User Gaze-Driven Adaptation of Omnidirectional Video Delivery Using Spatial Tiling and Scalable Video Encoding

Adam Polakovič, Gregor Rozinaj, and Gabriel-Miro Muntean, *Senior Member, IEEE*

***Abstract*—Omnidirectional video is becoming increasingly popular among viewers, but its delivery requires considerable amount of network bandwidth. Today's streaming services are transmitting the full spatial angle of omnidirectional videos, although most of the transmitted content is not utilised. Due to both limited bandwidth availability and its dynamic fluctuations, adaptive delivery solutions play a key role in supporting high user quality streaming of omnidirectional videos. This paper describes research which extends the MPEG-DASH Spatial Relationship Description by adding scalable video encoding to spatial tiling. It proposes a novel tile-layering based gaze adaptation algorithm for omnidirectional video delivery and employs it in conjunction with multiple tiling schemes. The benefits of the proposed algorithm with diverse tiling schemes are evaluated objectively in terms of bandwidth savings and adaptation latency. The results show a reduction of network bandwidth requirements to about 30 % of the original bandwidth value with a low processing latency of 70.87 ms.**

***Index Terms*—Omnidirectional video, tiled streaming, adaptive video delivery, MPEG-DASH SRD, scalable video encoding.**

I. Introduction

THE popularity of 360-degree videos is growing fueled especially by the immersive experience offered by virtual reality (VR) display systems such as head-mounted displays (HMD). One of the commonly available HMDs is HTC Vive with a resolution of 1080×1200 pixels for one eye, covering nominal field of view of about 110° and performing updates at 90 Hz [1]. Another commonly used HMD is Oculus Rift with the resolution of 1080×1200 pixels for one eye, covers horizontal field of view of about 94° with the same refresh rate as HTC Vive [3]. To prevent simulator sickness and support good user Quality of Experience (QoE), the overall display system should react in around 10 ms [2]. Stereo omnidirectional videos are usually encoded with resolutions from 3840×3840 pixels to 7680×7680 pixels and frame rates of 30 - 60 fps. These video characteristics are associated with large amounts of video data in the order of 150 - 360 Mbit/s [4]. Current VR video delivery systems are transmitting full 360-degree videos, but only a portion of each frame is utilised within a user viewport. With the increasing demand for 360-degree stereoscopic videos, the bandwidth requirements for delivery of such content providers increases, putting pressure on existing limited capacity infrastructure and potentially affecting user QoE. There are several strategies which try to address this problem and mitigate the negative impact on QoE.

Employing scalable video encoding (SVC) is a compression-driven approach and is based on the SVC principle introduced in various forms by international video standards e.g. MPEG-4 Visual [5], H.264 SVC [6], H.265 HEVC [7]. A conventional video bitstream is divided into a base layer and multiple enhancement layers which improve the base layer. In general sender-based adaptive algorithms use SVC to adjust some video encoding characteristics during video delivery to match the available network bandwidth and reduce loss, increasing user QoE [8], [9]. Such approaches allow for effective video transmission via networks with limited and/or highly variable bandwidth, but rely on client feedback which sometimes introduces delays in the adaptation process.

Client-based approaches rely on the MPEG Dynamic Adaptive Streaming over HTTP (DASH) standardized by ISO/EIC [10]. DASH divides the bitstream into temporal segments containing encoded media data and metadata using the Media Presentation Description (MPD) format. The video has multiple representations, encoded at different bit rates (e.g. using different resolution, encoding quality and/or framerate). Representations are sliced into temporal segments with given duration. Each segment has unique URL accessible via HTTP GET requests [11]. MPD contains resources identifiers (URLs) to access the video segments. DASH-based adaptive schemes enable clients to select most appropriate segments during video playout for delivery based on existing network conditions in order to improve video quality [12]. Adaptive video streaming can be improved in order to achieve some goals, such as for instance by exploiting the statistical equivalence of QoE for saving bandwidth [13]. DASH Spatial Relationship Description (SRD) [14] was introduced to enable streaming of sub-parts of a video with a combination of multiple bitrates supported by DASH. This extension enables the DASH client to download and display only relevant parts of the video with appropriate representations and bitrates to achieve high QoE. SRD enables a video to be spatially partitioned into tiles, which are independently decodable videos. Tile-based adaptive solutions such as zoomable and navigable video [15] are effective, but challenging. SRD-based approaches have some drawbacks. Temporal synchronization of multiple tiles must be ensured as more than one tile might be needed for reconstructing video user's viewport and encoding efficiency deteriorates as similarities between tiles cannot be utilised during compression [16].

This paper presents an innovative method for omnidirectional video delivery adaptation using spatial tiling and scalable video encoding. This method extends the idea of

MPEG DASH SRD and improves adaptability of omnidirectional video delivery by layering spatial video tiles. The paper proposes **a novel Tile-layering based gaze adaptation algorithm (TLGA)** which adjusts buffering of video tiles at appropriate scalable video encoding layers based on user's gaze position. It prioritizes the base layer of the currently playback segments, giving more importance to the tiles close to user's gaze. The paper presents an evaluation of the proposed algorithm for multiple spatial tiling schemes for on-demand client-server transmission of omnidirectional video. Testing results demonstrate high network bandwidth savings and low adaptation latency.

The paper makes the following contributions:

- An innovative method for dividing an omnidirectional video into tiles, i.e. square (based on longitude and latitude), or progressive (with comparable covered area) and organising these tiles into layers of quality.
- The Tile Layering-based Gaze Adaptation algorithm (TLGA) which prioritizes the delivered tile quality based on their position from user's gaze,
- A comparative evaluation of tiling schemes in terms of adaptation latency and bandwidth saving when subjected to changes in user gaze.

An overview of the proposed algorithm is provided in Section III and is detailed in Section IV. Experimental results are presented and discussed in Section V. In Section VI conclusions are drawn.

II. Related Works

Research community has put a lot of effort over the years trying to enable high quality multimedia delivery given various delivery network constraints, mostly in terms of bandwidth and latency. Proposing adaptive solutions which adjust the multimedia content to accommodate current network delivery conditions was a successful approach. Adaptive multimedia transmissions help support the increasing user expectation for improved quality of experience by dealing with content, device and network-related aspects. Diverse adaptive solutions were proposed to consider visual quality [8], navigation [17], Region-of-Interest (RoI) [18], energy consumption [19], load balancing [20], etc. on mobile and fixed networks. These solutions use a wide range of approaches from classic optimisation techniques [21] to innovative methods such as machine learning [22]. More recently many efforts were put to design adaptive streaming solutions which target non-traditional content, including 4K/8K, 360° [23] and multisensorial [24] and innovative settings, including multi-device [12] and multi-stream [25]. More adaptation research proposals, including some based on the MPEG-DASH standard, were discussed in terms of their benefits and limitations in [26] and [27].

The authors of [28] have proposed modifications of MPEG-DASH to create multiple representations of the same omnidirectional video. In addition to creating representations with multiple bitrates, representations will offer enhanced quality in some quality emphasis regions (QER). The quality deteriorates the further the area is from the center of this region and similarly to MPEG-DASH, videos are split into temporal segments. When a user moves suddenly, the video playback is not stopped, but the video is presented with a lower quality in the user's viewport.

One of the implementations of MPEG-DASH SRD used tiling scheme dividing omnidirectional video encoded with equirectangular projection into 6 tiles (top and bottom cap of a sphere and four sides) [29]. Authors of [30] extended the idea of MPEG-DASH SRD and proposed using a divide and conquer algorithm to prioritize download of tiles in user's field of view. Authors of [31] have also used a cubic projection, however they have divided each of the 6 tiles into 4 tiles, creating 24 HEVC encoded tiles with original and low resolution. Another modification of traditional DASH was presented in [32], this contribution makes use of a software-defined networking (SDN) architecture for streaming VR multimedia and optimizing the bitrate for viewer's region of interest.

Another approach is represented by foveated streaming solutions, which make use of the uneven distribution of photoreceptor cells in human eyes. Higher quality of video is projected onto the center of retina called *fovea centralis* and lower quality is presented in the eccentric parts of the retina. In foveated rendering it was shown that there is no observable difference in full rendering and foveated rendering if the adaptation to user's gaze is prompt [33], [34]. Authors of [35] have proposed a system for foveated video streaming. The system generates regions with different sizes of enhanced area and merges them into one frame. The frames are then encoded with H.264 and sent to the user. A blurring mask is applied on transitions between merged layers to fade the edges. The authors have shown 5-8 times decrease in bandwidth consumption and a double performance increase. A different approach to foveated streaming applied in cloud gaming was proposed with setting different quantization parameters to macroblocks depending on the real time gaze fixation. The authors have shown total of 110 ms end-to-end latency of their system [36]. On the same principle as MPEG-DASH, the authors of [37] have proposed a method, where client sends requests to the server with spherical coordinates and size of region of interest. According to this request, the server crops high-resolution segment with delay of 700 ms. The client also downloads low-resolution segments and merges high- and low-resolution segments together, where part of low-resolution segment is omitted in favor of high-resolution segment.

Techniques described above have limited possibilities of adaptation to changes in user's gaze, and can only work well with little to no movements of the user's gaze, which is often not the case in real-world conditions. When the gaze suddenly changes, the limitations of the above mentioned adaptation methods are demonstrated. They either do not fully utilise the already downloaded content and have to re-download new one, or even have to re-encode the content on server side, which can take considerable amount of computing power in case of multiple users using one server.

The proposed TLGA algorithm for omnidirectional video delivery is fully utilising all the transmitted data, and also offers fast adaptation to user's gaze downloading only differences between already buffered content and required quality. TLGA also allows for scalability in number of streaming users,

because the content is available encoded on the server and does not need to be re-encoded for each user.

Authors of [38] proposed a method of streaming layered tiles with a shared coded picture, which is in fact a base layer covering the full video. This approach helps exploit spatial similarities between tiles and saves 11% to 14% in comparison to regular tile-based streaming. Similarly, authors of [39] considered streaming a full 360-degree base layer and downloading enhancement layers and researchers in [40] proposed an approach based on layered video coding for 360-degree video which considers a spatial dimension in addition to the classic temporal one in the adaptation process. Unfortunately none of these approaches is compatible with MPEG-DASH, which makes is difficult in terms of adoption. Additionally, with fast enough adaptation to changes in user gaze, it might not be needed to transmit the full spatial angle even at base layer. Unlike these solutions, the proposed TLGA algorithm prioritizes transmission of tiles and their respective layers according to the user gaze, which provides the user with most suitable content.

Other works proposed solutions which have diverse optimisation targets, but considered specific wireless delivery settings, with many simultaneous users. For instance the authors of [41] focused on achieving optimal 360-degree video transmissions to multiple users and employ a multicast-based approach, whereas the same researchers in [42] targeted power optimisation in a MIMO setup. The authors of [43] concentrated on a multi-antenna setup and achieved positive results in both single and multi-user scenarios. Unlike these solutions TLGA is a generic solution and can be deployed in all scenarios (including in conjunction with other solutions), as it adjusts the amount of data to be delivered and thus reduces the pressure on the network.

III. Tile-layering based gaze adaptation (TLGA) Omnidirectional Video Delivery Solution - Principle and Architecture

The main goal of the proposed TLGA is to deliver best possible quality of experience to the user, while minimizing the bandwidth consumption. The idea is to deliver only parts of omnidirectional video that are in the gaze of an observer or are likely to be within user gaze in a short time. The proposed method employs a client-server architecture which employs MPEG-DASH and enables the client to acquire content by HTTP-GET requests. The functional scheme of this method is illustrated in Fig. 1, where the video encoded with conventional equirectangular projection is split into tiles, which are then split into scalable layers that can be transmitted to the user in varying quantity, depending on the position of user's gaze.

TLGA relies on a client-server architecture. The server side deploys a HTTP server which stores omnidirectional videos. Each video stored on the server is divided into tiles with fixed duration (usually 1 to 10 seconds). Tiles are also divided into SVC layers, which refers to a base layer with low quality and bitrate and multiple enhancement layers with increasing

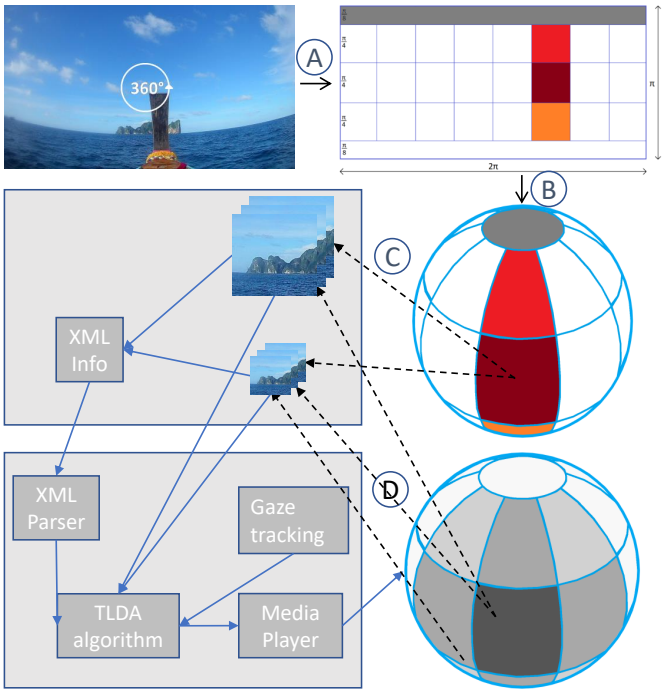

Fig. 1. Overview of the proposed method for adaptive delivery of omnidirectional video: a) splitting of conventional bitstream into tiles, b) projection of tiles on the surface of a sphere, c) splitting of tiles into scalable layers, d) tile in the gaze of an observer is composed of multiple layers as opposed to tiles further from the observer's gaze.

quality and bitrates. The server also provides metadata with a structured description of video representation in XML format, containing information about tile duration, area covered by a given tile and information about layering of the tiles.

The client side consists of a playback device (i.e. virtual reality headset, desktop player or smartphone) deploying the proposed TLGA algorithm. Client first accesses and reads the XML metadata file from server with the information about tiling. User's gaze coordinates are continuously monitored by built-in eye-tracking in the VR headset, head tracking in headset, mouse in case of a desktop playback or accelerometer and gyroscope in case of a playback on a smartphone. TLGA algorithm is responsible for downloading the base layer and enhancement layers according to their distance from gaze. In practice, the number of enhancement layers downloaded by the TLGA algorithm decreases with the increasing distance of a given tile from the user's gaze. When the user suddenly changes their gaze, a tile with an earlier starting time is downloaded and its playback is synchronized onto the same frame with currently play-back content. This also applies for enhancement layers, which are used as an overlay over the base layer. The described process allows for selective video delivery only in the regions of interest and thus helps reduce bandwidth utilisation. More technical details are provided in Section IV-D and in a patent application [44].

IV. TLGA Proof of Concept

In order to test the proposed method, a reference implementation was targeted, which involved effort to create the

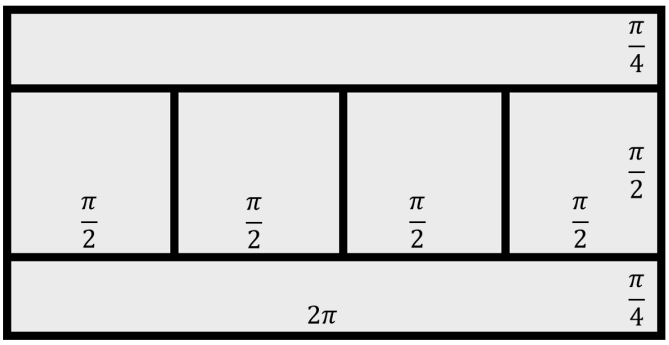

Fig. 2. Cubic projection tiling scheme.

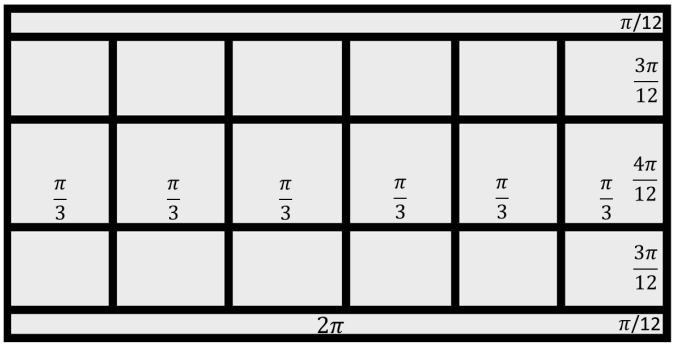

Fig. 3. Squares and poles tiling scheme.

TLGA-enhanced adaptable content at the server and a client application to play adaptable omnidirectional videos in a VR headset.

A. Creating TLGA-adaptable Content

FFmpeg was employed to generate the TLGA adaptable video content. It involves rearranging the keyframes in the video so that each segment starts with a keyframe. Segmenting the video into temporal segments of fixed duration to be able to deliver only parts of video with a short duration. For each segment, multiple spatial tiles are created to be able to deliver only a portion of the video which is in user's viewport. We have used multiple tiling schemes applicable for omnidirectional videos with equirectangular projection, as illustrated in Figs. 2, 3, 4 and 5: Cubic projection, Squares and poles, Progressive tiling and Vertical tiling.

Spatial tiles are then layered using the SVC principle and a simple implementation of signal-to-noise ratio (SNR) scalability based on a Constant Rate Factor (CRF). The Constant Rate Factor is a rate control mode that allows the encoder to attempt to achieve a certain output quality for the whole video file. The CRF scale is ranged between 0–51, where 0 is objectively lossless, 18 is considered to be visually lossless and 51 is the worst possible quality [45]. The base layer is encoded with a CRF of 30 and enhancement layers are encoded with lower CRFs (e.g. highest level enhancement layer is encoded with CRF of 18). The enhancement layer is created by calculating a difference video between the original video tile and it's base layer, where the output video file is shifted to 50% gray level, limited at black and white levels [46]. The described SVC layering is pictured in Fig. 6, along with the result of overlaying one enhancement layer over the base layer and subtracting away 50% gray color. We have used publicly available omnidirectional video for demonstration [47].

Along with the video files, we have also generated the XML media description file. This file includes video extensions, names, information about full video duration, temporal segment duration, spatial information about tiles and information about layering. An example of such a XML file is illustrated in Fig. 7. At the beginning, the naming convention of video files is defined, for example the enhancement layer of the fourth tile with sixth second will have a name of: *"17_UnderwaterPark_segment6_tile4_layer2.mp4"*.

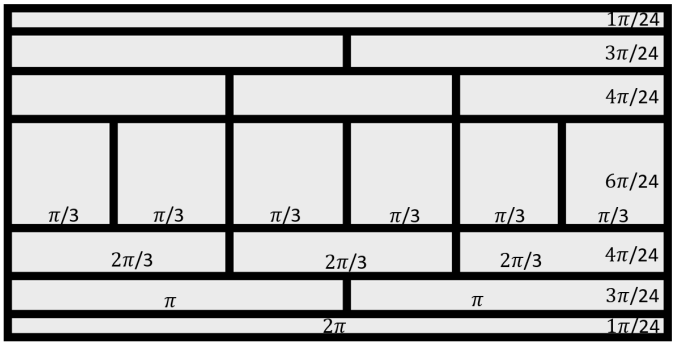

Fig. 4. Progressive tiling scheme.

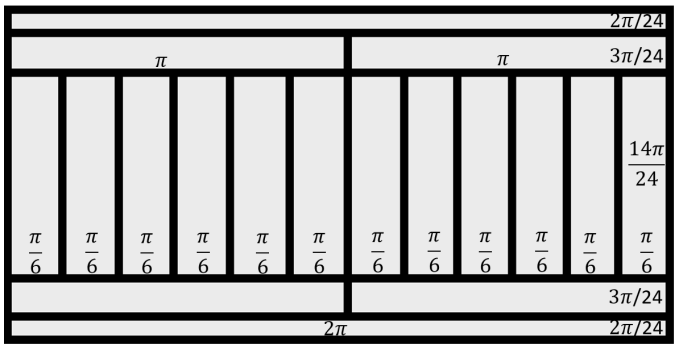

Fig. 5. Vertical tiling scheme.

The video consists of multiple tiles (sectors), where each sector has a defined covered scope (values [*x, y*] from 0 to 1) and offset (values [*x, y*] from 0 to 1). Each sector consists of one or more layers. First layer is the base layer and other layers are enhancement layers which should be added to the base layer. Each layer has a defined quality (CRF) and a video resolution (width and height).

B. HTTP Server Providing Adaptable Omnidirectional Video Content

The created video files with multiple layers are available at the HTTP server along with their associated media description files. For the testing described in this paper, we have used Microsoft internet information services (IIS) on a server on the same local area network (LAN) with the client. The server provided content via HTTP GET requests.

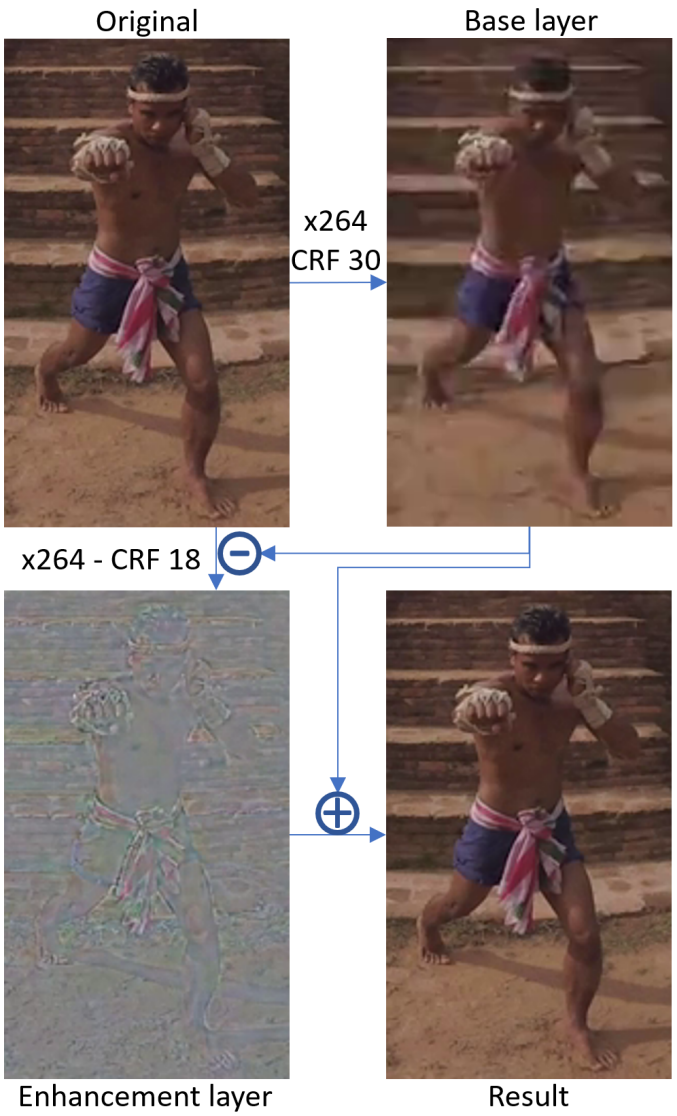

Fig. 6. Scalable video coding with one base and one enhancement layer.

```
<XML xmlns="vda" type="mp4" duration="20000"
segmentDuration="1000" videoName="17_UnderwaterPark"
segmentName="segment%" sectorName="tile%"
layerName="layer%">
  <Sector id="1" tiling="1.0000/0.0833"
offset="0.0000/0.9166">
    <Layer id="1" crf="30" width="3840" height="160"/>
    <Layer id="2" crf="18" width="3840" height="160"/>
  </Sector>
...
  <Sector id="4" tiling="0.1666/0.2500"
offset="0.3333/0.6666">
    <Layer id="1" crf="30" width="640" height="480"/>
    <Layer id="2" crf="18" width="640" height="480"/>
  </Sector>
...
  <Sector id="8" tiling="0.1666/0.3333"
offset="0.0000/0.3333">
    <Layer id="1" crf="30" width="640" height="640"/>
    <Layer id="2" crf="18" width="640" height="640"/>
  </Sector>
...
</XML>
```

Fig. 7. Example of media description XML file.

C. Playback of Adaptable Omnidirectional Video Content

The client *VideoPlayer* deploys the proposed adaptable omnidirectional video playout solution, which as proof of concept was implemented in Unity. *VideoPlayer* parses the XML media description file and according to the tiling scheme, it creates diverse components which render each video tile to a separate texture. Each texture has assigned tiling (scope of omnidirectional scene) and offset. Textures are rendered on a skybox (celestial sphere – an imaginary sphere with infinite radius). An example of client playback and comparison between the playback of base layer and base layer with enhancement layer is shown in Fig. 8.

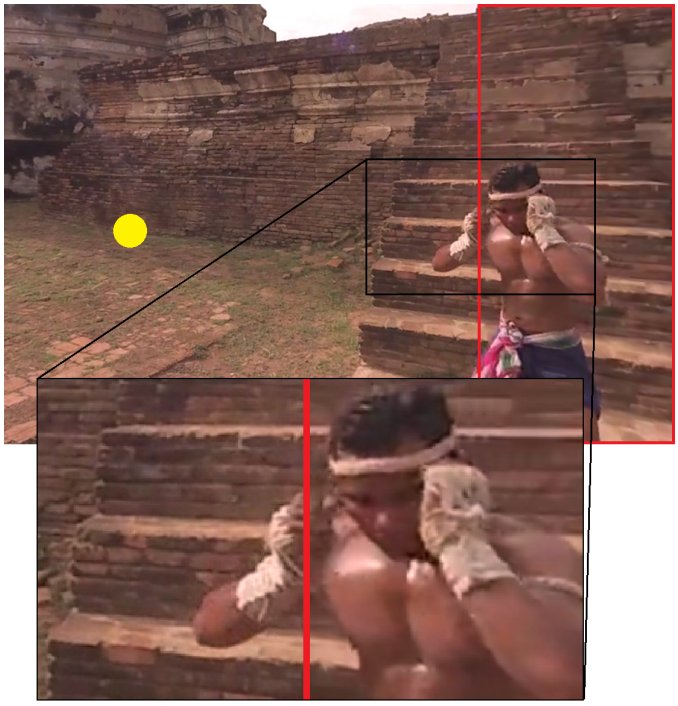

Fig. 8. Example of client playback. The yellow dot is the center of users' gaze. Tiles consist of both base and enhancement layer. Tiles in the red rectangle away from the yellow dot are rendered from the base layer only and have lower quality.

Tiles are buffered into a video sequence buffer from where the proposed TLGA algorithm, described in Subsection IV-D, enables adaptable playout. The video is navigable by mouse drag movements or headset movements in case of playback with a VR headset. There are two different ways to playback a video. *Playback on-time* means that the video tile of a given segment was already loaded when the next temporal segment playback is due to start. In this case all loaded video tiles start their playback at the same time. *Late playback* refers to the case when the video tile was loaded during the time when other tiles of the same segment are already being played. In this case, the tile's video sequence playback starts on the same frame number as the video tiles which are already being played. This means the playback of this tile is temporally synchronized with other video tiles.

D. Tile-layering-based Gaze Adaptation Algorithm (TLGA)

The proposed TLGA algorithm, listed in Algorithm 1, considers that the adaptable omnidirectional video content is already generated and therefore multiple omnidirectional video tiles are available. TLGA is responsible for filling the video sequence buffer with appropriate data and delivering this data

Algorithm 1 TLGA Algorithm

$M \leftarrow$ Number of tiles in a tiling scheme
$BV_{current} \leftarrow$ Current number of buffering video sequences
The update method is called each time a frame is rendered
procedure UPDATE()
  for $int\ s = s_{current}$ **to** $s_{current} + BL$ **do**
    for $int\ i = 1$ **to** M **do**
      if $d(V_s^{i,0}, Gaze) < D_{base}$ **then**
        $PriorityQueue.Enqueue(V_s^{i,0})$
      end if
      if $d(V_s^{i,1}, Gaze) < D_{enhancement}$ **then**
        $PriorityQueue.Enqueue(V_s^{i,1})$
      end if
    end for
  end for
  for each $V_s^{i,l}$ **in** $PriorityQueue$ **do**
    if $s < s_{current}$ **then**
      $PriorityQueue.Remove(V_s^{i,l})$
    else if s **equals** $s_{current}$ **and** $T_s - 2 \cdot \mu_N > t$ **then**
      $PriorityQueue.Remove(V_s^{i,l})$
    end if
  end for
  for $int\ i = 1$ **to** $BV_{max} - BV_{current}$ **do**
    $V_s^{i,l} = PriorityQueue.Dequeue()$
    $DownloadAndDecode(V_s^{i,l})$
  end for
end procedure

procedure DEQUEUE()
  $result = PriorityQueue[0]$
  for each $V_s^{i,l}$ **in** $PriorityQueue$ **do**
    if $P(result) < P(V_s^{i,l})$ **then**
      $result = V_s^{i,l}$
    end if
  end for
  $PriorityQueue.Remove(result)$
  return $result$
end procedure

to be played out. A concentric circular approach is considered for TLGA selection of the quality of the omnidirectional video tiles based on the distance from current user's gaze and possible future user's gaze.

An important parameter for TLGA adaptation is the distance from the center of each tile to user gaze. The shortest distance between two points on the surface of a sphere, the orthodromic distance d, is calculated using the following formula:

$$d = r \cdot arccos(sin(\varphi_1) \cdot sin(\varphi_2) + cos(\varphi_1) \cdot cos(\varphi_2) \cdot cos(\lambda_2 - \lambda_1)) \quad (1)$$

where r is the radius of the sphere and $[\varphi_1, \lambda_1]$, $[\varphi_2, \lambda_2]$ are the spherical coordinates of user gaze and tile center, respectively. A tile is considered for buffering by TLGA when it is in the region of user gaze and can have different quality levels. This region is defined based on the distance from user gaze: less than D_{base} (i.e. $1.8 \cdot r$ in our instance) and tiles are associated with basic quality content (base layer) and less than $D_{enhancement}$ (i.e. $0.9 \cdot r$ in our case) and tiles have improved quality (base and enhancement layers). The algorithm can be extended easily to include a more complex user gaze distance-based adaptation involving multiple enhancement layers. The values of D_{base} and $D_{enhancement}$ were such chosen so that TLGA adaptation will cover the user's field of view with enhanced quality when the user is not moving or moves slowly, and with the base layer (until enhancement layer is loaded) in case the user moves fast or much[1].

The TLGA algorithm uses a priority queue of candidate video sequences, which belong to certain video tiles, are split into temporal video segments and have certain quality levels. These segments need to be buffered and eventually delivered for playout. A video sequence V associated with a given segment number s, tile i, layer l at time t is denoted as $V_s^{i,l}(t)$.

1) Enqueueing video sequence candidates for buffering: The TLGA algorithm maintains a buffer with length BL. In order for a video sequence to be enqueued, the segment number s must be between current playback segment number $s_{current}$ and the $s_{current}$ plus buffer length, inclusive.

$$s_{current} \leq s \leq s_{current} + BL \quad (2)$$

In addition to this condition, if the distance d between a given video sequence of base layer and user gaze is lower than D_{base}, the video sequence is enqueued into the priority queue, as given by equation (3):

$$d(V_s^{i,0}(t), Gaze) < D_{base} \quad (3)$$

Similarly for enhancement layer we have:

$$d(V_s^{i,1}(t), Gaze)) < D_{enhancement} \quad (4)$$

The priority P of a video sequence is given by:

$$P = P_{max} - \alpha \cdot (s - s_{current}) - \beta \cdot d - \gamma \cdot l \quad (5)$$

P_{max} is the maximum priority (we have chosen $P_{max} = 1000$) , and α, β, γ are the prioritisation constants. Constant α is the weight of temporal distance between the current segment and segment of video sequence in question. Constant β is the weight of the spatial distance d between the video sequence tile center and user gaze and γ is the weight of the quality index given by layer l. In the tests reported in this paper, we have used $\alpha = 100$, $\beta = 10$ and $\gamma = 1$. The units of the constants are such chosen so that priority P has no units. The values of the constants were chosen so that the tiles for the already playback segment are considered first, then tiles closest to the user's gaze and lastly their enhancement layers. This prioritises base layer data before enhancement layer for a respective tile.

2) Dequeueing video sequence candidates: In parallel to the enqueuing process of TLGA algorithm, the algorithm is also continuously dequeuing videos from queue for remote delivery or local playout. In order to preserve the performance of a CPU and network with short buffering times, a maximum of number buffering videos BV_{max} can be buffered at once

[1]Values of 1.8 for D_{base} and 0.9 for $D_{enhancement}$ were set considering that conventional head mounted displays with field of view of 94°-110° are employed.

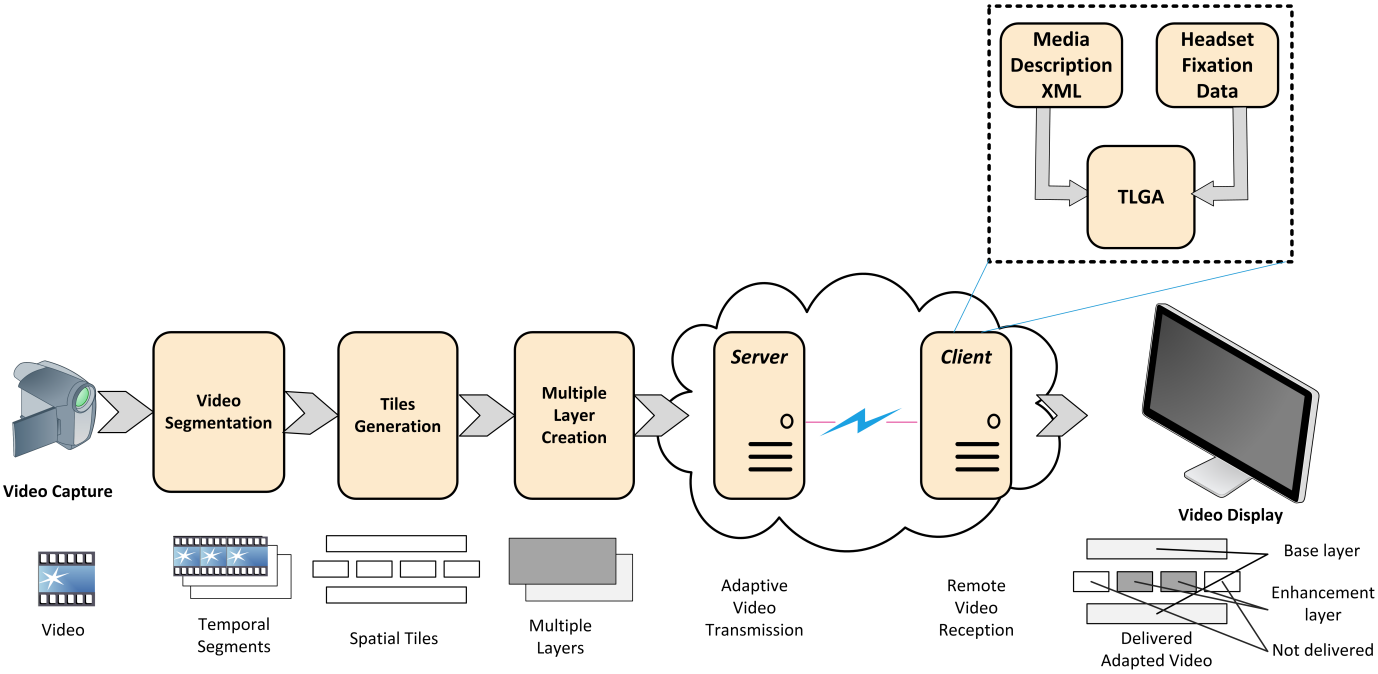

Fig. 9. TLGA was deployed in an adaptive client-server video delivery system for evaluation

(in our tests $BV_{max} = 2$). As the user's gaze can change at any given time, the dequeue priority of a video sequence is decided at the same time the TLGA algorithm requests a dequeue from the priority queue. As the queue is prioritised, the data is sent for delivery or local playout in order of video sequence importance.

3) Removing obsolete video sequences from the queue: With the enqueueing process, there are several candidates of video sequences added to the priority queue. Some video sequences may not be buffered due to having low priority and become obsolete after some time. Any video sequence that has lower segment number s than the current play-back segment $s_{current}$ is removed from the queue:

$$s < s_{current} \tag{6}$$

Also, video sequences where $s = s_{current}$ are considered obsolete when approaching the current segment playback time, as there is no time for delivery and/or decoding. This is done by the condition in equation (7).

$$T_s - 2 \cdot \mu_N > t \tag{7}$$

T_s is the duration of one segment, t is the current time of playback segment and μ_N is the average prepare time of a video sequence $V_s^{i,l}(t)$ and is counted as running average:

$$\mu_{N+1} = \frac{N \cdot \mu_N + t_{prepare}}{N+1} \tag{8}$$

Where N is number of downloaded tiles and $t_{prepare}$ is preparation time for the latest tile.

V. TLGA Evaluation

A. Testing Setup and Scenarios

To evaluate the proposed method in real world conditions we have used a dataset of head and eye movements in Virtual Reality [48]. We have used one omnidirectional video with the highest quality from this dataset with underwater content (*17_UnderwaterPark.mp4*). The video has a duration of 20 seconds and a framerate of 30 fps covering full sphere with an equirectangular projection and resolution of 3840 × 1920. This video comes with head movement trajectories of 57 observers (mean age: 25.7 years, 25 women and 32 men) sampled at 200 ms intervals (labeled *fixation*).

TABLE I
EXPERIMENTAL SETUP

Network setup	
Network router	TP-LINK TL-WR1043ND
LAN Connection	Category 5e ethernet cable
Server setup (notebook)	
Notebook model	Lenovo IdeaPad Y50 59-432682
CPU	Intel Core i5-4210H CPU @2.90GHz
Network adapter	Realtek PCIe GBE Family Controller
Client setup (desktop PC)	
CPU	AMD Ryzen 5 3600 @3.80GHz
GPU	AMD Radeon RX 5700 XT
Motherboard	MSI B450 TOMAHAWK
Network adapter	Realtek PCIe GBE Family Controller

A client-server system which deploys TLGA as described in the previous section was employed during testing. The system is illustrated in Fig. 9, which indicates all TLGA phases from video capture to its display, i.e. video segmentation, tile generation, multiple layer creation, server adaptive transmission and client content reception and display. The four tiling schemes discussed were used in turn: cubic projection, squares and poles, progressive and vertical. The omnidirectional video was prepared for adaptation, as described in section IV-A with segment duration of 1000 ms. The playback client was modified to consume headset fixation data and to simulate

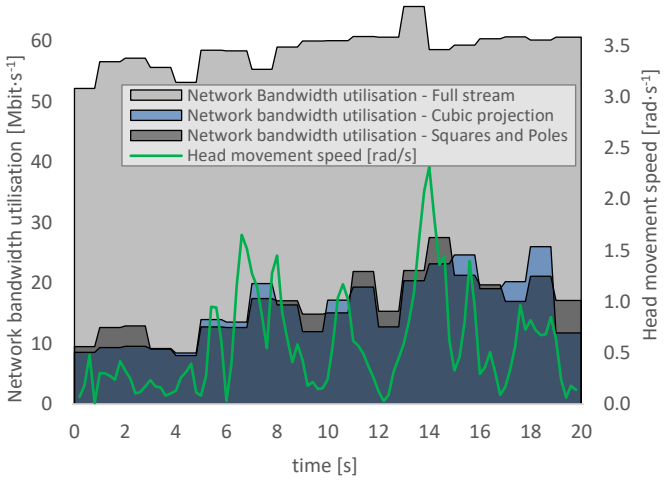

Fig. 10. Network bandwidth utilisation for streaming full video compared to proposed adaptive streaming with cubic projection and squares and poles tiling schemes with associated head movement speed of user number 4.

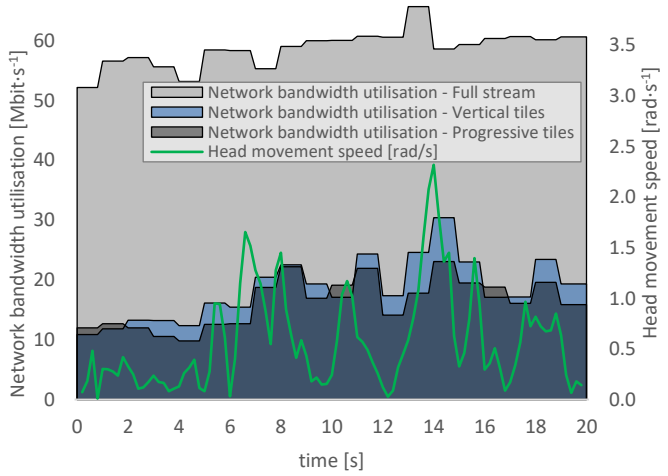

Fig. 11. Network bandwidth utilisation for streaming full video compared to proposed adaptive streaming with vertical and progressive tiling schemes with associated head movement speed of user number 4

head movements with a simple linear interpolation between two fixations. That means each next rendered frame (note: not video frame, but frame rendered into the display from GPU) would have new gaze position counted and view of the camera would be adapted (rotated) accordingly. The server-client-based experimental setup employed during testing has its technical details listed in Table I. The server and client were both connected to an isolated local area network (LAN) through Ethernet cable. The maximum theoretical throughput of the network was 1000 Mbps with a latency < 1 ms, so no negative network delivery influence on the omnidirectional video distribution was recorded. The evaluation is performed in terms of network bandwidth saving and adaptation latency.

B. Network Bandwidth Savings

Network bandwidth utilisation was monitored for each user and the results for a user are presented as an example in Fig. 10 and Fig. 11. These charts compare user head movement speed against current network bandwidth utilisation and demonstrate

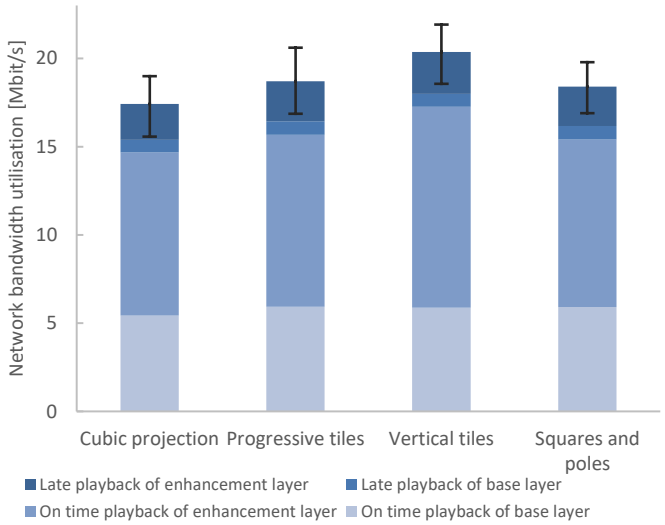

Fig. 12. Network bandwidth utilisation comparison between four proposed tiling schemes.

higher network bandwidth utilisation when the user gaze is changing very fast. This is given by the fact that TLGA algorithm needs to cover the changing user gaze with additional video tiles. Head movement speed ω_t was determined by:

$$\omega_t = \frac{\Delta\sigma_{t,t-\Delta t}}{\Delta t} \tag{9}$$

$\Delta\sigma_{t,t-\Delta t}$ is the angle between user's gaze at time t and time 200 ms earlier and Δt is 200 ms. Using equation (1) for orthodromic distance between user's gaze at these two times we get user's head movement speed ω_t at time t:

$$\omega_t = \frac{arccos(sin(\varphi_t)sin(\varphi_{t-0.2})+cos(\varphi_t)cos(\varphi_{t-0.2})cos(\lambda_{t-0.2}-\lambda_t))}{0.2s} \tag{10}$$

Average network bandwidth utilisation for all tiling schemes and traditional streaming with full viewport are compared with their distribution into on-time playback and late playback in Table II. Playback of tiles from buffer (on-time playback) represents 84% of average network bandwidth utilisation and tiles that were loaded at the same time their segment was already being played (late playback) represents 16% of the bandwidth utilisation. Late playback is caused exclusively by the user changing gaze and TLGA algorithm adapting to the new gaze by downloading tiles that should have been already playing. This could be partially mitigated by employing a user gaze prediction algorithm such as [49]. The proportion of late playback could also be mitigated by using a lower segment duration, however this would negatively impact the encoding efficiency, as each segment needs to have a key frame encoded.

Fig. 12 compares the average network bandwidth utilisation when the four proposed tiling schemes are used delimited by upper and lower quartile of average network bandwidth consumption for each user. Network bandwidth savings of the proposed tiling schemes in comparison to the traditional streaming of full spatial angle are listed in Table III. Results seen in Fig. 12 show similar bandwidth utilisation between all proposed tiling schemes. These results are determined by the selected values of D_{base} and $D_{enhancement}$ and the overlap between the concentric regions and tiles.

TABLE II
AVERAGE NETWORK BANDWIDTH CONSUMPTION DISTRIBUTION

	Average network bandwidth utilisation [Mbit/s]				
	On time playback		***Late playback***		**Total [Mbit/s]**
Layer	*Base*	*Enhancement*	*Base*	*Enhancement*	
Cubic projection	5.43 (31%)	9.23 (53%)	0.72 (4%)	2.02 (12%)	**17.41** (100%)
Progressive tiles	5.94 (32%)	9.74 (52%)	0.77 (4%)	2.27 (12%)	**18.71** (100%)
Vertical tiles	5.87 (29%)	11.40 (56%)	0.73 (4%)	2.36 (12%)	**20.36** (100%)
Squares and poles	5.90 (32%)	9.51 (52%)	0.76 (4%)	2.24 (12%)	**18.40** (100%)
Full viewport	9.86 (17%)	48.81 (83%)			**58.67** (100%)

TABLE III
AVERAGE NETWORK BANDWIDTH UTILISATION

	Bandwidth utilisation	Standard deviation
Cubic projection	30%	4%
Progressive tiles	32%	4%
Vertical tiles	35%	4%
Squares and poles	31%	4%

TABLE IV
ADAPTATION LATENCY

	Tile prepare time [ms]	Standard deviation [ms]
Cubic projection	125.94	5.66
Progressive tiles	73.59	2.55
Vertical tiles	73.78	2.18
Squares and poles	70.87	2.37

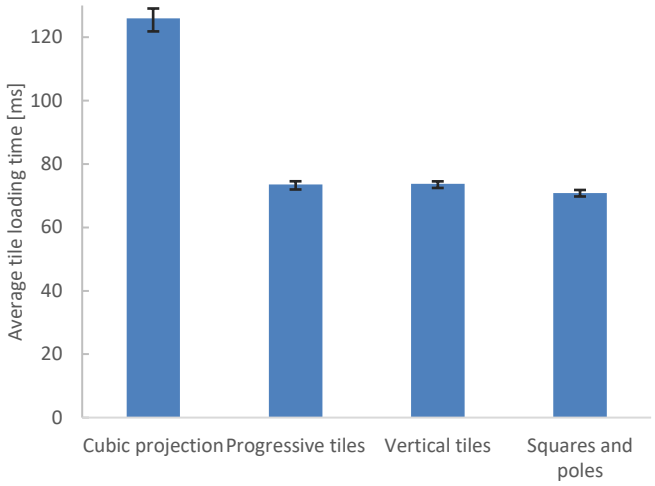

Fig. 13. Adaptation latency comparison between four tiling schemes.

It can be noted how by employing TLGA the network bandwidth consumption was decreased to **30-35%** of the original bandwidth consumption without adaptation, for all tiling schemes.

C. Adaptation Latency

The adaptive approach employed by TLGA streams only a portion of the scene, and one of the most important parameters to evaluate is the time it takes to adapt to changes in user's viewport. Fig. 13 depicts time to download and decode one tile for each tiling scheme delimited by upper and lower quartile of tile preparation time. Tile preparation time of Cubic projection was 125.94 $\pm$ 5.66 ms, highest of all tiling schemes. This is because the cubic projection is splitting the omnidirectional video into only 6 tiles, unlike progressive tiles (18 tiles), vertical tiles (18 tiles) and squares and poles (20 tiles). The tiles of other three tiling schemes cover a smaller region and are also smaller in file size and faster to decode. The lowest tile preparation time of **70.87** $\pm$ **2.37 ms** was achieved when using the squares and poles tiling scheme, which has the smallest tiles among the tested tiling options.

According to [50] latency of 50-70 ms in foveated rendering can be tolerated, however in this research, lower values of peripheral eccentricity were used (up to 20°, which is 0.349066 rad, in comparison, the proposed method had full quality within fovea size of 0.9 rad). Big tiles as used in cubic projection tiling scheme could prove to be effective in case of rapid changes in user gaze, where smaller tiles would need to be downloaded in high numbers. Tiling schemes with smaller tiles seem to be more suitable for ordinary movements, where the user gaze is changing slowly enough so that only one new tile is needed. Also, to keep the adaptation latency comparable between all tiles, they should be of comparable area, in order to enable both download and decoding with similar complexity. The original video could be split into more tiles than proposed in this paper to achieve better latency. However, the latency is also limited to other factors of end-to-end latency and also the encoding efficiency will be negatively affected, since the similarities between the tiles cannot be utilised when encoding.

VI. CONCLUSIONS

This paper has introduced the novel Tile-layering based gaze adaptation algorithm (TLGA) which adjusts the omnidirectional video delivery with user head movement such as the highest quality content is in the area where the user is looking at and the content quality decreases further from that. TLGA involves omnidirectional video split into multiple temporal segments, which are then further spatially split into tiles that can be encoded at multiple quality layers. A proof of concept implementation of the proposed method was realised in a client-server video delivery system and testing was performed with head movement and gaze-based data from real users. Efficiency of four different tiling schemes were compared. The results show that when employing the proposed TLGA method the network bandwidth consumption decreased to 30% of the original bandwidth consumption with adaptation latency of 70.87 ms, while showing the best quality of content around the viewer area of interest. The proposed method is not limited to head tracking and can be extended by considering eye-tracking to further decrease the network bandwidth consumption and adaptation latency and improve overall perceived QoE.

ACKNOWLEDGMENT

The research described in the paper was financially supported by the European Union Horizon 2020 project NEWTON, Grant Agreement No. 688503. G.-M. Muntean would also like to acknowledge the support of Science Foundation Ireland (SFI) Research Centres Programme under Grant 12/RC/2289_P2 (Insight SFI Centre for Data Analytics).

REFERENCES

[1] D. C. Niehorster, L. Li and M. Lappe, "The Accuracy and Precision of Position and Orientation Tracking in the HTC Vive Virtual Reality System for Scientific Research", *i-Perception*, 8(3), 2017.

[2] J. D. Moss and E. R. Muth. "Characteristics of headmounted displays and their effects on simulator sickness", *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(3):308–319, 2011.

[3] A. Borrego, J. Latorre, M. Alcañiz and R. Llorens. "Comparison of Oculus Rift and HTC Vive: Feasibility for Virtual Reality-Based Exploration, Navigation, Exergaming, and Rehabilitation", *Games for Health Journal*, Jun 2018, pp. 151-156.

[4] "Upload virtual reality videos" [Online], 2018, *Google Inc.*, Available at: https://support.google.com/youtube/answer/6316263?hl=en&ref_topic=2888648.

[5] "Coding of audio-visual objects—Part 2: Visual", *ISO/IEC 14492-2 (MPEG-4 Visual)*, ISO/IEC JTC 1, Version 1: Apr. 1999, Version 2: Feb. 2000, Version 3: May 2004.

[6] T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Joint Draft 11 of SVC Amendment", *Joint Video Team*, Doc. JVT-X201, Jul.2007.

[7] "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding", *ISO/IEC 23008-2:2017*, ISO/IEC JTC 1/SC 29, 2017.

[8] G.-M. Muntean, P. Perry and L. Murphy, "Subjective assessment of the quality-oriented adaptive scheme," in *IEEE Transactions on Broadcasting*, vol. 51, no. 3, pp. 276-286, Sept. 2005.

[9] Z. Yuan, G. Ghinea and G.-M. Muntean, "Beyond Multimedia Adaptation: Quality of Experience-Aware Multi-Sensorial Media Delivery," in *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 104-117, Jan. 2015.

[10] ISO/IEC 23009-1:2014(E), Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats, 2014.

[11] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62-67, April 2011.

[12] L. Zou, R. Trestian and G. Muntean, "E3DOAS: Balancing QoE and Energy-Saving for Multi-Device Adaptation in Future Mobile Wireless Video Delivery," in *IEEE Transactions on Broadcasting*, vol. 64, no. 1, pp. 26-40, March 2018.

[13] B. Rainer, S. Petscharnig, C. Timmerer and H. Hellwagner, "Statistically Indifferent Quality Variation: An Approach for Reducing Multimedia Distribution Cost for Adaptive Video Streaming Services," in *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 849-860, April 2017.

[14] ISO/IEC 23009-1:2014/Amd 2:2015, Spatial relationship description, generalized URL parameters and other extensions.

[15] L. D'Acunto, J. Berg, E. Thomas, O. Niamut. "Using MPEG DASH SRD for zoomable and navigable video", *Proceedings of the 7th International Conference on Multimedia Systems - MMSys '16*. New York, USA: ACM Press, pp. 1-4, 2016.

[16] O. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, S. Yong Lim. "MPEG DASH SRD: spatial relationship description", *Proceedings of the 7th International Conference on Multimedia Systems (MMSys '16). Association for Computing Machinery* , New York, NY, USA, Article 5, pp. 1–8.

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

[18] B. Ciubotaru, G. Ghinea and G. Muntean, "Subjective Assessment of Region of Interest-Aware Adaptive Multimedia Streaming Quality," in IEEE Transactions on Broadcasting, vol. 60, no. 1, pp. 50-60, March 2014

[19] J. Montalban, G.-M. Muntean and P. Angueira, "A Utility-based Framework for Performance and Energy-aware Convergence in 5G Heterogeneous Network Environments", IEEE Transactions on Broadcasting, vol. 66, no. 2, June 2020, pp. 589-599

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

[21] G.-M. Muntean, P. Perry and L. Murphy, "Objective and subjective evaluation of QOAS video streaming over broadband networks," in IEEE Transactions on Network and Service Management, vol. 2, no. 1, pp. 19-28, Nov. 2005

[22] L. Zhong, X. Ji, Z. Wang, J. Qin and G.-M. Muntean, "A Q-learning Driven Energy-aware Multipath Transmission Solution for 5G Media Services", IEEE Transactions on Broadcasting, Special Issue on "5G Media Production, Contribution and Distribution", vol. 68, no. 2, June 2022

[23] A. Yaqoob and G.-M. Muntean, "A Combined Field-of-View Prediction-assisted Viewport Adaptive Delivery Scheme for 360° Videos", IEEE Transactions on Broadcasting, vol. 67, no. 3, September 2021, pp. 746-760

[24] Z. Yuan, G. Ghinea and G.-M. Muntean, "Beyond Multimedia Adaptation: Quality of Experience-Aware Multi-Sensorial Media Delivery," in IEEE Transactions on Multimedia, vol. 17, no. 1, pp. 104-117, Jan. 2015

[25] M. Xu, P. Fu, B. Liu and J. Li, "Multi-Stream Attention-Aware Graph Convolution Network for Video Salient Object Detection," in IEEE Transactions on Image Processing, vol. 30, pp. 4183-4197, 2021

[26] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," in IEEE Communications Surveys Tutorials, vol. 21, no. 1, Firstquarter 2019, pp. 562-585

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

[28] X. Corbillon, G. Simon, A. Devlic a J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery", *2017 IEEE International Conference on Communications (ICC)*, Paris, 2017, s. 1-7.

[29] M. Hosseini, V. Swaminathan, "Adaptive 360 VR Video Streaming based on MPEG-DASH SRD", *2016 IEEE International Symposium on Multimedia (ISM)*, pp. 407-408, 2017.

[30] M. Hosseini, V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer!", *Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM '16)*, San Jose, USA, 2016.

[31] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge and T. Schierl, "HEVC tile based streaming to head mounted displays", *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2017, pp.613-615

[32] S. Zhao, D. Medhi, "SDN-Assisted adaptive streaming framework for tile-based immersive content using MPEG-DASH", *IEEE NFV-SDN Conference*, 2017.

[33] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, "Foveated 3D graphics", *ACM Transactions on Graphics (TOG) 31*, no. 6 (2012):164

[34] P. Anjul, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and David Luebke, "Perceptually-based foveated virtual reality", *ACM SIGGRAPH 2016 Emerging Technologies*, pp. 17. ACM, 2016.

[35] MHD. Y. Saraiji, K. Minamizawa, S. Tachi, "Foveated Streaming: Optimizing video streaming for Telexistence systems using eye-gaze based foveation", *Virtual Reality Society of Japan*, 2017.

[36] G. Illahi, M. Siekkinen, and E. Masala, "Foveated video streaming for cloud gaming," *IEEE Int. Workshop on Multimedia Signal Proc.(MMSP)*, 2017.

[37] M. F. R. Rondon, L. Sassatelli, F. Precioso, R.A. Pardo. "Foveated Streaming of Virtual Reality Videos", *ACM International Conference on Multimedia Systems (MMSys)*, 2018.

[38] R. Ghaznavi-Youvalari, A. Zare, A. Aminlou, M. M. Hannuksela and M. Gabbouj, "Shared Coded Picture Technique for Tile-Based Viewport-Adaptive Streaming of Omnidirectional Video," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 29, no. 10, pp. 3106-3120, Oct. 2019.

[39] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R Prakash, "Adaptive 360-Degree Video Streaming using Scalable Video Coding," *ACM International Conference on Multimedia (MM)*, New York, NY, USA, pp. 1689–1697, 2017.

[40] A. T. Nasrabadi, A. Mahzari, J. D. Beshay and R. Prakash, "Adaptive 360-degree video streaming using layered video coding," 2017 IEEE Virtual Reality (VR), 2017, pp. 347-348

[41] C. Guo, Y. Cui and Z. Liu, "Optimal Multicast of Tiled 360 VR Video," in IEEE Wireless Communications Letters, vol. 8, no. 1, pp. 145-148, Feb. 2019

[42] C. Guo, L. Zhao, Y. Cui, Z. Liu and D. W. K. Ng, "Power-Efficient Wireless Streaming of Multi-Quality Tiled 360 VR Video in MIMO-OFDMA Systems," in IEEE Transactions on Wireless Communications, vol. 20, no. 8, pp. 5408-5422, Aug. 2021
[43] L. Zhao, Y. Cui, Z. Liu, Y. Zhang and S. Yang, "Adaptive Streaming of 360 Videos With Perfect, Imperfect, and Unknown FoV Viewing Probabilities in Wireless Networks," in IEEE Transactions on Image Processing, vol. 30, pp. 7744-7759, 2021
[44] A. Polakovič, G. Rozinaj, R. Vargic, and G.-M. Muntean, System and method for adaptation of omnidirectional video delivery. Patent application number 134-2019, 09.09.2019. Available at: https://wbr.indprop.gov.sk/WebRegistre/Patent/Detail/101-2019.
[45] "H.264 Video Encoding Guide" [Online], FFmpeg, available at: https://trac.ffmpeg.org/wiki/Encode/H.264, 2020.02.16.
[46] M. Koch, "An Introduction to FFmpeg, Timelapse and Fulldome Video Production, Color Grading, Audio Processing, Panasonic LUMIX GH5S, Image Processing and Astronomy Software", Feb. 14, 2020. [Online] Available at: http://www.astro-electronic.de/FFmpeg_Book.pdf.
[47] "Omnidirectional video: Ayutthaya". [Online] Available at: https://www.mettle.com/360vr-master-series-free-360-downloads-page/.
[48] E. David, J. Gutiérrez, P. L. Callet, A. Coutrot, M. P. D. Silva, "A Dataset of Head and Eye Movements for 360° Videos", *ACM on Multimedia Systems Conference (MMSys)*, Amsterdam, Netherlands, June 2018.
[49] F. Qian, L. Ji, B. Han, V. Gopalakrishnan. "Optimizing 360 video delivery over cellular networks". *ACM Workshop on All Things Cellular Operations, Applications and Challenges*, New York, USA, 2016, pp. 1-6
[50] R. Albert, A. Patney, D. Luebke, and J. Kim. 2017. "Latency Requirements for Foveated Rendering in Virtual Reality". *ACM Transactions on Applied Perception 14*, 4, Article 25 (September 2017), 13 pages.

Adam Polakovič received his PhD degree from the Institute of Multimedia Information and Communication Technologies at the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology Bratislava in 2021. He holds BSc and MSc degrees in Biomedical Physics from the Faculty of Mathematics, Physics and Informatics, Comenius University, Slovakia awarded in 2015 and 2017. His research interests include adaptation of multimedia streaming for bandwidth saving and increased end user quality of experience.

Gregor Rozinaj received MSc. degree from the Slovak University of Technology in Bratislava, Slovak Republic in 1981 and PhD at the same university in 1990. Now he works as a full professor in Telecommunications at the Slovak University of Technology in Bratislava and serves as a director of the Multimedia Information and Communication Technologies Institute. He also served as a vice-dean for international relations at the Faculty of Electrical Engineering and Information Technology of STUBA. His research interests include multimedia and speech processing. He published more than 140 research articles. He gave 15 invited lectures in several countries. He is an author of several patents, three of them patented worldwide. He was an STU coordinator of H2020 project NEWTON and technical coordinator of FP7 project HBB-NEXT. He is also experienced as a coordinator and manager of many national research and development projects oriented to multimedia processing. He was a supervisor of 15 successfully finished PhD students. He has participated in several educational projects under Tempus PHARE, LdV, ERASMUS+ and other EC programmes and as a trainer in several LLL EC and national projects. He serves a member of Editorial board of Journal of Advanced Signal Processing and other international journals and IPC member and reviewer of many international conferences.

Gabriel-Miro Muntean (S'02–M'04-SM'17) received the B.Eng. and M.Sc. degrees in computer science engineering from the Politehnica University of Timisoara Romania, in 1996 and 1997, respectively, and the Ph.D. degree in electronic engineering from Dublin City University (DCU) Ireland, in 2003. He is currently a Professor with the DCU School of Electronic Engineering, co-Director of the DCU Performance Engineering Laboratory, and a Consultant Professor with the Beijing University of Posts and Telecommunications, China. He has authored or coauthored over 450 papers in prestigious international journals and conferences, has authored 4 books and 23 book chapters, and has edited 9 other books. His current research interests include quality-oriented and performance-related issues of adaptive multimedia delivery, performance of wired and wireless communications, energy-aware networking, and technology-enhanced learning. Prof. Muntean is a senior member of IEEE and IEEE Broadcast Technology Society. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, Multimedia Communications Area Editor of the IEEE COMMUNICATION SURVEYS AND TUTORIALS, and a reviewer for other international journals, conferences, and funding agencies. He coordinated the EU Horizon 2020 project NEWTON.