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

ABID YAQOOB1,2, (Student Member, IEEE), MOHAMMED AMINE TOGOU1,2, (Member, IEEE), GABRIEL-MIRO MUNTEAN1,2, (Senior Member, IEEE)

1Insight Centre for Data Analytics, Dublin City University, Ireland.
2Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Ireland.

Corresponding author: Gabriel-Miro Muntean (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 numbers 12/RC/2289_P2 (Insight SFI Centre for Data Analytics) and 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. 1 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.

1The equirectangular image used is sourced from GoPro VR-Tahiti Surf video available at https://www.youtube.com/watch?v=MkWWh8RAV8
The ever-increasing immersive real-time rendering causes tremendous bandwidth consumption since the entire captured content, including the visible and non-visible scenes, has to be downloaded in higher resolution (≥8k) Viewport-adaptive streaming is a straightforward approach to reduce the transmission bandwidth, where the quality of the scene is dynamically adjusted in real-time depending on the user’s Field-of-View (FoV). 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 adaptation 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. Such solutions work reasonably well under 1s segment duration with an accuracy of 58~80% 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. Some solutions consider prediction angles as well as prediction errors to estimate the viewport. However, they do not involve any neighbor region or viewpoint extension to truly reflect the proactive streaming. Moreover, including neighboring tiles surrounding viewpoint in all directions or sub-areas-based streaming 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.

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 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, DVS solutions do not include the neighboring regions throughout the streaming session (evidenced by Table). The extended viewpoint in DVS1 and DVS2 solutions is more profound than 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 4x3, 6x4, and 8x6 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 view-
based streaming; (2) Neighbor region-based adaptive streaming into three main categories: (1) Fixed viewport-based adaptive 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 methods are presented 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.

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 et al. [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 involv-
TABLE 1: Viewport selection solutions in tile-based adaptive 360° video streaming

| Works | Prediction Angles | Viewport Selection | Prediction Errors | Streaming Region | Videos | VR Traces | Resolution | Experimental Evaluations | Tiling | Exp. Duration |
|-------|------------------|--------------------|-------------------|------------------|--------|-----------|------------|--------------------------|--------|--------------|
| [14]  | ✓                | ✓                  | ✓                 | Fixed VP         | 2      | 2         | ✓          | 1.067s                   | ✓      | 84 frames    |
| [15]  | ✓                | ✓                  | X                 | Fixed VP         | 8      | 2K        | ✓          | 8, 10, 12 m/s            | ✓      | Video Duration|
| [16]  | ✓                | ✓                  | ✓                 | Fixed VP         | 3      | 4K        | ✓          | 16.5, 24.2, 44.4, 84.6, 88.8, 161.2/s | ✓      | Video Duration|
| [17]  | ✓                | ✓                  | ✓                 | Fixed VP         | 3      | 4K        | ✓          | 5.6, 10, 15, 20 m/s       | ✓      | Video Duration|

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. 5a) for 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. [29] 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 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.

Hooft et al. [8] proposed the center tile first (CTF) scheme, which mainly focuses on the quality improvements starting from the center tile similar to [9, 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. NSF-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 viewpoint versions. Table 1 highlights the significant differences between the proposed and existing viewpoint 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 [3]. 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 planer format, i.e., equirectangular format [33]. The transformed video content is divided into several segments, i.e., $S = \{S(1), S(2), ..., S(i), ..., S(f)\}$. The segmented videos are further divided into a number of rectangular grids/tiles, i.e., $T = \{T^{1}(i), T^{2}(i), ..., T^{k}(i), ..., T^{K}(i)\}$. The tiled segments are then prepared into multiple quality representations, i.e., $L = \{L^{1}_{k}(i), L^{2}_{k}(i), ..., L^{j}_{k}(i), ..., L^{J}_{k}(i)\}$. Let $L^{j}_{k}(i)$ 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 selects suitable tiles to stream belonging to different priority regions, i.e., viewport ($T^{v}(i)$), neighboring ($T^{n}(i)$), and background ($T^{b}(i)$), 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 viewpoint 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 P910 [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 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 $(i)\text{th}$ segment. The standard deviation reflects how much dispersion occurs over a while. The variation for the $(i)\text{th}$ segment is given as:

$$ST(i) = \text{std}(S(i) \ast T(i) : S(i-1) \ast T(i-1))$$  \hspace{1cm} (1)

The content complexity for $(i)\text{th}$ segment is approximated as follows:

$$CC(i) = \frac{ST(i)}{\max\{ST(i)|i \in [1:S]\}}$$  \hspace{1cm} (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 ($TP(i)$) and secondary viewport ($T^s(i)$) tiles sets considering the predicted coordinates and viewport size ($VS$). 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 $ADD(i)$ 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 $\Omega_1$ and $\Omega_2$ represents the thresholds of great circular distance between two predicted viewpoints, such as $\Omega_1 < \Omega_2$ and $\Omega_1, \Omega_2 \in [0,VS]$. If the adjusted distance between two predicted viewpoints is less than $\Omega_1$ for the $(i)\text{th}$ 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 $\Omega_1$ and $\Omega_2$, 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 $(i)\text{th}$ 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 $\Omega_2$. 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 $(i)\text{th}$ 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 $\Omega_1'$ and $\Omega_2'$ represents the thresholds of great circular distance between original and predicted viewpoints, such as $\Omega_1' < \Omega_2'$ and $\Omega_1', \Omega_2' \in [0,VS]$. Suppose the adjusted distance between predicted and actual viewpoints during $(i-1)\text{th}$ segment is not greater than the $\Omega_1'$. This refers to Fixed viewport scenario, where only the viewport and background tiles sets are calculated by considering the predicted angles and the actual viewpoint 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...
Algorithm 1: Viewport Selection Algorithm-DVS1

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

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

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

Algorithm 2: Viewport Selection Algorithm-DVS2

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

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

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

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

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

1. if $(\hat{B}(i) \leq \sum_{k \in T(i)} L^b_k(i))$ then
   2. $L^T(i) = L^b_k(i)$, forall $k \in T(i)$
   3. end
4. else if $(\hat{B}(i) \geq \sum_{k \in T(i)} L^b_k(i))$ then
   5. $L^T(i) = L^b_k(i)$, forall $k \in T(i)$
   6. end
7. else
   8. $L^T(i) = L^b_k(i)$, forall $k \in T(i)$
   9. $B(i) = \hat{B}(i) - \sum_{k \in T(i)} L^b_k(i)$
10. if $(|T^n(i)| == 0)$ then
11.     if $(\hat{V}_S \neq 0)$ then
12.         $L^T(i) = \max_j \{L^b_j(i)|L^b_j(i) \leq B(i) - \sum_{k \in T(i)} \sum_{j \in [2:j]} L^b_k(i)\}$
13.     end
14. else
15.     $B^v(i) = B(i)$
16.     $L^{T^v}(i) = \max_j \{L^b_j(i)|\sum_{k \in T^v(i)} L^b_k(i) \leq B^v(i)\}$
17.     $B^n(i) = B(i) - (\sum_{k \in T^n(i)} L^b_k(i))$
18.     $L^{T^n}(i) = \max_j \{L^b_j(i)|\sum_{k \in T^n(i)} L^b_k(i) \leq B^n(i)\}$
19. end
20. else
21.     $w^{T^n}(i) = (|T^n(i)|/((2 \ast |T^v(i)| + |T^n(i)|))$
22.     $w^{T^v}(i) = 1 - w^{T^n}(i)$
23.     $B^v(i) = B(i) \ast w^{T^v}(i)$
24.     $B^n(i) = B(i) \ast w^{T^n}(i)$
25.     $L^{T^v}(i) = \max_j \{L^b_j(i)|\sum_{k \in T^v(i)} L^b_k(i) \leq B^v(i)\}$
26.     $L^{T^n}(i) = \max_j \{L^b_j(i)|\sum_{k \in T^n(i)} L^b_k(i) \leq B^n(i)\}$
27.     $B^b(i) = B(i) - (\sum_{k \in T^n(i), T^v(i)} L^b_k(i))$
28.     $L^{T^b}(i) = \max_j \{L^b_j(i)|\sum_{k \in T^b(i)} L^b_k(i) \leq B^b(i)\}$
29. end
30. end
31. end

FIGURE 6: Viewport selection and bitrate adaptation scenarios in DVS solutions.
(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 $Ω$. 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 1 determines the appropriate bitrate selection for the tiles belonging to different priority regions. After completely downloading the previous segment, the playback adaptation is performed for each segment. Algorithm 2 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 viewpoint content at a higher quality than the neighboring tiles, higher priority weights are allocated to the viewpoint tiles. The viewpoint and neighboring tiles’ bandwidth are computed based on priority-related weights. Next, the quality allocation for the viewpoint 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 viewpoint 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 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 [3] involving recorded head movement patterns to perform the experiments 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°.

![Image](https://creativecommons.org/licenses/by/4.0/)
Three different motion 360° videos with at least 4K resolution were sourced from YouTube: (1) Conan360°-Sandwich; (2) Google Spotlight-HELFI; (3) GoPro VR-Tahiti Surf. 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 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 4x3 (12), 6x4 (24), and 8x6 (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 video segments are produced using GPAC MP4Box. 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 \( \Omega_1, \Omega_2, \Omega_1', \) and \( \Omega_2' \). The reported experimental

---

Three different motion 360° videos with at least 4K resolution were sourced from YouTube: (1) Conan360°-Sandwich; (2) Google Spotlight-HELFI; (3) GoPro VR-Tahiti Surf. 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 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 4x3 (12), 6x4 (24), and 8x6 (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 video segments are produced using GPAC MP4Box. 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 \( \Omega_1, \Omega_2, \Omega_1', \) and \( \Omega_2' \). The reported experimental
results are obtained by setting $\Omega_{1} = V S / 3$ and $\Omega_{2} = V S / 2$. 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 trace 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

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 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 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 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.

#### 1) Streaming Tiles Selection

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

#### 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 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) next 1s (Fig. 10a), and 13.97% and 15.07% for next 2s (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 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 6x4 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 6x4 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). The HOS method undergoes 33.81% quality fluctuations for 6x4 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. 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.
Yaqoob et al.: Dynamic Viewport Selection-based Prioritized Bitrate Adaptation for Tile-based 360° Video Streaming

![Graph](image)

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

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.

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 view-
ing 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. CONCLUSIONS
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] David Li, Ruofei Du, Adharsh Babu, Camelia D. Brumar, and Amithabh Varshney. A Log-Rectilinear Transformation for Foveated 360-degree Video Streaming. IEEE Transactions on Visualization and Computer Graphics, 27(5):2638–2647, 2021.
[2] Taehyun Rhee, Lohit Petikam, Benjamin Allen, and Andrew Chalmers. MR360: Mixed Reality Rendering for 360° Panoramic Videos. IEEE Transactions on Visualization and Computer Graphics, 23(4):1379–1388, 2017.
[3] A. Yaqoob, T. Bi, and G. M. Muntean. A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities. IEEE Communications Surveys Tutorials, 22(4):2801–2838, 2020.
[4] Ching-Ling Fan, Wen-Chih Lo, Yu-Tung Pai, and Cheng-Hsin Hsu. A Survey on 360° Video Streaming: Acquisition, Transmission, and Display. ACM Comput. Surv., 52(4), August 2019.
[5] Zhiqian Jiang, Xu Zhang, Yiling Xu, Zhan Ma, Jun Sun, and Yunfei Zhang. 360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based Adaptive Streaming. IEEE Journal on Emerging and Selected Topics in Information Systems, 2019, 2019.
[6] Xiaoli Chen, Di Wu, and Ishfaq Ahmad. Optimized viewport-adaptive 360-degree video streaming. CAAI Transactions on Intelligence Technology, 2021.
[7] Cagri Ozcinar, Ana De Abreu, and Aljosza Smolic. Viewport-aware adaptive 360 video streaming using tiles for virtual reality. In 2017 IEEE International Conference on Image Processing (ICIP), pages 2174–2178, IEEE, 2017.
[8] Jeroen Van der Hooft, Maria Torres Vega, Stefano Petrangeli, Tim Wauters, and Filip De Turck. An HTTP/2-Based Adaptive Streaming Framework for 360° Virtual Reality Videos. In ACM Multimedia Conference, pages 306–314, ACM, 2017.
[9] Kevin Sipteri, Rahul Urgaonkar, and Ramesh K Sitaranam. BOLA: Near-optimal bitrate adaptation for online videos. In INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, pages 1–9. IEEE, 2016.
[10] Dimitrios J Vergados, Angelos Michalas, Aggeliki Sgora, Dimitrios V Vergados, and Periklis Chatzimisios. FDASH: A fuzzy-based MPEG/DASH adaptation algorithm. IEEE Systems Journal, 10(2):859–868, 2016.
[11] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, pages 197–210, 2017.
[12] Junchen Jiang, Vyas Sekar, and Hui Zhang. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming With FESTIVE. IEEE/ACM Transactions on Networking (TON), 22(1):326–340, 2014.
[13] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mary Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. ACM SIGCOMM Computer Communication Review, 44(4):187–198, 2015.
[14] Saba Qasim Jabbar, Dheyaa Jasim Kadhim, and Yu Li. Proposed an Adaptive Bitrate Algorithm based on Measuring Bandwidth and Video Buffer Occupancy for Providing Smoothly Video Streaming. technology, 9(2), 2018.
[15] Meiue Hu, Wei Quan, Tao Guo, Yu Liu, and Lin Zhang. Mobile Edge Assisted Live Streaming System for Omnidirectional Video. Mobile Information Systems, 2019, 2019.
[16] Abid Yaqoob and Gabriel-Miro Muntean. A Combined Field-of-View Prediction-Assisted Viewport Adaptive Delivery Scheme for 360° Videos. IEEE Transactions on Broadcasting, 67(2):409–423, 2021.
[17] Mohammad Hosseini and Viswanathan Swaminathan. Adaptive 360 VR Video Streaming: Divide and Conquer. In Multimedia (ISM), 2016 IEEE International Symposium on, pages 107–110. IEEE, 2016.
[18] M. Dasari, A. Bhattacharya, S. Vargass, P. Sahu, A. Balasubramanian, and S. R. Das. Streaming 360-Degree Videos Using Super-Resolution. In IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, pages 1977–1986, 2020.
[19] Jiang Yu and Yong Liu. Field-of-View Prediction in 360-Degree Videos with Attention-based Neural Networks. In Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems, pages 37–42. ACM, 2019.
[20] Cheng Li, Weixi Zhang, Yong Liu, and Yao Wang. Very Long Term Field of View Prediction for 360-degree Video Streaming. In 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pages 297–302. IEEE, 2019.
[21] Anh Nguyen, Zhisheng Yan, and Klara 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, pages 1190–1198. ACM, 2018.
[22] Duc V Nguyen, Huyen TT Tran, Anh T Pham, and Truong Cong Thang. An Optimal Tile-Based Approach for Viewport-Adaptive 360-Degree Video Streamings. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9(1):29–42, 2019.
[23] Yuning Hu, Yu Liu, and Yumei Wang. VAS360: QoE-Driven Viewport Adaptive Streaming for 360 Video. In 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pages 324–329. IEEE, 2019.
[24] Nuowen Kan, Junni Zou, Chenglin Li, Wenrui Dai, and Hongkai Xiong. RAPT360: Reinforcement Learning-Based Rate Adaptation for 360-degree Video Streaming with Adaptive Prediction and Tiling. IEEE Transactions on Circuits and Systems for Video Technology, pages 1–1, 2021.
[25] Stefano Petrangeli, Viswanathan Swaminathan, Mohammad Hosseini, and Filip De Turck. An HTTP/2-Based Adaptive Streaming Framework for 360° Virtual Reality Videos. In ACM Multimedia Conference, pages 306–314. ACM, 2017.
[26] Kevin Sipteri, Rahul Urgaonkar, and Ramesh K Sitaranam. BOLA: Near-optimal bitrate adaptation for online videos. In INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, pages 1–9. IEEE, 2016.
[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 Proceedings of the ACM 21st International Workshop on Multimedia Signal Processing (MPS), pages 1–6, Sep. 2019.
[28] Sohee Park, Arani Bhattacharya, Zhibo Yang, Mallesh Dasari, Samir R. Das, and Dimitris Samaras. Advancing User Quality of Experience in 360-Degree Video Streaming. In 2019 IFIP Networking Conference (IFIP Networking), pages 1–9, 2019.
[29] Nuowen Kan, Junni Zou, Kexin Tang, Chenglin Li, Ning Liu, and Hongkai Xiong. Deep Reinforcement Learning-based Rate Adaptation for Adaptive 360-Degree Video Streaming. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4030–4034. IEEE, 2019.
[30] Dongbiao He, Cedric Westphal, and J Garcia-Luna-Aceves. Joint Rate and FoV Adaptation in Immersive Video Streaming. In ACM Sigcomm workshop on AR/VR Networks, 2018.

[31] Feng Qian, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan. Optimizing 360 Video Delivery Over Cellular Networks. In Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, pages 1–6. ACM, 2016.

[32] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In Proceedings of the 8th ACM on Multimedia Systems Conference, pages 193–198, 2017.

[33] David Salomon. Transformations and Projections in Computer Graphics. Springer-Verlag, Berlin, Heidelberg, 2006.

[34] Miguel Fabian Romero Rondon, Lucile Sassatelli, Ramon Aparicio Pardo, and Frederic Precioso. Revisiting Deep Architectures for Head Motion Prediction in 360° Videos. arXiv e-prints, page arXiv:1911.11702, November 2019.

[35] Yanan Bao, Huasen Wu, Tianxiao Zhang, Albara Ah Ramli, and Xin Liu. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In 2016 IEEE International Conference on Big Data (Big Data), pages 1161–1170. IEEE, 2016.

[36] ITU-T Recommendation P.910. Subjective video quality assessment methods for multimedia applications. International Telecommunication Union, Geneva, 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 15th Int. Wireless Communications Mobile Computing Conference (IWCMC), pages 643–649, June 2019.

[38] Abid Yaqoob, Ting Bi, and Gabriel-Miro Muntean. A priority-aware dash-based multi-view video streaming scheme over multiple channels. In Int. Wireless Communications and Mobile Computing (IWCMC), pages 297–303, 2020.

[39] Nabajyet Barman, Nabeel Khan, and Maria G Martini. Analysis of spatial and temporal information variation for 10-bit and 8-bit video sequences. In 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pages 1–6. IEEE, 2019.

[40] Marko Viitanen, Ari Koivula, Ari Lemmetti, Arttu Ylä-Outinen, Jarno Vanne, and Timo D. Hamäläinen. Kvazaar: Open-Source HEVC/H.265 Encoder. In Proceedings of the 24th ACM International Conference on Multimedia, 2016.

[41] Jeroen Van Der Hooft, Stefano Petrangeli, Tim Wanters, Rafael Huysegems, Patrice Rondao Alfate, Tom Bostoen, and Filip De Turck. Http2-based adaptive streaming of hevc video over 4g/lte networks. IEEE Communications Letters, 20(11):2177–2180, 2016.

***