RESEARCH ARTICLE

Viewport History as a Heuristic for Quality Enhancement and Quality Variation Control in Viewport-Aware Tile-Based 360-Degree Video Streaming

KIANA DZIUBINSKI AND MASAKI BANDAI\(^{(\text{Member, IEEE})}\)
Faculty of Science and Technology, Sophia University, Tokyo 102-8554, Japan
Corresponding author: Kiana Dziubinski (kianadziubinski@eagle.sophia.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant 19K11931.

ABSTRACT Despite the growing popularity of Virtual Reality (VR), 360-degree videos are often regarded as challenging to stream due to their large bandwidth requirement. As a solution, the 360-degree video content is spatially divided into tiles, and the quality level for each tile is selected based on the user’s network environment and viewport information. To determine the high quality tiles, viewport prediction and viewport history methods are used to estimate the user’s viewport. However, due to the unpredictability of user head movements, generating accurate viewport estimates are difficult, which can severely degrade the Quality of Experience (QoE) for the user. In this paper, to sustain high user QoE, we detail a novel tile quality selection algorithm that employs viewport prediction, viewport history, viewport extensions, and a viewport tile count limit. In addition, we also include comparison analysis on six 360-degree videos that vary in content pace. Based on simulations, using viewport history as a heuristic for tile quality selection demonstrated a significant increase in the perceived quality while suppressing quality variation inside the viewport and across segments compared to eight reference methods; and secondly, 360-degree videos slow in content pace tended to result in lower viewport prediction accuracy, QoE performance, and weaker viewport history trends compared to 360-degree videos fast in content pace.

INDEX TERMS 360-degree video streaming, adaptive streaming, virtual reality (VR).

I. INTRODUCTION

As of late, 360-degree video streaming technology such as Virtual Reality (VR) hold great promise to various sectors of society. These sectors include the architecture, education, entertainment, defense, medical, and sport industry [1], [2]. Through this technology, users with a Head Mounted Display (HMD) can enter complete, immersive virtual environments and can freely adjust their Field of View (FoV)/viewport of the virtual scene by adjusting their head orientation. However, there are challenges in delivering 360-degree videos to users. In particular, these challenges include the bandwidth requirement, the motion-to-photon (MTP) latency, and the user Quality of Experience (QoE) [3]. Moreover, in consideration of the growing popularity of VR and related technology, such challenges are imperative for research communities and industries to resolve in order to support 360-degree video streaming at large scales [3]. The three challenges are further described in the following.

The first challenge is satisfying the bandwidth requirement to stream 360-degree video content. Unlike 2D video content, the bandwidth requirement is magnitudes larger for 360-degree video content. For example, to stream 360-degree and 2D video content in 4K, the data rate is 400Mbps and 25Mbps, respectively [3]. However, the user’s viewport covers less than 20% of the entire 360-degree virtual scene [4], meaning that over 320Mbps from the above 400Mbps of data is needlessly expended. This data wastage is particularly
severe for 360-degree video streaming applications because the video content is typically delivered in high video resolutions such as 4K and 8K. Nevertheless, streaming only the portion perceived by the user in high quality is the most optimal for the network.

The second challenge is reducing the MTP latency for the user’s comfort and virtual experience. MTP latency is defined as the amount of time between the input of the user’s head movement and the corresponding output of the visual feedback on the user’s HMD. A large MTP latency can induce discomfort such as cybersickness and the loss of virtual immersion to the user [5], [6]. To avoid inducing cybersickness to the user, the visual feedback of the viewport must be in high quality and consistent with an end-to-end latency or MTP latency under 15-20 milliseconds [5]. Then, to maintain the immersive virtual experience, a MTP latency under 10 milliseconds is ideal [6]. This places a tight constraint in the latency and stability of the network to deliver the current viewport in high quality.

Lastly, the third challenge is maintaining a high user QoE. The QoE elements that affect the user’s viewing experience in 360-degree video streaming are the perceived quality, quality switching, and rebuffering [4], [7], [8], [9]. Here, the perceived quality refers to the bitrate of the viewport; quality switching refers to the quality consistency of the viewport; and rebuffering refers to the stalling of the video playback. To achieve the best viewing experience, the perceived quality should be kept at a maximum while both quality switching and rebuffering should be kept at a minimum. However, achieving such a viewing experience creates a tradeoff with network resources.

In this paper, we primarily focus on resolving the first and third challenge of the bandwidth requirement and user QoE, respectively, for on demand tile-based 360-degree video streaming. Tile-based streaming is one of two existing adaptation methods for 360-degree video streaming [10], [11], [12], [13], [14], with the other being monolithic streaming [15]. Tile-based streaming holds the merit of decreasing the bandwidth requirement for 360-degree video streaming by dividing the video content temporally into segments and spatially into tiles, wherein each tile can be requested at a different quality level. The quality level of each tile is selected based on the user’s network environment and viewport information, and to realize high user QoE with conservative network resources, only the tiles inside the user’s viewport are requested in high quality levels. However, due to the round trip delay of the request, it becomes necessary to prefetch the tiled segment in advance in order to maintain the immersive virtual experience and to avoid both cybersickness and rebuffering for the user. Therefore, viewport prediction methods must be used to estimate the user’s anticipated viewport in advance. In practice, viewport predictions for one second in advance yield an accuracy ranging from 58% to 80% [7], [16], and this accuracy decreases further with time, as presented in Figure 1 for example. Viewport mispredictions result in a degraded user QoE according to the extent of the viewport misprediction. This is because the tiles comprising the user’s viewport will be displayed entirely or partially in low quality. Based on this tradeoff between viewport prediction accuracy and user QoE, tiled segments are typically prefetched in advance by 1~2 seconds [10]. Larger segments are generally preferred over shorter segments by incurring less network overhead, but larger segments provide less adaptability to changes in the user’s network environment and viewing experience [15]. Therefore, reducing the severity of viewport mispredictions is imperative for maintaining high user QoE with respect to the network environment.

To improve user QoE due to viewport mispredictions, we propose a viewport prediction-based and viewport history-based tile quality selection algorithm for on demand tile-based 360-degree video streaming. The proposed algorithm uses the viewport information of past users as a heuristic for quality control inside the viewport and for performing viewport extensions. Moreover, unlike our conventional algorithm in [18], we expand this work by improving the algorithmic logic during the viewport enhancement and viewport extension phase to further increase the perceived quality and reduce quality switching across segments. In addition, we also include four additional 360-degree video contents to the evaluation environment. This is for highlighting tendencies concerning the viewport prediction accuracy, viewport history, and user QoE in relation to the video content pace.

The main contributions of this paper are as follows.

- We propose, to our knowledge, the first formally combined viewport prediction-based and viewport history-based tile quality selection algorithm.
- We propose a viewport tile count limit as a control means to increase the perceived quality and suppress quality variation across segments.
- We demonstrate the difference in viewport prediction accuracy, viewport history, and user QoE performance in relation to the content pace of the 360-degree video.
- We demonstrate the difference in user QoE performance in relation to viewport prediction-based and viewport history-based tile quality selection methods.
The rest of this paper is organized as follows. In section II, we introduce the background and related work. In section III, we describe our proposed system architecture and tile quality selection algorithm. In section IV, we present the evaluation environment, user QoE evaluation index, and performance evaluation of our proposed algorithm against the conventional algorithm and seven other reference methods. In section V, we describe our discussion on the results. Lastly, in section VI, we conclude this work and describe our prospects.

II. RELATED WORK
In this section, we introduce the background and related work behind the proposed method.

A. VIEWPORT PREDICTION
In practice, 360-degree video streaming is difficult to efficiently execute in terms of network resources because of the unpredictability of user head movements. To perform viewport predictions, solutions include extracting sensor information such as head orientations and content-related features such as saliency maps and motion maps from the 360-degree video [19]. This information can also be deduced by users’ viewport history, however [20], [21]. In this paper, we adopt a linear regression method based on sensor information to predict the user’s viewport for the next segment as in [11]. The linear regression method uses the roll, pitch, and yaw head orientation information contained in the time interval of $t_0 - 1$ to $t_0$ to predict the user’s future head orientation for a time $t_0 + \Delta$, where $\Delta$ denotes the prediction window size. We adopt a simple viewport prediction method because it is not within the scope of this paper to design such a method. In addition, the prediction method is not limited to linear regression for the proposed method.

Figure 1 presents the average accuracy of the linear regression method for all 50 users at each prediction window size of 1s, 2s, and 4s in the Roller Coaster, Pac-Man, Chariot Race, Kangaroo Island, Perils Panel, and Shark Shipwreck dataset [17]. Here, the prediction is considered accurate if the predicted roll, pitch, and yaw of the user’s head orientation together deviate less than 10 degrees from the user’s actual head orientation. Notably, the accuracy of the prediction decreases for longer prediction intervals, especially for video content slow in content pace by referring to Table 1. Based on this, it can be inferred that there is a tendency for greater degrees of head movement variation in slow-paced content than fast-paced content.

B. VIEWPORT EXTENSIONS
Viewport extensions can reduce the severity of viewport mispredictions. This is because viewport extensions enlarge the predicted viewport by increasing the number of high quality tiles surrounding the predicted viewport. There are different techniques to perform viewport extensions. For example, references [11], [22] enlarge the predicted viewport by enhancing the surrounding tiles according to the degree of the last viewport prediction error, but [22] formally details an algorithm when to enhance the quality of all surrounding tiles or a certain direction of surrounding tiles; reference [13] gradually reduces the quality level of all surrounding tiles-based on their distance to the viewport center; and reference [23] classifies all tiles as either viewport, adjacent, or outside, and enhances each tile group accordingly.

However, there is a limit to how much the viewport can be extended and to what quality. This means that it is just as important as viewport predictions to select effective viewport extensions. Accordingly, to realize effective viewport extensions, we showed in our previous work of [18] that users’ viewport history can be used, even if the viewport was mispredicted. In addition, as a viewport extension technique, we do not suggest using strict tile group classifications like in [23] because severe quality level gaps between tiles can occur when there is a misprediction. In detail, this happens because in [23] all tiles located in the north and south poles of the equirectangular projection, the most common two-dimensional plane projection method for 360-degree content [10], are classified as outside tiles when not a viewport tile, and the quality level difference between the two tile groups of viewport and outside, especially if large, can further decrease user QoE by inducing higher quality variation inside the viewport when mispredicted [8], [9]. Therefore, to avoid inducing such quality variation, all tiles should be initially considered as a candidate for quality enhancement instead of predetermining which tiles are outside/background tiles.

C. VIEWPORT HISTORY
Viewport history is simply the aggregated viewport information of multiple users for a 360-degree video. To illustrate, Figure 2 presents the viewport history of 50 users for 60s of video content in a 5 × 5 equirectangular tile map, in which the popularity of a tile is defined as the affinity score from Equation 1. As illustrated, viewport history can characterize popular tiles within the 360-degree space and can also assist with viewport prediction, viewport extensions, bandwidth allocation, and caching [14], [18], [21], [24], [25]. In particular, references [14], [21], [24] directly apply viewport history. Reference [14] uses tile popularity with the distortion characteristics of the projected video to determine the quality level of each tile; reference [21] proposes an adaptation method for tile-based streaming that transitions from viewport prediction-based to tile-popularity-based according to the network environment; and reference [24] prefetches the enhancement layers of select tiles in advance from a descending order tile popularity list for scalable video coding.

However, when only non-user dependent features such as tile popularity are used as the basis for tile quality selection, the user is discouraged from looking around the virtual space. This is because only such tiles will be predominantly encoded in higher quality, irrespective of the user’s viewport information. Therefore, to promote the user’s individualized viewing experience, non-user dependent features should be used as a supplementary tool for viewport-aware systems.
FIGURE 2. User affinity of tiles from sample size of 50 users for 60 s of video content in 5 × 5 equirectangular tile map.

such as viewport prediction and viewport extensions. Moreover, the popular content is not commonly shaped among videos. As further described in section IV, the shape of the popular content can affect user QoE for such tile quality selection methods. For example, in Figure 2, the six videos generally share a common focal point towards the center, but the slow-paced videos tended to yield a less prominent focal point that is also more dispersed, and this can result in distributed network resources. Lastly, as exhibited in Figure 1, this dispersion is similarly reasoned to be the result of the user looking around the virtual space more often and prominently.

III. PROPOSED METHOD

In this section, we introduce the proposed system architecture and tile quality selection algorithm. The proposed method is for on demand 360-degree video streaming. In the proposed method, the server collects and prepares the users’ viewport information in advance.

A. SYSTEM ARCHITECTURE

Figure 3 presents the proposed system architecture. Similar to typical client and server systems using Dynamic Adaptive Streaming over HTTP Spatial Relationship Description for tile-based 360-degree video content, the server holds the video in \( L \) quality levels for \( M \) segments, and each segment is spatially divided into \( N \) tiles. The server stores the above information inside its Media Presentation Description (MPD) file. In the proposed system, however, the server also holds information on the popularity of each tile, and this information is also stored inside the MPD file.

To start the streaming session, the client downloads the server’s MPD file, and the client’s MPD Parser confirms the quality and number of tiles that are available. From reading the MPD file, the user’s Adaptive Bit Rate controller selects the appropriate quality for each tile according to the client’s network environment and viewport information. Afterwards, the client sends a request for the tiled segment to the server using a HTTP/GET request, and when the requested segment arrives, the client places the segment inside a buffer and plays the video content sequentially from the buffer.

To determine the popularity of a tile, the tile’s affinity score is calculated. The affinity score of a tile for a given segment is derived from the viewport history of all past users. In particular, to calculate the affinity score \( a_{i,j} \) for tile \( j \) in segment \( i \), the mean number of occurrences tile \( j \) overlapped with all users’ viewports for all frames contained in segment \( i \) is taken, where \( a_{i,j} \) in Equation 1 represents the normalized tile affinity score. The affinity score of a tile ranges from 0~1. Accordingly, an affinity score closer to 1 indicates a tile greater in popularity. This affinity score calculation is performed in advance by the server, and the server can perform this calculation as needed.

\[
\bar{a}_{i,j} = \frac{a_{i,j}}{\sum_j a_{i,j}}
\]  

B. TILE QUALITY SELECTION ALGORITHM

To address viewport mispredictions and quality variation, we adopt tile affinity scores as a heuristic for both tile quality level selection and control. Algorithm 1 presents the pseudocode of the proposed tile quality selection
algorithm, which consists of four phases. Namely, the four phases are Exceptional Initial, Quality Initialization, Viewport Enhancement, and Extension Enhancement, respectively. Each phase is executed in the order above for a segment.

Regarding the input, Algorithm 1 takes the perceived bandwidth \( B \), the bitrate of the selected tile quality level \( b(l) \), the segment number \( M \), the number of tiles in a frame \( N \), the number of tiles in the viewport \( N_{\text{viewport}} \), the number of tiles in the selected viewport extension group \( N_{\text{Sub-Adj.}} \), the viewport tile count limit \( \text{cutoff} \), and the number of demoted viewport tiles \( N_{\text{cutoff}} \) to output the assigned quality level of a tile \( Q(j) \) for all tiles in the requested segment. The four phases of Algorithm 1 are described in detail next.

1) EXCEPTIONAL INITIAL PHASE
First, in the Exceptional Initial phase at L1-L4, the segment number is confirmed to determine the viewport estimation method. If it is the first segment, then a history-based method is used; otherwise, a prediction-based method is used. This binary check at L1 is specifically for the first segment because viewport prediction is unavailable during this segment, but this causes the first segment to be in low quality \([13]\). Therefore, to conservatively begin the playback in high quality, we define the history-based viewport to be the top nine tiles in affinity scores in this paper. This is because in a 5 \( \times \) 5 equirectangular tile map, the top nine tiles typically hold an affinity score sum over 90% and encompass all center tiles in the tile map, and because nine tiles can cover over the average number of tiles perceived by a user at a given frame.

2) QUALITY INITIALIZATION PHASE
Second, in the Quality Initialization phase at L5-L7, all tiles are allocated bandwidth. This is to ensure all tiles are delivered to the user. At L5-L6, all tiles are assigned the lowest quality level, and these tiles are called background tiles in Figure 4. Lastly, at L7, the consumed bandwidth is deducted from the perceived bandwidth, now called the bandwidth budget \( B_{\text{budget}} \).

3) VIEWPORT ENHANCEMENT PHASE
Third, in the Viewport Enhancement phase at L8-L26, the viewport is enhanced. This phase can be divided into two parts, with the first part being L8-L11 and the other L12-L26.

In the first part, all viewport tiles are enhanced to the highest possible quality level. At L8, the bandwidth used in the Quality Initialization phase is rewarded back to the bandwidth budget for the viewport tiles. At L9-L10, all viewport tiles are uniformly assigned the highest possible quality level, and these tiles are called viewport tiles in Figure 4.

**Algorithm 1: Tile Quality and Viewport Extension Selection**

| Input: | \( B \) \( \leftarrow \) bandwidth available [Kbps] \( b(l) \) \( \leftarrow \) bitrate of quality level \( j \) [Kbps], \( M \) \( \leftarrow \) segment number \( N \) \( \leftarrow \) all tiles in frame \( N_{\text{viewport}} \), \( N_{\text{Sub-Adj.}} \) \( \leftarrow \) number of tiles in tile group \( \text{cutoff} \) \( \leftarrow \) viewport tile count limit \( N_{\text{cutoff}} \) \( \leftarrow \) number of demoted viewport tiles |
|---|---|
| Output: | \( Q(j) \) \( \leftarrow \) assigned quality level \( I \) to tile \( j \) |

**Exceptional Initial:**
1. if \( \text{Seg} \ M = 1 \) then
2. Viewport \( \leftarrow \) History-based
3. else
4. Viewport \( \leftarrow \) Prediction-based

**Quality Initialization:**
5. for tile \( j \) in \( N \) do
6. \( Q(j) \) \( \leftarrow \) 0
7. \( B_{\text{budget}} \) \( \leftarrow \) \( B \) \( − \) \( N \) \( \times \) \( b(0) \)

**Viewport Enhancement:**
8. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( + \) \( N_{\text{viewport}} \) \( \times \) \( b(0) \)
9. for tile \( j \) in \( N_{\text{viewport}} \) do
10. \( Q(j) \) \( \leftarrow \) \( \text{max} \) \( \left\{ N_{\text{viewport}} \times b(l) \right\} \) \( \leq \) \( B_{\text{budget}} \)
11. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( − \) \( N_{\text{viewport}} \) \( \times \) \( b(l) \)
12. if \( l < L - 1 \) and \( N_{\text{viewport}} \) \( \geq \) \( \text{cutoff} \) then
13. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( + \) \( N_{\text{viewport}} \) \( \times \) \( b(0) \)
14. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( − \) \( N_{\text{viewport}} \) \( \times \) \( b(l) \)
15. for tile \( j \) in \( N_{\text{viewport}} \) \( \times \) \( \text{cutoff} \) do
16. \( Q(j) \) \( \leftarrow \) \( \text{max} \) \( \left\{ N_{\text{viewport}} \times b(l) \right\} \) \( \leq \) \( B_{\text{budget}} \)
17. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( − \) \( N_{\text{viewport}} \) \( \times \) \( b(l) \)
18. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( − \) \( N_{\text{viewport}} \) \( \times \) \( b(l) \)
19. for tile \( j \) in \( N_{\text{viewport}} \) \( \times \) \( \text{cutoff} \) do
20. \( Q(j) \) \( \leftarrow \) \( \text{max} \) \( \left\{ N_{\text{viewport}} \times b(l) \right\} \) \( \leq \) \( B_{\text{budget}} \) and \( l + 1 \) \( \leq \) \( N_{\text{viewport}} \)
21. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( − \) \( N_{\text{viewport}} \) \( \times \) \( b(l) \)
22. repeat
23. until \( B_{\text{budget}} \) or tile \( j \) unavailable

**Extension Enhancement:**
24. repeat
25. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( + \) \( N_{\text{Sub-Adj.}} \) \( \times \) \( b(0) \) /!\ in group rank order
26. for tile \( j \) in \( N_{\text{Sub-Adj.}} \) do
27. \( Q(j) \) \( \leftarrow \) \( \text{max} \) \( \left\{ N_{\text{Sub-Adj.}} \times b(l) \right\} \) \( \leq \) \( B_{\text{budget}} \) and \( l \) \( \leq \) \( N_{\text{viewport}} \)
28. \( B_{\text{budget}} \) \( \leftarrow \) \( B_{\text{budget}} \) \( − \) \( N_{\text{Sub-Adj.}} \) \( \times \) \( b(l) \)
29. repeat
30. until \( B_{\text{budget}} \) or tile \( j \) unavailable
31. until \( B_{\text{budget}} \) or Sub-Adj. unavailable
Lastly, at L11, the consumed bandwidth is deducted from the bandwidth budget.

However, unlike in [18], the Viewport Enhancement phase does not end here. Instead, a conditional argument is met at L12. This is because the conventional algorithm did not address quality switching due to the fluctuation in the number of viewport tiles. At a given frame and position in the virtual space, the number of tiles needed to reconstruct the user’s viewport can greatly vary. For example, in a 5 × 5 equirectangular tile map, the number of tiles needed to reconstruct the user’s viewport can range from 6 to 11 tiles, with the average ranging from 7.24 to 8.11 tiles, for all six videos and 50 users. As a result, quality switching between segments can occur if a limit on the number of viewport tiles to enhance is absent.

Therefore, in the second part, a viewport tile count limit called cutoff is used to rearrange the viewport quality level based on affinity scores. At L12, if the current quality level of the viewport is less than the highest quality level available and the number of viewport tiles exceeds cutoff, then the number of viewport tiles exceeding cutoff will be demoted in quality and the remaining viewport tiles will be promoted in quality. To determine which viewport tiles will be either demoted or promoted, ascending tile affinity scores are used, in which the viewport tiles holding the lowest affinity scores are demoted first. For example, if cutoff is 5 viewport tiles, then from the 6 viewport tiles presented in Figure 4b, only the viewport tile with the affinity score of 8.177 is demoted. Like this, at L13, the bandwidth used in the first part above is returned, and at L14 the consumed bandwidth in assigning the lowest quality level to all demoted viewport tiles is deducted. At L15-L16, the promoted viewport tiles are uniformly assigned to the highest possible quality level, and at L17 the consumed bandwidth is deducted. At L18, the bandwidth granted for the demoted viewport tiles at L14 is returned. Then, at L19-L20, the demoted viewport tiles are uniformly assigned to the highest possible quality level, such that the quality level is less than or equal to the quality level of the promoted viewport tiles, and at L21 the consumed bandwidth is deducted. Lastly, at L22-L26, in descending affinity score order, each demoted viewport tile is individually enhanced by one quality level, such that the quality level is less than or equal to both the bandwidth budget and promoted viewport quality level, and at L25 if and only if the individual demoted viewport tile is enhanced, the consumed bandwidth is accordingly deducted; L22-L26 are repeated until either the bandwidth budget or last demoted viewport tile is met. The purpose of L22-L26 is to further reduce the quality level gap between the promoted and demoted viewport tiles.

The value for cutoff is derived from empirical analysis concerning various factors. These factors include the tiling scheme, the average number of tiles comprising the viewport of users, the bitrates of the available quality levels, the segment size, and the perceived bandwidth available. Among these, the tiling scheme and the bitrates of the available quality levels hold the greatest influence. This is because these two factors are, in general, fixed and influence the outcome of the other factors. For example, a larger tiling scheme (i.e., more tiles) will result in a larger value in the average number of tiles comprising users’ viewport, and larger bitrates for the available quality levels will result in less high quality tiles for a given network environment. A “good” cutoff value will cover over the average number of tiles perceived by a user at a given frame, which is 7.24 to 8.11 tiles here; however, the bitrates of the available quality levels and the network environment can limit the value for cutoff. In this case, the cutoff value should be lowered from its “good” value until the best quality level can be achieved but still be within the perceived tile count range, which is 6 to 11 tiles here. Based on the above, in this paper for a 5 × 5 equirectangular tile map and the network environment described in section IV, we select a cutoff value of 8 viewport tiles.

4) EXTENSION ENHANCEMENT PHASE

Fourth and last, in the Extension Enhancement phase at L27-L37, the viewport is extended. Viewport extensions are performed by enhancing the tiles surrounding the viewport by creating direction groups. We consider the four direction groups of left, right, top, and bottom, and these tiles are called sub-adjacent tiles in Figure 4. We ignore corner tiles because of their peripheral location and to conserve network resources. We also do not set a fix ratio of bandwidth between the Viewport Enhancement and Extension Enhancement phase. This is because viewport prediction takes precedence over viewport history. After each direction group is defined, the rank of each group is determined by taking the average of the tile affinity scores comprising each group. For example, in Figure 4b, the group rank order is red,

---

**FIGURE 4.** Proposed tiling scheme, wherein tile affinity scores are multiplied by 100 for visual clarity.
green, pink, and blue, respectively. Like this, at L27-L37, in descending group rank order, each group is individually and uniformly enhanced to the highest possible quality level, such that the quality level is less than or equal to the viewport tile quality level, and L27-37 are repeated until either the bandwidth budget or last direction group is met. At L33-L36, similar to L22-L24 from the Viewport Enhancement phase, in descending affinity score order, tiles in a group are individually enhanced by one quality level, such that the quality level is less than or equal to both the bandwidth budget and viewport tile quality level; and the bandwidth budget at L35 is only adjusted if and only if the individual tile at L34 is enhanced. Lastly, due to the distortion of the projected video, a tile may be classified into two groups. In this case, only the group with the higher rank keeps the tile.

**IV. PERFORMANCE EVALUATION**

In this section, we introduce the evaluation environment, reference methods, user QoE evaluation index, and evaluation results of all tile quality selection algorithms.

**A. EVALUATION ENVIRONMENT**

For all analysis, we used the dataset from [17]. The dataset contains the viewport and head orientation information of 50 users at every frame. The viewports in the dataset are modeled by perfect 100 \times 100 degree circles. We use six videos from the dataset, and a 5 \times 5 equirectangular tile map projection for all videos. Table 1 summarizes the videos used and their specifications in image type, content pace, and the used segment. All video are 60s long and contain 1800 frames. References [26], [27], [28], [29], [30], [31] provide links of each video. These videos were selected for their variability in image type and content pace. However, the criteria for determining the content pace of a video is missing in [17]. Intuitively, fast-paced content refers to highly dynamic content, and slow-paced content refers to less dynamic content. Medium-paced content was not considered. For generating tile affinity scores, we used the viewport information of all users in the dataset except the user under evaluation, meaning that the sample size is 49 users. Lastly, we evaluated the performance of all tile quality selection methods with the 50 users in the dataset.

For the network, we simulated a constant bandwidth environment of 8Mbps and 10Mbps. We used segment sizes of 1s and 2s, wherein each segment comprises of the four quality levels of Q0:1500Kbps, Q1:3000Kbps, Q2:6000Kbps, and Q3:12000Kbps. For our assumptions, we first assume that a segment is not re-requested. Secondly, we assume that all tiles in a segment are at least encoded in the lowest quality level. Thirdly, we assume that all prediction-based methods use the prediction method described in section II. Lastly, we assume that the data size and distortion of all tiles in a segment are fixed. The simplicity of the above network environment is to provide ease in performing direct comparison on the performance of all videos in each user QoE metric and tile quality selection method.

**B. REFERENCE METHODS**

For comparison against the proposed tile quality selection algorithm, we adopt the following eight reference tile quality selection methods. Among these eight, reference methods 2), 3), 4), and 5) are in [25].

1) The Conventional method, from [18], selects the quality of tiles by groups and is the former algorithm to Algorithm 1. Bandwidth is allocated to the groups of viewport, direction, and background, respectively. The conventional method follows Algorithm 1, except L12-L26 and L32-L36. This means that cutoff and individual tile quality enhancement are disregarded.

2) The Baseline method, from [32], presents the user’s current viewport in high quality, with all other tiles in low quality. This method is also called Full Delivery Basic in [32]. Viewport extensions are not used.

3) The State Art 1 method, from [23], selects the quality of tiles by groups. Bandwidth is allocated to the three groups of viewport, adjacent, and outside, respectively. In particular, bandwidth is allocated for the highest possible quality level to the viewport tiles, the remaining bandwidth after enhancing the viewport is allocated for the highest possible quality level to the adjacent tiles, and bandwidth is allocated for the lowest quality level to the outside tiles.

4) The State Art 2 method, from [13], selects the quality of tiles individually based on weights. Bandwidth is allocated to tiles inside the viewport based on their distortion and to tiles outside the viewport based on their distance to the viewport center.

5) The State Art 3 method, from [14], selects the quality of tiles individually based on weights. Bandwidth is allocated to all tiles based on their popularity. Viewport prediction is not used.

6) The Popular 1 method, from [18], selects the quality of tiles individually based on descending tile popularity. Bandwidth is allocated to tiles individually for the highest possible quality level in descending tile affinity score order. Viewport prediction is not used.

7) The Popular 2 method selects the quality of tiles by both groups and individually and is a modified version of the conventional method. This method executes the same as the conventional method, except that viewport extensions are performed by individual tile quality enhancement instead of groups. Bandwidth is allocated.

**TABLE 1. Video content specifications.**

| Video Name       | Image Type | Content Pace | Used Segment |
|------------------|------------|--------------|--------------|
| Roller Coaster   | NI         | fast         | 0:20 - 1:20  |
| Pac-Man          | CG         | fast         | 0:10 - 1:10  |
| Chariot Race     | CG         | fast         | 0:02 - 1:02  |
| Kangaroo Island  | NI         | slow         | 0:01 - 1:01  |
| Perils Panel     | NI         | slow         | 0:10 - 1:10  |
| Shark Shipwreck  | NI         | slow         | 0:30 - 1:30  |
K. Dziubinski, M. Bandai: Viewport History as a Heuristic for Quality Enhancement and Quality Variation Control

FIGURE 5. User QoE performance in perceived quality bitrate, averaged across all 50 users.

| User QoE Evaluation Index |
|---------------------------|
| We evaluated all tile quality selection methods in the perceived quality bitrate and quality switching to measure user QoE. Here, quality switching can be classified as either internal or external quality switching. Respectively, the former is the quality variation inside the viewport during a segment, and the latter is the quality variation inside the viewport between segments. The three evaluation indexes used in this paper are from [4], [7], [8], [9] and are defined next. |

First, the segment perceived quality bitrate $R_i$ is defined as the sum of the quality level bitrate for all tiles, including partial tiles, that overlapped with the user’s viewport $V_i$ during segment $i$. Accordingly, the perceived quality bitrate $R$ is defined as the above segment perceived quality bitrate $R_i$, averaged across all segments.

$$R_i = \sum_{j=1}^{N} b(l_j) | j \in V_i$$  \hspace{1cm} (2)

$$R = \frac{1}{M} \sum_{i=1}^{M} R_i$$  \hspace{1cm} (3)

Second, internal quality switching $I_1$ is defined as the standard deviation of all tile quality levels comprising $V_i$, averaged across all segments.

$$I_1 = \frac{1}{M} \sum_{i=1}^{M} StdDev[b(l_{i,j}) | \forall j \in V_i]$$  \hspace{1cm} (4)

Third and last, external quality switching $I_2$ is defined as the absolute difference between the average tile quality level
of $V_i$ and $V_{i-1}$, averaged across all segment transitions.

$$I_2 = \frac{1}{1 - M} \sum_{i=2}^{M} \left| \frac{R_i}{V_i} - \frac{R_{i-1}}{V_{i-1}} \right|$$

(5)

D. EVALUATION RESULTS
In this section, we introduce the evaluation results on each user QoE index for all tile quality selection methods. All results show the mean and 95% Confidence Interval (CI). For reference, we present these results as only numerical values in the Appendix. We also include the performance results of the proposed method without cutoff to the Appendix.

1) PERCEIVED QUALITY BITRATE
First, the average perceived quality bitrate observed across all 50 users is shown in Figure 5. As shown, the highest perceived quality bitrate possible is unique for each video. The reason is because the number of tiles overlapping a user’s viewport throughout each video is also unique. Regarding the performance, the proposed method generally performed the best compared to the other reference methods in all network environments, except under the network environment of 8Mbps on 2s segments. This is because the bandwidth requirement for requesting the highest quality level could not be achieved for all viewport tiles, even by introducing cutoff. Although lowering the value for cutoff enough here to support the highest quality level will result in a higher perceived quality bitrate, the performance in internal quality switching and external quality switching will both be negatively affected. The Popular 1 method, on the other hand, performed the best under this network environment by enhancing tiles irrespective of quality uniformity.
Notably, the proposed method always outperformed the conventional method. The most notable difference between the proposed method and conventional method is under the network environment of 10Mbps on 2s segments. This significant difference in performance is primarily by the introduction of cutoff. Under the other network environments, the proposed method performed better than the conventional method by the introduction of L32-L36. Secondly, depending on the video, the State Art 3 method and Popular 2 method at times performed better than the proposed method. In particular, the State Art 3 method performed better than the proposed method in Roller Coaster, Pac-Man, Chariot Race, and Kangaroo Island under the network environment of 10Mbps on 2s segments; and the Popular 2 method performed better than the proposed method in Pac-Man under both network environments of 10Mbps on 1s segments and 8Mbps on 2s segments and Chariot Race under the network environment of 8Mbps on 2s segments. This suggests that individual tile quality enhancement based on tile popularity is better for videos centralized in popular content.

Regarding the videos, the performance behavior is consistent. However, a notable difference in performance behavior is observed between fast- and slow-paced content under the network environment of 10Mbps on 2s segments. Compared to fast-paced content, the State Art 3 method and Popular 1 method both performed worse than the FoV method, except in Kangaroo Island, and this is because Kangaroo Island is less dispersed in tile popularity compared to Perils Panel and Shark Shipwreck. From this performance behavior, it can be seen that tile quality selection methods using primarily...
only viewport history are not suitable for all videos, particularly slow-paced content videos. Therefore, based on the above, the perceived quality performance for history-based methods will be affected according to the tile popularity dispersion of a video. Figures 1 and 2 also confirm this by presenting how slow-paced content tended to yield a lower accuracy in viewport prediction and, inversely, a higher dispersion in popular content. This behavior is important to note for 360-degree video content providers on deployment decisions for the appropriate tile quality selection method.

2) INTERNAL QUALITY SWITCHING

Second, the average internal quality switching observed across all 50 users is shown in Figure 6. The State Art 2 method generally performed the best compared to the proposed method and the other reference methods in all network environments, except in Pac-Man. The proposed method performed best in Pac-Man, which is reasoned to be because of the highly centralized popular content. The State Art 2 method yielded a better general performance in internal quality switching than the proposed method by having less quality gaps between tiles in the viewport. However, this is because the State Art 2 method had less high quality tiles perceived by the user, as shown in Figure 5, thus less quality gaps. Though, it is interesting that effective viewport extensions can also be realized by only using the distance to the predicted viewport center as a heuristic for quality level selection.

Following next in general performance is the proposed method. The proposed method performed the second best but did not perform well under the network environment of 10Mbps on 2s segments. This is because of the introduction of cutoff. Higher quality level gaps between tiles are introduced to the viewport as a consequence for reaching a higher perceived quality bitrate by cutoff. With the exception of this network environment, the proposed method outperformed the conventional method by the introduction of L32-L36. In addition, while excluding the above network environment, the proposed method, each time, and the conventional method, at most times, outperformed the Popular 2 method. This suggests that enhancing tiles by groups is better for quality uniformity inside the viewport than individual tiles when using viewport history.

Regarding the other methods, first, the Baseline method and FoV method illustrate the necessity for viewport extensions. The Baseline method, despite retrieving the user’s current viewport, assumes that the user will not change their viewport throughout the segment. However, this is simply not the case, as shown, and re-requesting the segment every time the user’s viewport changes is not practical for the network. The same rational applies to the FoV method, except its performance is further reduced by introducing uncertainty through using viewport prediction. Second, the State Art 1 method yielded the least fruitful performance compared to the other prediction-based methods using viewport extensions by using strict tile groups. This verifies our assertion in section II that using strict tile groups as a viewport extension technique is not advisable. Lastly, the State Art 3 method and Popular 1 method both yielded the worst performance compared to the prediction-based methods by not considering the actual user’s viewport information.

Regarding the videos, a notable difference in performance is observed. The proposed method performed the second best but did not perform well under the network environment of 10Mbps on 2s segments. This is because of the introduction of cutoff. Higher quality level gaps between tiles are introduced to the viewport as a consequence for reaching a higher perceived quality bitrate by cutoff. With the exception of this network environment, the proposed method outperformed the conventional method by the introduction of L32-L36. In addition, while excluding the above network environment, the proposed method, each time, and the conventional method, at most times, outperformed the Popular 2 method. This suggests that enhancing tiles by groups is better for quality uniformity inside the viewport than individual tiles when using viewport history.

|  | 5 Mbps | 10 Mbps |
|---|---|---|
| Proposed | 4029.36 | 4152.57 |
| Conventional | 4250.46 | 4320.74 |
| Baseline | 3960.60 | 3960.60 |
| State Art 1 | 3948.40 | 4080.16 |
| State Art 2 | 4219.66 | 4278.14 |
| Popular 1 | 3679.28 | 3778.62 |
| Popular 2 | 4248.56 | 4276.46 |
| FoV | 3709.86 | 3709.86 |

TABLE 3. Pac-Man QoE performance in perceived quality bitrate, averaged across all 50 users.

|  | 5 Mbps | 10 Mbps |
|---|---|---|
| Proposed | 3257.52 | 3180.12 |
| Conventional | 2249.28 | 3091.76 |
| Baseline | 2065.60 | 3030.72 |
| State Art 1 | 2024.64 | 2939.04 |
| State Art 2 | 2214.24 | 3058.92 |
| State Art 3 | 2301.60 | 3383.04 |
| Popular 1 | 2816.32 | 3559.20 |
| Popular 2 | 2722.56 | 3129.56 |
| FoV | 1944.40 | 2845.04 |

TABLE 2. Roller Coaster QoE performance in perceived quality bitrate, averaged across all 50 users.
TABLE 4. Chariot Race QoE performance in perceived quality bitrate, averaged across all 50 users.

| 8 Mbps Mean | 10 Mbps Mean | 95% CI | 95% CI |
|-------------|--------------|--------|--------|
| Proposed 4050.20 | 4110.18 | [3986.79, 4113.66] | [4042.59, 4177.65] |
| Conventional 4029.00 | 4103.32 | [3966.08, 4091.91] | [4037.97, 4172.66] |
| Baseline 3688.98 | 3688.98 | [3636.49, 3741.46] | [3636.49, 3741.46] |
| State Art 1 3705.60 | 3890.16 | [3726.67, 3814.45] | [3835.37, 3944.94] |
| State Art 2 3971.52 | 4048.22 | [3907.69, 4035.24] | [3977.92, 4118.51] |
| State Art 3 3487.44 | 3561.30 | [3414.67, 3560.20] | [3485.10, 3633.49] |
| Popular 1 3888.90 | 4065.54 | [3820.83, 3956.96] | [3996.00, 4133.07] |
| Popular 2 4023.24 | 4110.14 | [3959.27, 4087.20] | [4041.04, 4179.18] |
| FoV 3640.50 | 3640.50 | [3413.67, 3507.32] | [3413.67, 3507.32] |

TABLE 5. Kangaroo Island QoE performance in perceived quality bitrate, averaged across all 50 users.

| 8 Mbps Mean | 10 Mbps Mean | 95% CI | 95% CI |
|-------------|--------------|--------|--------|
| Proposed 3177.32 | 3203.24 | [3135.75, 3231.88] | [3161.34, 3270.53] |
| Conventional - | - | [3275.36, 3377.74] | [3275.36, 3377.74] |
| Baseline 2150.64 | 2371.36 | [2110.95, 2190.32] | [2324.28, 3347.41] |
| State Art 1 1938.44 | 2184.04 | [1909.81, 1967.06] | [2142.47, 2275.60] |
| State Art 2 1922.00 | 2135.68 | [1894.67, 1949.32] | [1907.27, 2164.08] |
| State Art 3 2100.56 | 2311.00 | [2063.80, 2217.31] | [2163.82, 2358.17] |
| State Art 4 2359.88 | 2524.02 | [2312.87, 2407.08] | [2348.38, 2321.24] |
| Popular 1 2786.44 | 2947.64 | [2727.31, 2845.56] | [2841.50, 3573.77] |
| Popular 2 2718.12 | 2937.32 | [2213.23, 2220.96] | [2286.44, 3388.19] |
| FoV 1822.88 | 2032.34 | [1799.38, 1846.37] | [1974.16, 3074.47] |

TABLE 6. Perils Panel QoE performance in perceived quality bitrate, averaged across all 50 users.

| 8 Mbps Mean | 10 Mbps Mean | 95% CI | 95% CI |
|-------------|--------------|--------|--------|
| Proposed 2197.64 | 2201.99 | [2136.30, 2233.37] | [2136.30, 2233.37] |
| Conventional - | - | [2078.73, 2023.88] | [2078.73, 2023.88] |
| Baseline 2169.72 | 2334.85 | [2136.15, 2203.28] | [2328.37, 3347.09] |
| State Art 1 1973.92 | 2064.80 | [1941.27, 2006.56] | [2034.91, 3144.68] |
| State Art 2 1955.52 | 2020.24 | [1925.65, 1985.38] | [1936.77, 2348.10] |
| State Art 3 2135.64 | 2328.76 | [2100.10, 2171.87] | [2136.35, 3239.16] |
| State Art 4 1893.12 | 2554.84 | [1852.83, 1933.40] | [2284.51, 2621.16] |
| Popular 1 2584.36 | 2897.16 | [2468.51, 2877.53] | [2420.22, 2992.95] |
| Popular 2 2175.92 | 2340.34 | [2123.81, 2192.02] | [2305.99, 3390.48] |
| FoV 1835.20 | 2094.40 | [1805.84, 1864.55] | [2034.73, 3146.06] |

TABLE 7. Shark Shipwreck QoE performance in perceived quality bitrate, averaged across all 50 users.

| 8 Mbps Mean | 10 Mbps Mean | 95% CI | 95% CI |
|-------------|--------------|--------|--------|
| Proposed 4107.54 | 4073.54 | [4104.83, 4104.24] | [4140.62, 4140.58] |
| Conventional 4170.50 | 4439.74 | [4113.73, 4272.56] | [4179.96, 4359.47] |
| Baseline 3950.80 | 3950.80 | [3889.98, 4011.61] | [3889.98, 4011.61] |
| State Art 1 3800.94 | 3910.84 | [3738.78, 3863.19] | [3798.59, 4104.72] |
| State Art 2 4182.18 | 4164.46 | [4041.44, 4249.89] | [4128.56, 4271.99] |
| State Art 3 3199.96 | 3958.66 | [3134.31, 3265.60] | [3358.89, 3658.42] |
| State Art 4 3517.76 | 4014.14 | [3441.75, 3593.76] | [3592.36, 4101.91] |
| Popular 1 4097.65 | 2896.40 | [4041.65, 4153.62] | [4242.56, 4507.53] |
| Popular 2 3550.54 | 3550.54 | [3470.78, 3639.29] | [3470.78, 3639.29] |

3) EXTERNAL QUALITY SWITCHING

Third and last, the average external quality switching observed across all 50 users is shown in Figure 7. Here, the best performing tile quality selection method varied according to the video and network environment. In summary, the following four methods performed the best: the proposed method, the State Art 3 method, the Popular 1 method, and the Popular 2 method. Interestingly, these methods share the common point of using viewport history. In addition, the State Art 3 method tended to perform the best in slow-paced content, and the Popular 2 method tended to perform the best in fast-paced content. The Popular 1 method performed the best only in Roller Coaster and Pac-Man. The above suggests that viewport history can provide consistency in quality across segments, especially for videos condensed in tile popularity, and that there were more severe instances of viewport mispredictions in slow-paced content. Viewport mispredictions can result in large quality level changes to occur between segments, but strictly history-based methods can provide stability in performance.

Overall, the proposed method tended to perform the best on 1s segments. In addition, the proposed method notably outperformed the conventional method. The most notable difference between the proposed method and conventional method is under the network environment of 10Mbps on 2s segments. This significant difference in performance is primarily by introducing cutoff to address viewport tile count fluctuations. Under the other network environments, the proposed method degraded the performance of all history-based methods. This behavior is important to note for 360-degree video content providers on deployment decisions for the appropriate viewport estimation method.

4) VOLUME 10, 2022 94663
performed better than the conventional method by introducing L32-L36.

Regarding the videos, similar to internal quality switching, the same performance behavior is observed between fast- and slow-paced content, and it is also observed under all network environment conditions. Likewise, the degree of external quality switching tended to be higher in slow-paced content.

### V. DISCUSSION

First, the dispersion of popular content tended to influence the performance of the tile quality selection methods. In particular, slow-paced video content tended to yield the worst performance in user QoE. This is because the dispersion of popular content is a reflection of multiple users’ viewing behavior. When the popular content is highly dispersed, such as in Shark Shipwreck, the likelihood for the user to significantly change their viewport is higher, which can decrease the performance of prediction-based methods and lower the tile-likelihood for history-based methods. Conversely, when the popular content is not as dispersed, such as in Roller Coaster, the performance is generally better for both prediction-based and history-based methods. The above relation between content pace and user head movements should be further investigated.

Second and last, the proposed method is in a tradeoff with the conventional method for both internal and external quality switching. As exhibited on 2s segments, both methods yield a different performance in internal and external quality switching because of the introduction of cut-off in the proposed method. Though introducing cut-off can suppress external quality switching and increase the perceived quality bitrate, internal quality switching is higher because the quality level difference between tiles that were not as enhanced is also higher. Depending on the user’s preferred

### TABLE 8. Roller Coaster QoE performance in internal quality switching, averaged across all 50 users.

| 1s  | 8 Mbps | Mean 95% CI | 10 Mbps | Mean 95% CI |
|-----|--------|-------------|--------|-------------|
| Proposed | 21.7810 | [17.1650, 26.3969] | 11.7520 | [8.4943, 15.0096] |
| Conventional | 23.9610 | [19.1692, 28.7527] | 12.8810 | [9.5261, 16.2358] |
| Baseline | 78.3130 | [70.2904, 86.3355] | 78.3130 | [70.2904, 86.3355] |
| State 1 | 63.2900 | [55.4900, 71.0899] | 51.0290 | [42.6287, 57.7722] |
| State 2 | 15.3600 | [10.8509, 19.8664] | 7.7400 | [5.1065, 10.3843] |
| State 3 | 90.3990 | [81.0927, 99.7052] | 80.1540 | [70.6644, 89.6615] |
| Popular 1 | 43.4220 | [33.9618, 52.8821] | 24.3900 | [17.1139, 31.5840] |
| Popular 2 | 25.2840 | [20.1804, 31.2755] | 13.4500 | [9.8640, 17.0332] |
| FoV 2 | 95.2520 | [86.1905, 104.313] | 95.2520 | [86.1905, 104.313] |

### TABLE 9. Pac-Man QoE performance in internal quality switching, averaged across all 50 users.

| 1s  | 8 Mbps | Mean 95% CI | 10 Mbps | Mean 95% CI |
|-----|--------|-------------|--------|-------------|
| Proposed | 20.3530 | [16.4302, 23.7799] | 13.7464 | [11.0362, 16.4529] |
| Conventional | 23.3720 | [19.5265, 27.2174] | 74.5570 | [67.9590, 81.1549] |
| Baseline | 49.2190 | [41.9139, 56.5240] | 93.4750 | [85.2633, 101.731] |
| State 1 | 42.9210 | [39.0750, 46.7666] | 88.0670 | [80.7660, 95.4270] |
| State 2 | 19.1590 | [15.3960, 22.9219] | 72.9010 | [65.7511, 80.0508] |
| State 3 | 102.4650 | [99.5294, 105.322] | 142.1680 | [137.359, 146.736] |
| Popular 1 | 177.6920 | [174.393, 180.990] | 155.297 | [149.509, 161.084] |
| Popular 2 | 20.3370 | [16.6170, 24.0659] | 74.3040 | [67.317, 81.3384] |
| FoV 2 | 52.7160 | [48.2505, 57.1814] | 98.1940 | [90.1204, 106.267] |

### TABLE 10. Chariot Race QoE performance in internal quality switching, averaged across all 50 users.

| 1s  | 8 Mbps | Mean 95% CI | 10 Mbps | Mean 95% CI |
|-----|--------|-------------|--------|-------------|
| Proposed | 21.9990 | [19.5925, 24.4647] | 123.469 | [115.548, 131.389] |
| Conventional | 24.8810 | [22.2166, 27.5453] | 90.3470 | [82.5574, 98.1365] |
| Baseline | 51.7870 | [47.6208, 55.9351] | 106.344 | [97.7037, 114.984] |
| State 1 | 42.9350 | [39.8297, 46.0762] | 103.268 | [95.2465, 111.289] |
| State 2 | 21.2350 | [18.0026, 24.4673] | 93.8680 | [84.9497, 102.786] |
| State 3 | 118.3170 | [115.765, 120.868] | 141.650 | [136.242, 147.057] |
| Popular 1 | 174.3660 | [171.541, 177.190] | 147.399 | [140.547, 154.250] |
| Popular 2 | 23.1430 | [20.3529, 25.9310] | 90.4410 | [82.8323, 98.4996] |
| FoV 2 | 54.5120 | [50.6013, 58.4225] | 113.222 | [104.622, 121.821] |

### TABLE 11. Kangaroo Island QoE performance in internal quality switching, averaged across all 50 users.

| 1s  | 8 Mbps | Mean 95% CI | 10 Mbps | Mean 95% CI |
|-----|--------|-------------|--------|-------------|
| Proposed | 26.9150 | [22.9540, 30.8599] | 15.4430 | [12.2120, 18.6739] |
| Conventional | 29.8650 | [25.7889, 33.9410] | 17.1710 | [13.8136, 20.5263] |
| Baseline | 87.5480 | [81.2391, 93.8609] | 87.5480 | [81.2391, 93.8609] |
| State 1 | 50.6540 | [45.5029, 55.8050] | 40.8260 | [36.6282, 45.0237] |
| State 2 | 26.6380 | [21.7950, 31.4809] | 14.0870 | [9.6865, 18.4872] |
| State 3 | 97.9120 | [91.3906, 104.433] | 87.5990 | [81.0412, 94.1387] |
| Popular 1 | 72.5320 | [65.5423, 79.5036] | 31.4410 | [26.7855, 36.0964] |
| Popular 2 | 41.5660 | [37.5669, 45.8650] | 19.4130 | [16.7141, 22.1118] |
| FoV 2 | 101.1450 | [95.9329, 108.360] | 101.145 | [95.9329, 108.360] |

94664 VOLUME 10, 2022
TABLE 12. Perils Panel QoE performance in internal quality switching, averaged across all 50 users.

| 1s | 8 Mbps | 10 Mbps |
|----|--------|---------|
| Mean | 5% CI | Mean | 5% CI |
| Proposed | 37.9330 (31.0858, 42.7601) | 17.5250 (14.0332, 21.0167) |
| Conventional | 40.4080 (35.6331, 45.1828) | 19.6640 (16.0226, 23.2693) |
| Baseline | 91.4840 (85.6213, 97.3466) | 91.4840 (85.6213, 97.3466) |
| State Art 1 | 71.3610 (65.8140, 76.9079) | 58.0960 (53.2871, 62.9048) |
| State Art 2 | 26.9700 (22.8572, 31.0827) | 62.3340 (53.0761, 71.6946) |
| State Art 3 | 114.105 (108.282, 119.927) | 96.9140 (92.8335, 101.004) |
| Popular 1 | 112.919 (103.606, 122.231) | 47.0900 (39.6327, 54.4452) |
| Popular 2 | 62.2790 (55.4844, 68.7372) | 29.6100 (23.5633, 35.9196) |
| FoV | 109.122 (102.460, 115.783) | 109.122 (102.460, 115.783) |

TABLE 13. Shark Shipwreck QoE performance in internal quality switching, averaged across all 50 users.

| 1s | 8 Mbps | 10 Mbps |
|----|--------|---------|
| Mean | 5% CI | Mean | 5% CI |
| Proposed | 33.6970 (30.5455, 36.8449) | 138.425 (130.690, 146.159) |
| Conventional | 103.196 (96.3111, 110.080) | 100.731 (97.9034, 113.511) |
| Baseline | 56.5620 (53.1915, 59.9288) | 117.760 (110.558, 125.964) |
| State Art 1 | 47.1190 (44.4159, 49.8220) | 112.994 (106.183, 119.804) |
| State Art 2 | 29.9170 (26.9735, 32.8604) | 104.695 (97.4219, 111.964) |
| State Art 3 | 75.8040 (73.6946, 77.9133) | 111.140 (107.767, 114.512) |
| Popular 1 | 162.372 (157.581, 167.162) | 167.514 (162.505, 172.522) |
| Popular 2 | 42.1560 (38.6877, 45.6242) | 110.429 (103.139, 117.718) |
| FoV | 58.9970 (55.5803, 62.3396) | 122.152 (114.773, 129.530) |

TABLE 14. Roller Coaster QoE performance in external quality switching, averaged across all 50 users.

| 1s | 8 Mbps | 10 Mbps |
|----|--------|---------|
| Mean | 5% CI | Mean | 5% CI |
| Proposed | 12.5410 (10.1833, 14.9006) | 46.5040 (41.3857, 51.6222) |
| Conventional | 12.4400 (10.1833, 14.9006) | 77.2400 (72.1715, 82.3081) |
| Baseline | 14.0580 (11.7381, 16.3778) | 76.2670 (71.2167, 81.3172) |
| State Art 1 | 25.1550 (23.0400, 27.2699) | 85.2190 (79.8940, 90.4539) |
| State Art 2 | 26.9830 (25.1543, 28.8120) | 85.3310 (80.2417, 90.4202) |
| State Art 3 | 17.4660 (15.2749, 19.6579) | 84.1490 (78.7038, 89.5941) |
| Popular 1 | 45.3400 (43.9360, 47.0893) | 39.7310 (36.5514, 42.9105) |
| Popular 2 | 32.5760 (30.2648, 35.4051) | 37.1870 (35.2825, 39.1435) |
| FoV | 31.5590 (29.2429, 33.6930) | 92.0790 (86.8564, 97.2925) |

TABLE 15. Pac-Man QoE performance in external quality switching, averaged across all 50 users.

| 1s | 8 Mbps | 10 Mbps |
|----|--------|---------|
| Mean | 5% CI | Mean | 5% CI |
| Proposed | 7.75600 (5.82341, 9.68585) | 3.63600 (2.95495, 4.37105) |
| Conventional | 9.39600 (7.26163, 11.5303) | 4.42400 (3.09163, 5.79365) |
| Baseline | 53.9000 (48.4214, 57.9711) | 53.0000 (48.4214, 57.9711) |
| State Art 1 | 19.4760 (16.6186, 22.9239) | 17.3520 (14.5499, 19.9740) |
| State Art 2 | 19.4650 (16.1842, 22.9457) | 11.8050 (10.0788, 13.5311) |
| State Art 3 | 25.8340 (22.6262, 29.0405) | 21.8550 (18.1326, 25.7733) |
| Popular 1 | 10.2440 (7.6843, 13.1636) | 4.22800 (3.97208, 4.56591) |
| Popular 2 | 8.45000 (6.28001, 10.6899) | 3.50700 (2.41949, 4.59450) |
| FoV | 62.0300 (57.0066, 67.0623) | 62.0300 (57.0066, 67.0623) |

VI. CONCLUSION AND FUTURE WORK

In this paper, to resolve the challenges concerning the bandwidth requirement and user QoE, we proposed a novel viewport prediction-based and viewport history-based tile quality selection algorithm. The proposed algorithm uses viewport prediction to maintain the user’s individualized viewing experience and past users’ viewport history to address viewport mispredictions and quality variation. From the performance evaluation, we demonstrated that using users’ viewport history as a heuristic for quality level selection can further increase user QoE. In summary, the proposed method performed best at increasing the perceived bitrate quality while minimizing both internal and external quality switching on 1s segments. In addition, by the newly added algorithmic logic, the proposed method performed better than the conventional algorithm in all QoE metrics on 1s segments.

For our future work, we have a list of items that we will like to resolve for this future work. First, we will like to investigate in detailing a tile quality selection algorithm that can realize different viewing experiences, such as prioritizing either internal or external quality switching. Second, we will like to investigate on how to remove the empirical analysis for determining cutoff by formalizing an equation. Third, we will like to investigate deeper on the relation between content pace and user head movements. Fourth, we will like to perform analysis on using advanced viewport estimation methods. Here, this is because we used a simple linear regression model as our viewport estimation method, but advanced viewport estimation methods, such as the neural network-based method in [19], could

viewing experience, the user may or may not want to use cutoff as a control means for quality control, but we would like to reserve this for our future work.
have different impacts on the performance of the tile quality selection methods shown in section IV. Fifth and last, we like to perform analysis on using different equirectangular tiling schemes. This is because the proposed algorithm is specifically tailored for a 5 × 5 equirectangular tile map, and therefore modifications to the present algorithm will likely be needed for optimal performance according to the tiling scheme. For example, in a large tiling scheme, multiple layers of viewport extensions and corner tiles should be considered for quality enhancement. In addition, the tiling scheme can impact the performance of the tile quality selection method [25]. In conclusion, we expect the findings from this study to be of use to research communities and industries on deployment decisions for 360-degree video streaming solutions.

## REFERENCES

[1] P. Cipresso, I. A. C. Giglioli, M. A. Raya, and G. Riva, “The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature,” Frontiers Psychol., vol. 9, p. 2086, Nov. 2018.

[2] O. R. L. Farley, K. Spencer, and L. Baudinet, “Virtual reality in sports coaching, skill acquisition and application to surfing: A review,” J. Hum. Sport Exerc., vol. 15, no. 3, pp. 535–548, 2019.

[3] M. Zink, R. Sitaraman, and K. Nahrstedt, “Scalable 360° video stream delivery: Challenges, solutions, and opportunities,” Proc. IEEE, vol. 107, no. 4, pp. 639–650, Feb. 2019.

[4] 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 Netw.), May 2019, pp. 1–9.

[5] C. Perfecto, M. S. Elbamby, J. D. Ser, and M. Bennis, “Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework,” IEEE Trans. Commun., vol. 68, no. 4, pp. 2491–2508, Apr. 2020.
[6] M. Deveci, (May 2021). Measuring Head-Mounted Display’s (HMD) Motion-to-Photon (MTP) Latency. [Online]. Available: https://www.optofidelity.com/blog/measuring-head-mounted-displays-hmd-motion-to-photon-mtp-latency

[7] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, “Flare: Practical viewport-adaptive 360-Degree video streaming for mobile devices,” in Proc. Ann. Int. Conf. Mobile Comput. Netw., New Delhi, India, Nov. 2018, pp. 99–114.

[8] A. Sekine and M. Bandai, “Suppression in quality variation for 360-degree tile-based video streaming,” IEICE Trans. Commun., vol. E104.B, no. 6, pp. 616–623, Jun. 2021.

[9] A. Sekine and M. Bandai, “Tile quality selection method in 360-degree tile-based video streaming,” in Proc. Int. Workshop Telecommun. Netw., Appl. Syst., Apr. 2020, pp. 535–544.

[10] A. T. Nasrabadhi, A. Mahzari, J. D. Beshay, and R. Prakash, “Adaptive 360-degree video streaming using scalable video coding,” in Proc. ACM Multimedia Conf., Oct. 2017, pp. 1689–1697.

[11] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, “Optimizing 360° video delivery over cellular networks,” in Proc. 5th Workshop All Things Cellular, Oct. 2016, pp. 1–6.

[12] 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.

[13] C. Ozcinar, A. D. 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.

[14] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, “Viewport-driven rate-distortion optimized 360° video streaming,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2018, pp. 1–7.

[15] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, “Viewport-adaptive navigable 360-degree video delivery,” in Proc. IEEE Int. Conf. Commun., May 2017, pp. 1–7.

[16] 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.

[17] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu, “360° video viewing dataset in head-mounted virtual reality,” in Proc. ACM Multimedia Syst. Conf. (MMSys), Jun. 2017, pp. 211–216.

[18] K. Dziubinski and M. Bandai, “Advancing user quality of experience using viewport archives in viewport-aware tile-based 360-degree video streaming,” in Proc. IEEE Int. Workshop Tech. Committee Commun. Qual. Rel. (CQR), May 2021, pp. 32–37.

[19] C. Fan, J. Lee, W. Lo, C. Huang, K. Chen, and C. Hsu, “Fixation prediction for 360° video streaming in head-mounted virtual reality,” in Proc. 27th ACM Workshop Netw. Operating Syst. Support Digit. Audio Video, Jun. 2017, pp. 67–72.

[20] X. Corbillon, F. De Simone, and G. Simon, “360-degree video head movement dataset,” in Proc. ACM Multimedia Syst. Conf., Jun. 2017, pp. 199–204.

[21] C. Koch, A.-T. Rak, M. Zink, R. Steinmetz, and A. Rizk, “Transitions of viewport quality adaptation mechanisms in 360 degree video streaming,” in Proc. 29th ACM Workshop Netw. Operating Syst. Support Digit. Audio Video, 2019, pp. 14–19.

[22] D. V. Nguyen, H. T. T. Tran, A. T. Pham, and T. C. Thang, “An optimal tile-based approach for viewport-adaptive 360-degree video streaming,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 9, no. 1, pp. 29–42, Mar. 2019.

[23] 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.

[24] T. Ogasawara and M. Bandai, “Buffer based adaptation using scalable video coding for 360-degree video streaming over NDN,” in Proc. Int. Conf. Inf. Netw. (ICOIN), Jan. 2020, pp. 142–145.

[25] D. V. Nguyen, H. T. T. Tran, and T. C. Thang, “An evaluation of tile selection methods for viewport-adaptive streaming of 360-degree video,” ACM Trans. Multimedia Comput., Commun., Appl., vol. 16, no. 1, pp. 1–24, Feb. 2020.

[26] Roller Coaster. Accessed: Jan. 2022. [Online]. Available: https://youtu.be/8lsB-P8nGSM

[27] Pac-Man. Accessed: Jan. 2022. [Online]. Available: https://youtu.be/p9h3Zj4lA1A

[28] Chariot Race. Accessed: Jan. 2022. [Online]. Available: https://youtu.be/fMyDqZez7M

[29] Perils Panel. Accessed: Jan. 2022. [Online]. Available: https://youtu.be/kjIP5vWqPryY

[30] Kangaroo Island. Accessed: Jan. 2022. [Online]. Available: https://youtu.be/MXIHitXtcN6

[31] Shark Shipwreck. Accessed: Jan. 2022. [Online]. Available: https://youtu.be/aQd41nbQM-U

[32] M. Graf, C. Timmerer, and C. Mueller, “Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation,” in Proc. ACM Multimedia Syst. Conf. (MMSys), 2017, pp. 261–271.

KIANA DZIUBINSKI received the B.S. degree in engineering and applied sciences and the M.S. degree in green science and engineering from Sophia University, Japan, in 2019 and 2021, respectively, where she is currently pursuing the Doctoral degree with the Department of Information and Communication Sciences.

Her current research interests include VR, IoT, and smart home.

MASAKI BANDAI (Member, IEEE) received the B.E. and M.E. degrees in electrical engineering and the Ph.D. degree in information and computer science from Keio University, Japan, in 1996, 1998, and 2004, respectively.

From 2004 to 2010, he was an Assistant Professor at Shizuoka University, Japan. He is currently a Professor with the Department of Information and Communication Sciences, Sophia University, Japan. His current research interests include computer networks, network computing, and applications.