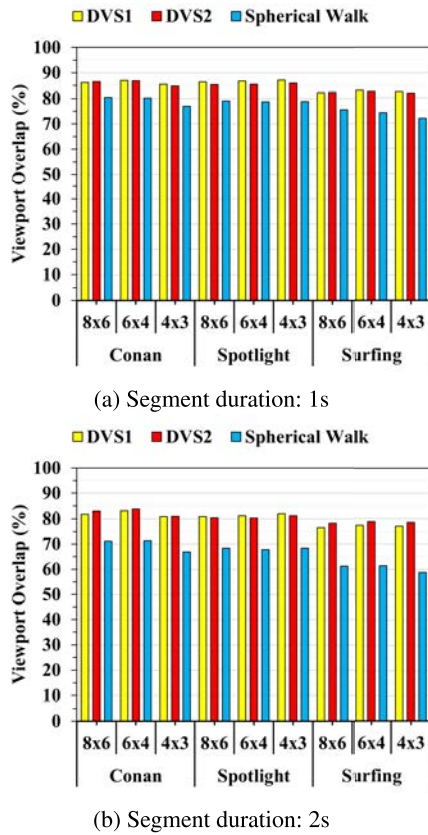


FIGURE 10. Average viewport overlap achieved by DVS1, DVS2, and UVP streaming methods for Conan, Spotlight, and Surfing videos.

experiences an average tiles overlap of 86.23% and 81.94% (Conan), 86.74% and 81.42% (Spotlight), and 82.62% and 76.98% (Surfing) for 1s (Fig. 10a) and 2s (Fig. 10b) segment lengths, respectively. The reason is that the region-based tiling pattern selection mechanism adapts better to the diverse spatial and temporal information of different motion scenes. As seen, the average gain achieved by DVS1 and DVS2 over *spherical walk* method for all videos are 7.88% and 7.35% for next 1s (Fig. 10a), and 13.97% and 14.45% for 2s prediction horizons (Fig. 10b). At the same time, the *spherical walk* method for UVP observes the lowest average viewport overlap than the proposed methods. The viewport overlap for the DVS1 is reduced by 4.28% (Conan), by 5.31% (Spotlight), and by 5.64% (Surfing) when segment duration is increased from 1s to 2s, while for the DVS2 and UVP methods, it is reduced by 3.53% and 9.41% (Conan), by 4.89% and 10.65% (Spotlight), and by 3.73% and 13.42% (Surfing), respectively. Consequently, it can be noted that the advanced viewport selection in the proposed solutions leads to better overlap between actual and predicted viewport than the traditional streaming method.

3) PERCEIVED QUALITY LEVELS

The average perceived quality results are shown in Fig. 11 for the Conan, Spotlight, and Surfing videos, prepared in

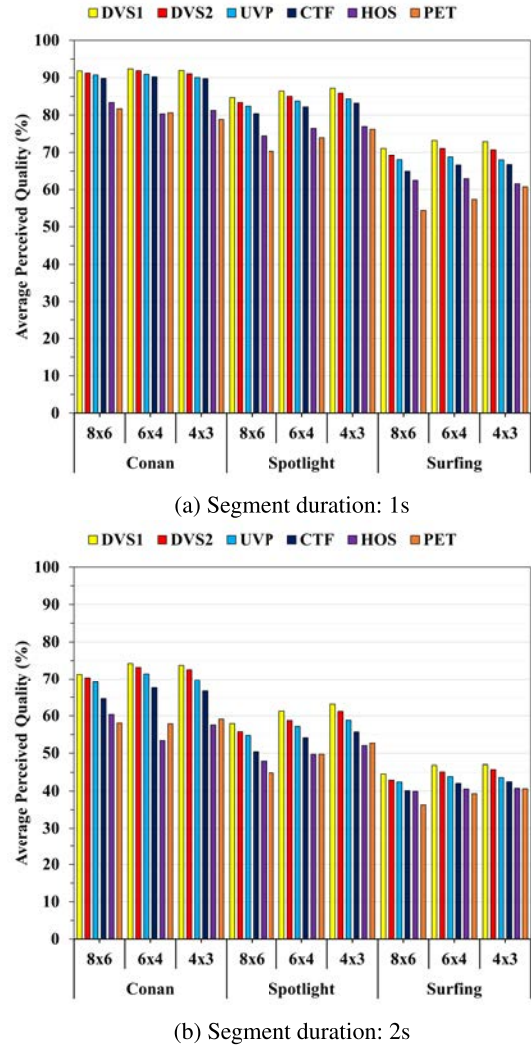


FIGURE 11. Average perceived quality achieved by DVS1, DVS2, UVP, CTF, HOS, and PET streaming methods for Conan, Spotlight, and Surfing videos.

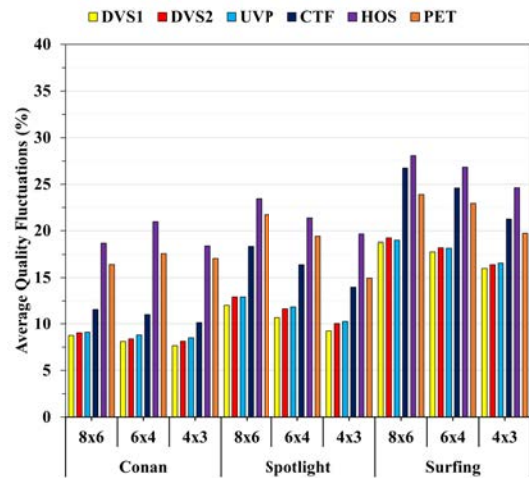
three different tiling patterns. The Conan video's most petite, average segment sizes result in overall higher quality values. The drastic viewing directions due to the multiple moving objects and higher data size for the Surfing video result in relatively lower quality results even under 1s segment duration. As can be seen, from Fig. 11a, the actual viewport quality levels of the DVS methods are more significant than the others because the interactive tiles selection cases effectively support the visual quality enhancements. Mainly, DVS1 and DVS2 improve the visual quality by up to 2.98% and 1.5% compared to the UVP, up to 5.23% and 3.75% compared to the CTF, up to 10.53% and 9.05% compared to the HOS, and up to 12.17% and 10.69% compared to the PET methods for the entire test datasets. This is because, instead of entirely relying on the positional information for different motion content, the joint content complexity and prediction results can improve the visual quality levels along with a lower discrepancy between actual and predicted viewports (as reported in Fig. 10). In Fig. 11a, the DVS methods

attain the most noticeable results for Surfing video with all three tiling patterns, where DVS1 and DVS2 achieve average quality improvements of about 2.02%-4.10% and 14.88%-12.80% compared to the UVP and PET methods, respectively. For the Spotlight video with a 6×4 tiling pattern, DVS1 can improve the average quality by up to 4.21% and by up to 12.51% compared to the CTF and HOS methods, respectively.

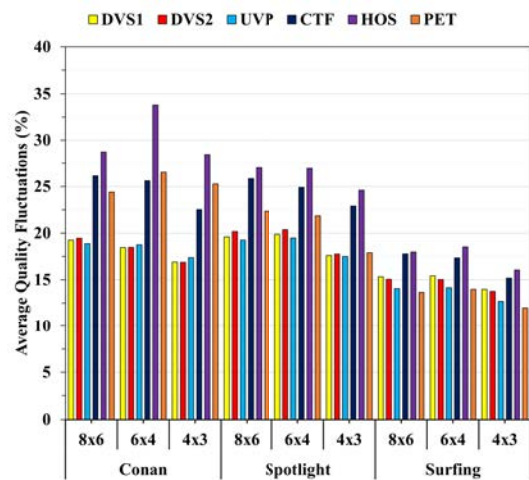
The average quality values of the DVS and reference methods for 2s segment duration are shown in Fig. 11b. Similar to 1s segment length results, DVS solutions perform better than alternative approaches in terms of achieving high viewport quality for 2s segment duration. Notably, DVS1 and DVS2 improve the average visual quality levels by 7.91% and 6.31%, respectively, compared to all other methods during the entire streaming sessions. DVS1 and DVS2 achieve about 74.24% and 73.18% average quality levels compared to 71.43%, 67.54%, 53.37%, and 57.83% of comparative algorithms, respectively, for the Conan video with a 6×4 tiling setting. For the Spotlight video, DVS1 and DVS2 methods achieve an average quality improvement of about 3.86% and 1.64% compared to UVP, 7.4% and 5.18% compared to CTF, 10.89% and 8.67% compared to HOS, and 11.67% and 9.45% compared to PET methods. Similarly, the DVS1 method achieves an average quality value of 46.13%, compared to CTF and PET algorithms, which nearly touch 41.54% and 38.72%, respectively, for the Surfing video. UVP receives higher performance than the other comparative approaches because it uniformly increases the quality levels. The HOS and CTF algorithms tend to focus more on the center tile leading to lower quality levels under inaccurate prediction results. The degraded performance of the PET method even under stable head movements is due to the fact that it unnecessarily increases the quality of the adjacent tiles. Comparatively, our proposed solutions implements *Uniform*, *Weighted*, and *Non-Uniform* quality allocations to support higher perceived visual quality levels.

4) QUALITY FLUCTUATIONS

In 360° video, the users can only view the tiles within the viewport; therefore, we measure the quality fluctuations as the weighted summation of spatial (quality variations within the viewport) and temporal (average perceived quality difference between two consecutive viewports) by setting the weights to 0.5 for each variation. Fig. 12 summarizes the performance of each scheme in terms of average quality oscillations for three different motion videos under dynamic network conditions. As seen, in terms of average quality oscillations, DVS1 experiences 8.16%, 10.65%, and 17.52% average quality fluctuations compared to 10.90%, 16.25%, and 26.51% of CTF and 10.9%, 21.49%, and 26.51% of HOS methods, for Conan, Spotlight, and Surfing videos, respectively (Fig. 12a). This is because CTF and HOS implement *Non-Uniform* quality allocations for each adaptation interval. A similar phenomenon can also be observed when the segment duration is increased from 1s to 2s (Fig. 12b).



(a) Segment duration: 1s



(b) Segment duration: 2s

FIGURE 12. Average temporal quality oscillations achieved by DVS1, DVS2, UVP, CTF, HOS, and PET streaming methods for Conan, Spotlight, and Surfing videos.

The HOS method undergoes 33.81% quality fluctuations for 6×4 Conan video, while the proposed methods experience up to 45.39% fewer quality fluctuations. It is interesting to note that compared to Fig. 12a, Fig. 12b results in higher quality fluctuations for Conan and Spotlight videos because the increased segment duration results in the lower prediction performance, which leads to more quality versions inside the viewport. However, this difference is decreased for Surfing video due to the larger average segment sizes. DVS1 and DVS2 methods observe slightly higher quality fluctuations than UVP and PET methods for Spotlight video; this is because UVP and PET algorithms mostly stick to the lower quality levels, and not many quality fluctuations are observed. However, our proposed methods observe the highest average perceived quality levels and acceptable quality fluctuations with this video. On the other hand, UVP allocates bitrate for tiles belonging to the same classification to reduce the computational complexity based on the estimated bandwidth.

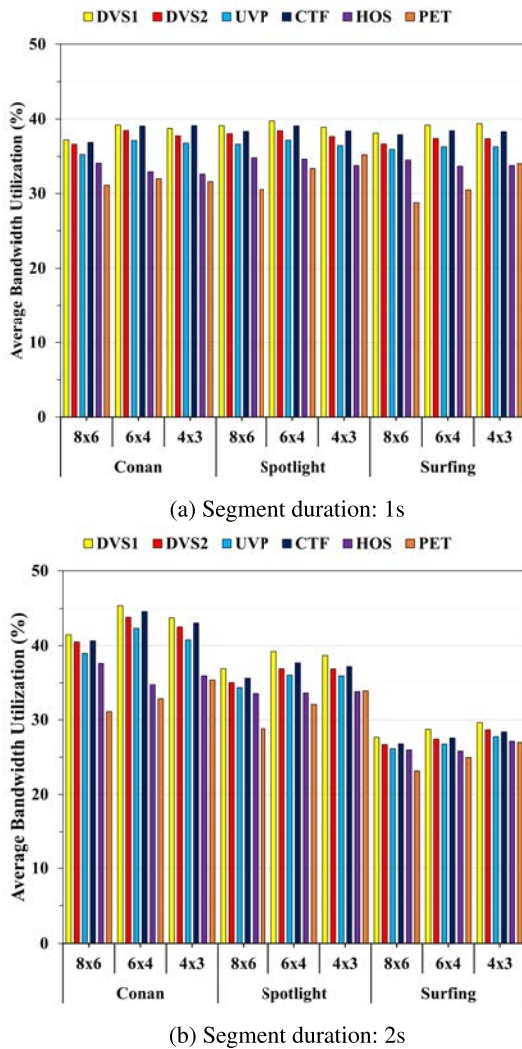


FIGURE 13. Average bandwidth utilization achieved by DVS1, DVS2, UVP, CTF, HOS, and PET streaming methods for Conan, Spotlight, and Surfing videos.

However, it is highly limited by the viewport prediction errors observed during segments playback. As a result, we can conclude that DVS better assesses the network conditions and time-varying positional information to boost the playback smoothness objectives.

5) BANDWIDTH UTILIZATION

The results for bandwidth utilization for the viewport region are discussed in this section. The bandwidth utilization for the (i) th segment is measured as the ratio of the video bitrates of the viewport and the bitrates of the whole 360° segment. Fig. 13 shows the average bandwidth utilization for the viewport region. The depicted results are achieved by six streaming clients for three videos watched using 48 VR users in two different segment lengths. Fig. 13a shows the average bandwidth utilization results for 1s segment duration. It can be noted that the DVS1 solution significantly achieves higher viewport bandwidth utilization in comparison to other

streaming solutions. At the same time, the lowest average bandwidth utilization is observed for the PET streaming client. Interestingly, PET solution occupies about 31.89% bandwidth for the viewport region and 68.11% for the non-visible region. This is unacceptable for 360° video streaming. Among proposed solutions, DVS1 achieves on average 6.21%, 1.15%, 12.81%, and 17.87% higher bandwidth utilization for all three videos compared to UVP, CTF, HOS, and PET streaming solutions. Notably, the CTF solution has slightly higher bandwidth utilization compared to DVS2. This is because it allocates higher bitrates to the viewport tiles compared to the background tiles. Fig. 13b shows the average bandwidth utilization when the segment duration is set to 2s. For Conan video, DVS1 solution exceeds in performance compared to UVP by 6.46%, CTF by 1.72%, HOS by 17.02%, and PET by 23.94%. DVS1 solution achieves up to 7.44% (UVP), 3.77% (CTF), 12.14% (HOS), and 17.48% (PET) higher bandwidth utilization for Spotlight video with a 2s segment duration (Fig. 13b). This is because the proposed framework streams non-viewport tiles in a lower quality to improve the bandwidth utilization for viewport streaming. While other methods unnecessarily increase the bitrate for the non-viewing region due to lower prediction performance.

C. DISCUSSION

The key observations and findings following the design of the novel solutions and simulation-based experiments performed can be summarized as follows:

- It is difficult to achieve higher perceived quality levels for the content with higher data rates and uncertain viewing motions (i.e., Surfing) in comparison to relatively stable motion content (i.e., Conan) with lower segment sizes under the same bandwidth conditions.
- With the increase in the segment duration, the average perceived quality values decrease since the user head movements become more challenging to predict over longer prediction horizons.
- The PET method performs worse than all the tiles-based solutions. The average perceived quality levels and the bandwidth utilization for the viewport tiles are observed lowest with this solution.
- Employing neighbor region tiles in all directions as in PET can lead to inefficient bandwidth utilization and thus lower perceived quality levels. Instead, dynamically defining the neighbor region assisted by content information and viewing differences significantly improves streaming performance for content with different motion characteristics.
- DVS1 solution achieves slightly higher visual quality levels and bandwidth utilization than DVS2. However, DVS2 achieves the highest viewport overlap compared to DVS1. The reason is that the DVS2 solution performs a higher Extended viewport tiles selection scenario under defined thresholds than DVS1.

- Dynamically performing *Uniform* and *Non-Uniform* quality allocations along with prediction angles-based (DVS1) and prediction angles and prediction differences-based (DVS2) tiles selection scenarios can facilitate achieving improved quality scores compared to the methods focusing on constantly increasing the quality (UVP and PET) and methods improving the quality starting from the center tile (CTF and HOS). The quality improvements were observed for all tiling patterns and across all segment durations.
- The weighted quality increment in the neighbor region scenario provides a cushion and absorbs the quick quality transitions from viewport to background image regions, thereby lowering spatial oscillations under both medium or medium-to-high prediction performance due to the equal-quality differentiated streaming regions.

In summary, incorporating content information and viewing preferences for tiles selection in proposed solutions assisted by the priority-oriented bitrate adaptation method significantly improves the performance during 360° video delivery; they support streaming high-quality visible tiles along with higher bandwidth utilization while maintaining visual smoothness.

V. CONCLUSION

This work focuses on enabling dynamic viewport selection based on content information and different watching behaviors of the users. It proposes two dynamic viewport selection solutions (DVS1 and DVS2) for 360° omnidirectional videos by employing content and positional information to achieve a balance between visual quality and viewport smoothness. The proposed approach extracts the content complexity variations and adjusts the difference between estimated prediction angles and actual and estimated angles to transmit high-quality tiled videos. When tested for different motion clips, DVS1, and DVS2 significantly improve the perceived quality levels. Experimental evaluations using a publicly available dataset show that DVS1 and DVS2 achieve substantially higher streaming performance than when employing other state-of-the-art methods. Improved streaming performance has been observed in terms of perceived quality levels, quality variations, and bandwidth utilization within the viewport averaged across all the VR users.

REFERENCES

- [1] D. Li, R. Du, A. Babu, C. D. Brumar, and A. Varshney, "A log-rectilinear transformation for foveated 360-degree video streaming," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 5, pp. 2638–2647, May 2021.
- [2] T. Rhee, L. Petikam, B. Allen, and A. Chalmers, "MR360: Mixed reality rendering for 360° panoramic videos," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 4, pp. 1379–1388, Apr. 2017.
- [3] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360° video streaming: Solutions, challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2801–2838, 4th Quart., 2020.
- [4] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu, "A survey on 360° video streaming: Acquisition, transmission, and display," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Jul. 2020, doi: [10.1145/3329119](https://doi.org/10.1145/3329119).
- [5] Z. Jiang, X. Zhang, Y. Xu, Z. Ma, J. Sun, and Y. Zhang, "Reinforcement learning based rate adaptation for 360-degree video streaming," *IEEE Trans. Broadcast.*, vol. 67, no. 2, pp. 409–423, Jun. 2021.
- [6] X. Chen, D. Wu, and I. Ahmad, "Optimized viewport-adaptive 360-degree video streaming," *CAAI Trans. Intell. Technol.*, vol. 6, no. 3, pp. 347–359, 2021.
- [7] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360° video streaming using tiles for virtual reality," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2174–2178.
- [8] J. V. D. Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. D. Turck, "Tile-based adaptive streaming for virtual reality video," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 4, pp. 1–24, Nov. 2019.
- [9] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 315–323, doi: [10.1145/3123266.3123291](https://doi.org/10.1145/3123266.3123291).
- [10] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 107–110.
- [11] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1977–1986.
- [12] J. Yu and Y. Liu, "Field-of-view prediction in 360-degree videos with attention-based neural encoder-decoder networks," in *Proc. 11th ACM Workshop Immersive Mixed Virtual Environ. Syst. (MMVE)*, 2019, pp. 37–42.
- [13] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very long term field of view prediction for 360-degree video streaming," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Mar. 2019, pp. 297–302.
- [14] 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 *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1190–1198.
- [15] 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 J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 29–42, Feb. 2019.
- [16] Y. Hu, Y. Liu, and Y. Wang, "VAS360: QoE-driven viewport adaptive streaming for 360 video," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2019, pp. 324–329.
- [17] N. Kan, J. Zou, C. Li, W. Dai, and H. Xiong, "RAPT360: Reinforcement learning-based rate adaptation for 360-degree video streaming with adaptive prediction and tiling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1607–1623, Mar. 2022.
- [18] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An HTTP/2-based adaptive streaming framework for 360° virtual reality videos," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 306–314.
- [19] K. Spiteri, R. Ugaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.
- [20] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados, and P. Chatzimisios, "FDASH: A fuzzy-based MPEG/DASH adaptation algorithm," *IEEE Syst. J.*, vol. 10, no. 2, pp. 859–868, Jun. 2016.
- [21] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 197–210.
- [22] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326–340, Feb. 2014.
- [23] 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 Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, 2015.
- [24] S. Qasim, D. Jasim, and Y. Li, "Proposed an adaptive bitrate algorithm based on measuring bandwidth and video buffer occupancy for providing smoothly video streaming," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 2, pp. 191–195, 2018.
- [25] X. Hu, W. Quan, T. Guo, Y. Liu, and L. Zhang, "Mobile edge assisted live streaming system for omnidirectional video," *Mobile Inf. Syst.*, vol. 2019, pp. 1–15, May 2019.

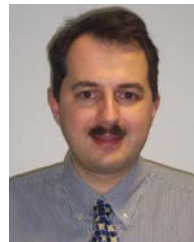
- [26] A. Yaqoob and G.-M. Muntean, "A combined field-of-view prediction-assisted viewport adaptive delivery scheme for 360° videos," *IEEE Trans. Broadcast.*, vol. 67, no. 3, pp. 746–760, Sep. 2021.
- [27] D. V. Nguyen, H. Van Trung, H. L. Dieu Huong, T. T. Huong, N. P. Ngoc, and T. C. Thang, "Scalable 360 video streaming using HTTP/2," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2019, pp. 1–6.
- [28] S. Park, A. Bhattacharya, Z. Yang, M. Dasari, S. R. Das, and D. Samaras, "Advancing user quality of experience in 360-degree video streaming," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, May 2019, pp. 1–9.
- [29] N. Kan, J. Zou, K. Tang, C. Li, N. Liu, and H. Xiong, "Deep reinforcement learning-based rate adaptation for adaptive 360-degree video streaming," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 4030–4034.
- [30] D. He, C. Westphal, and J. Garcia-Luna-Aceves, "Joint rate and FoV adaptation in immersive video streaming," in *Proc. ACM SIGCOMM Workshop AR/VR Netw.*, 2018, pp. 27–32.
- [31] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. 5th Workshop All Things Cellular, Oper., Appl. Challenges*, Oct. 2016, pp. 1–6.
- [32] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *Proc. 8th ACM Multimedia Syst. Conf.*, Jun. 2017, pp. 193–198.
- [33] D. Salomon, *Transformations and Projections in Computer Graphics*. Berlin, Germany: Springer-Verlag, 2006.
- [34] M. F. R. Rondon, L. Sassatelli, R. A. Pardo, and F. Precioso, "Revisiting deep architectures for head motion prediction in 360° videos," Nov. 2019, *arXiv:1911.11702*.
- [35] 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 *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1161–1170.
- [36] *Subjective Video Quality Assessment Methods for Multimedia Applications*, document ITU-T Rec. P.910, International Telecommunication Union, Geneva, Switzerland, 2009.
- [37] A. Yaqoob, T. Bi, and G.-M. Muntean, "A DASH-based efficient throughput and buffer occupancy-based adaptation algorithm for smooth multimedia streaming," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 643–649.
- [38] A. Yaqoob, T. Bi, and G.-M. Muntean, "A priority-aware dash-based multi-view video streaming scheme over multiple channels," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2020, pp. 297–303.
- [39] N. Barman, N. Khan, and M. G. Martini, "Analysis of spatial and temporal information variation for 10-bit and 8-bit video sequences," in *Proc. IEEE 24th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–6.
- [40] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Härmäläinen, "Kvazaar: Open-source HEVC/H.265 encoder," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 1179–1182, doi: 10.1145/2964284.2973796.
- [41] J. Van Der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "HTTP/2-based adaptive streaming of HEVC video over 4G/LTE networks," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2177–2180, Nov. 2016.



University, Ireland. His research interests include immersive multimedia processing and delivery solutions, priority-aware multimedia video streaming, QoE, and mobile wireless communication.



blockchain-enabled solutions for next generation networks (5G/6G). He has published more than 25 peer-reviewed scientific papers in top journals and flagship conferences. He has served as a member of the technical programme committee of several international conferences and reviewer for top international journals.



formance, and energy efficiency issues related to rich media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, and chair, technical programme committee member, and reviewer of top international journals and conferences. He coordinated the EU Horizon 2020 project NEWTON (<http://www.newtonproject.eu>) and leads the DCU team in the EU project TRACTION (<https://www.traction-project.eu>).

...