Gaze-aware streaming solutions for the next generation of mobile VR experiences

Pietro Lungaro, Rickard Sjöberg, Alfredo José Fanghella Valero, Ashutosh Mittal and Konrad Tollmar

Abstract—This paper presents a novel approach to content delivery for video streaming services. It exploits information from connected eye-trackers embedded in the next generation of VR Head Mounted Displays (HMDs). The proposed solution aims to deliver high visual quality, in real time, around the users' fixations points while lowering the quality everywhere else. The goal of the proposed approach is to substantially reduce the overall bandwidth requirements for supporting VR video experiences while delivering high levels of user perceived quality. The prerequisites to achieve these results are: (1) mechanisms that can cope with different degrees of latency in the system and (2) solutions that support fast adaptation of video quality in different parts of a frame, without requiring a large increase in bitrate. A novel codec configuration, capable of supporting near-instantaneous video quality adaptation in specific portions of a video frame, is presented. The proposed method exploits in-built properties of HEVC encoders and while it introduces a moderate amount of error, these errors are indetectable by users. Fast adaptation is the key to enable gaze-aware streaming and its reduction in bandwidth. A testbed implementing gaze-aware streaming, together with a prototype HMD with in-built eye tracker, is presented and was used for testing with real users. The studies quantified the bandwidth savings achievable by the proposed approach and characterize the relationships between Quality of Experience (QoE) and network latency. The results showed that up to 83% less bandwidth is required to deliver high QoE levels to the users, as compared to conventional solutions.

Index Terms—Eye-tracking, VR, QoE, video streaming, content delivery

1 Background

Based upon the latest available data [5], the VR industry seems to be finally gaining momentum: 2.25M VR headsets were shipped in Q1 2017, representing a 69% increase as compared to the same quarter 2016. As expected, this young market is dominated by the most affordable devices, with Samsung VR and PlayStation VR jointly having a market share of about 40%. Both solutions reuse end-users' own hardware, thus they can be considered "upgrades" rather than entirely new systems, in comparison with the more advanced HTC Vive and Oculus Rift. These advanced systems support higher resolutions but require dedicated high performance machines to run. Their substantially higher system costs are reflected in their lower market shares: HTC Vive and Oculus Rift jointly share slightly less than 13% of the market.

Immersive experiences delivered via 360° videos are increasingly becoming popular and an extensive ecosystem is steadily growing around both "live" and "on demand" video provisioning. Led by technology giants such as Facebook and Google, a series of innovations is fostering 360° videos, including novel camera rigs and sophisticated encoding solutions [25]. However, one of the major problems of video streaming is the large amount of bandwidth required to support 4K videos. While YouTube guidelines [3] recommend bitrates in the range of 35-85 Mbps, depending on frame rate and dynamic range, the average throughput for the top ten countries in the world are 18.7-28.6 Mbps [2], suggesting that acceptable user experience is highly unlikely for the vast majority of potential users. This gap between bandwidth requirements for high resolution video and available bandwidth is likely to increase further, with both the introduction of 8K capable cameras and HMDs and the growing expectations for higher resolution video also in *mobile* VR systems. Rather than "throwing more bandwidth at the problem" an alternative approach is to develop novel video streaming solutions capable of disrupting current content provisioning paradigms.

- *Pietro Lungaro, Alfredo José Fanghella Valero, Ashutosh Mittal and Konrad Tollmar are with KTH Royal Institute of Technology, Stockholm, Sweden. E-mails: {pietro, ajfv, amittal, konrad}@kth.se.*
- *Rickard Sjöberg is with Ericsson Research, Stockholm, Sweden. E-mail: rickard.sjoberg@ericsson.com.*

Manuscript received 11 Sept. 2017; accepted 8 Jan. 2018.
Date of publication 19 Jan. 2018; date of current version 18 Mar. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2794119

1.1 Human vision and foveal rendering

The "fovea centralis" is a very small region of the retina with a high density of cones. This is typically considered as the area of the eye which is responsible for sharp *central vision*. In fact, the acuity of the human eye decays rapidly within 1-5 degrees from the fovea and visual information not directly mapped onto the fovea is typically considered as contributing to *peripheral vision* [9]. While the latter component of visual perception is very important for the overall visual experience, e.g. detecting movements, it requires and processes information at lower resolution. Further, since the fovea covers only about the central five degrees of the visual field, the perception of an entire scene is achieved in the brain by merging together individual images captured at various fixation points throughout successive "saccadic" eye movements. These represent quick transitions when changing from one focus point to another. During these rapid eye movements, a process called "saccadic suppression" [35], [28], [29] dampens visual sensitivity, starting about 50 ms before the saccade and continuing up to 40-60 ms after the saccade ends [35]. Eye trackers provide accurate measurements of users' eye gaze, and that information can be used to estimate the portion of the screen that is mapped onto the user's fovea. This insight has in recent years inspired a series of research efforts towards optimization of content display systems. *Foveal rendering* in particular is considered a crucial technique to reduce processing power requirements and costs for using content with resolution beyond 4K [19]. The latest investigations predict the landing positions of saccades. This can be used to preemptively optimize the selection of high quality regions, thus compensating for potential system delays [8]. Other interesting approaches include fovea inspired computer vision applications, such as object detection [7] and video and image coding [38]. The common idea is to achieve spatial optimization of the visual information by assigning higher image resolution to the areas estimated to match the user's fovea, while lowering the image quality in those portions of the scene that contribute only to peripheral vision.

2 Proposed solution

Foveal rendering [10] can optimize computational resources at the client [19], once the content is locally available in the user device. However, the real challenge for supporting data intensive applications like 360° video in future mobile systems is to effectively transport the **relevant** information from where the content is created and/or stored to end-user devices. To address this content transport problem,

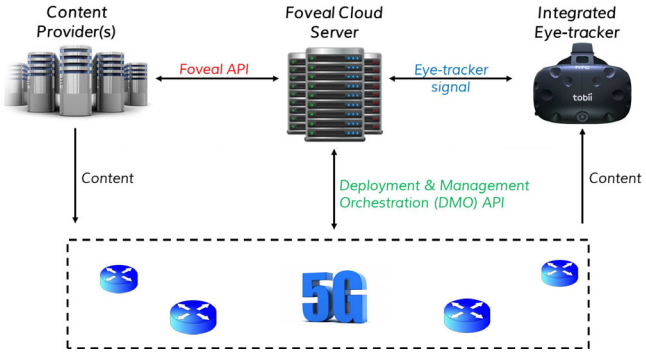


Fig. 1. Overview of the envisioned end-to-end architecture for supporting gaze-aware streaming solutions for VR.

this paper explores a novel gaze-aware streaming solution designed to extend the benefits of foveal rendering to the overall end-to-end content delivery. The basic principle of our proposed approach is *to minimize the transport of bits that do not contribute meaningfully to the end-user experience*. A pre-requisite for this to work is the definition of a novel streaming protocol with a "feedback loop", transmitting updated eye-gaze information from eye-trackers to content caches.

A possible design of a system architecture supporting *gaze-aware* service provisioning is shown in Figure 1. An eye-tracker is co-located with the user's screen where the content is displayed. This could be a standard 2D screen or a VR HMD. The eye-gaze information captured by the eye-tracker is communicated to a *Foveal Cloud Server*, where most functionalities for supporting gaze-aware content provision are located. These include the possibility of storing eye-gaze data for different users and different content items and types. This data can in turn be used by a "profiler" to build models of both how specific users visually consume different types of content and *saliency maps* describing how specific multimedia objects are consumed by the general population. In fact, by systematically storing and processing large volumes of user data over time, the Foveal Server can improve its models for predicting content consumption, e.g. [31] and [8]. An interesting application includes pre-encoding with high quality the areas adjacent to predicted Points of Interest (PoIs) in order to improve the user's experience even when eye-tracking devices are not used. At the same time, the foveal cloud server can also monitor, or receive information on, network and user context information. In Figure 1, this information exchange is represented via the Deployment and Management Orchestration (DMO) API. This API is intended for usage in evolved 5G cloud services, to adapt system and service parameters to significant variations in network conditions, thus compensating for the potential losses or increases in QoE. These operations are performed in the *Service Composition* unit within the Foveal Cloud Server. This unit specifies which portions of a video frame need to be activated with high and low qualities and which resolutions to be used for given eye-gaze positions and network conditions. This information is then exchanged, via a dedicated *Foveal API*, to the providers' caches for orchestrating the content retrieval. The specific functionalities and syntax of the Foveal API are currently being tested and defined before public release.

2.1 Latency

In order for the aforementioned gaze-aware streaming solution to deliver acceptable QoE, the system needs to quickly react to variations in users' eye-gaze positions. If accurate gaze predictions are not available, end-to-end latency plays a major role in determining the user experience levels attainable by this novel service provision method. The main components contributing to the *overall system latency* (Δ_t) are:

- **Eye-tracker latency (EL)** - time between the capture of an eye image and the availability of its gaze position in the client device.
- **Network latency (RTT)** - this is the network Round-Trip Time (RTT), referring to the time between when a new eye-gaze sample is sent to the foveal server and when the corresponding updated content is received at the user side.
- **Processing latency (PL)** - time spent to decode and process the received content and to submit it for rendering in HMD.
- **Display latency (DL)** - time required to render and output to the user HMD a content frame.

Even assuming 5G latencies of 1ms for RTT, an important pre-requisite for a feasible gaze-aware solution is the definition of a coding technique that can support fast quality switching in a specific portion of a video frame. Moreover, apart from being fast, a candidate solution needs to achieve high levels of compression efficiency. As one of the goals of our approach is to enable mobile VR, it is important to exploit gaze-awareness to dramatically reduce the overall bandwidth needed for content delivery.

3 Problem Statement

One of the main goals of the paper is to assess the feasibility of a novel content delivery method exploiting eye-gaze information to optimize the provisioning of 360° video streaming services. The evaluation of both system performance and user experience, for different system configurations, is a key component of our experimental investigations. Specifically, the following research questions are addressed and presented in the remainder of the paper:

- Can a novel video codec configuration be designed to support the rapid quality adaptations as required by gaze-aware streaming? What is the impact on content storage requirements?
- What are the bandwidth requirements of the proposed solution as compared to current non gaze-aware streaming? How does performance change with network latency?
- What is the impact on user experience? Is it possible for this novel approach to deliver a level of QoE that is indistinguishable for the end-users from delivering the entire video at full resolution?

4 Related Literature

Feng et al. [17] used eye-tracking to optimize conventional video streaming. They describe a Hidden Markov Model used to predict the user's gaze based on the available data, as a way to deal with latency. They use a real time encoder to encode the region around the user's predicted gaze in higher quality than the rest. They achieved up to 29% bitrate savings without the users reporting quality degradation, with RTT as high as 200 ms. Both Ghinea and Muntean [18], and Ghuinea, Muntean, and Sheehan [26] propose adaptive streaming techniques that leverage eye-tracking. Similar to [17], the work in [18] adapts the content in real-time as a function of the user's gaze. Instead of aiming at reducing bitrate, their system only performs this adaptation when the network conditions make it necessary. Through simulations, they show that their system can outperform non-adaptive streaming in terms of average client throughput, loss, and estimated user-perceived quality. The earlier paper [26] pre-encodes regions within a frame with different qualities, based on eye-tracking data, in an attempt to preserve perceived quality for lower available datarates. Simulation results, together with subjective tests, show a positive impact on QoE.

El-Ganainy and Hafeeda [16] present an overview of the state of the art in content delivery and streaming solutions for VR. They mention the concept of tiled streaming, central to our work, and state that adapting rapidly enough is one of the challenges to overcome. Hosseini and Swaminathan [20] split equirectangular videos into independent tiles, encoded using H.264 with various bitrates. Notably, this is one of the very few studies about VR streaming that includes user testing results. Tiles are selected based on the user's current viewport. However, the tests are quite restricted, with the experimental design limiting the direction of head movement of the users. Under these optimistic settings bandwidth savings of up to 72% were reported. Facebook's pyramid mapping [25] also uses a viewport based streaming scheme with 5 different resolutions for 30 possible viewports, resulting in 150 encodings of the same video. Every second, a different version can be selected according to the user's viewport direction and available

bandwidth. They claim that their pyramid mapping can reduce storage size by 80%, but no specific results are presented for bandwidth saving or QoE. A similar approach using a "Truncated Square Pyramid (TSP)" geometry has been proposed by Qualcomm [32]. It utilizes 30 partially overlapping viewports and shows coding gains of about 10% as compared to downsampled cube mapping. An optimization of viewport selection targeting specific bandwidth and storage constraints is presented in [11]. This solution is designed to operate with latencies of a few seconds and can outperform a static system transmitting the entire 360° video by 2.9dB in Peak Signal-to-Noise ratio (PSNR). A coding and streaming solution for VR is presented by Kammachi et. al. in [23] using a layered solution with a high quality layer obtained by upsampling the individual views in a lower layer. The latter comprises non overlapping low quality views that are always transmitted, effectively covering the entire 360°. In the higher layer, only the high quality views corresponding the latest head direction of the user are selected for display. Whenever the head direction points at areas not included in the current high quality view, images from the background layer are served. This approach shows bandwidth reductions of 42% as compared to when all high quality views are simultaneously streamed.

Quian et al. [27], Corbillon et al. [14], and Zare et al. [37] propose viewport adaptation techniques that include tiles. Quian et al. describe a way to predict the position of the users' heads. Using simulations driven by recorded eye-tracking data, they claim their system can save up to 80% bandwidth. Neither content preparation or how fast new tiles can be requested are described in detail. Corbillon et al. compare different mappings for content storage and different segments lengths (i. e. different times between random access pictures). A cubemap projections with 2 seconds long segments provides the best performance, according to the metric considered in their paper. Zare et al. propose using HEVC tiles, since only a single decoder is required. They analyze their system's performance in terms of storage and bitrate, using simulations. Their results show bitrate reductions between 30% and 40% with minimal extra storage requirements.

5 A NOVEL VIDEO CODEC CONFIGURATION

Our proposed approach is based on the High Efficiency Video Coding (HEVC) compression standard [21]. Of special importance is the concept of video tiles, which is central to this standard. As described in [30], tiles partition a picture into rectangular sections at fixed locations. This is primarily done to enable parallel processing. Moreover, since tiles are independently decodable, this approach can be used to separately address and process individual portions of a frame. HEVC supports three types of frames: I-frames, which use only intra-prediction and can be decoded by themselves; P-frames, which are encoded using only one previous or future picture; and B-frames, which can use two, typically one past and one future pictures. Figure 4a shows what a typical chain of frames can look like. Since I-frames can be decoded independently, they serve as Random Access Points (RAPs): a video can be decoded starting from any I-frame, ignoring all previous data. However, I-frames require more storage space than the other types of frame. An important parameter of HEVC is the Quantization Parameter (QP). For 8-bit video sequences, QP ranges from 0 to 51, with lower QP values leading to better image quality but lower compression ratios.

5.1 Tiles and regions

In our proposed solution, videos are split into tiles and these are dynamically assigned to one of two regions: the **foreground region**, in which tiles have high quality and the **background region**, where content is instead displayed with lower quality. As discussed in Section 1.1, visual acuity drops with increasing angular distance from the fovea. A set of central and peripheral regions are illustrated in Figure 2a. Visual acuity declines by about 50% every 2.5° from the center up to 30°, while beyond this it declines more rapidly. Color perception is strong at 20° but weak at 40° [9], thus 30° is typically considered also a threshold for color perception. In our approach, the foreground region is centered around the gaze point reported by the eye-tracker, identified by all the tiles intercepted by a right circular cone whose vertex is the center of the HMD, with an axis passing through the estimate gaze position,

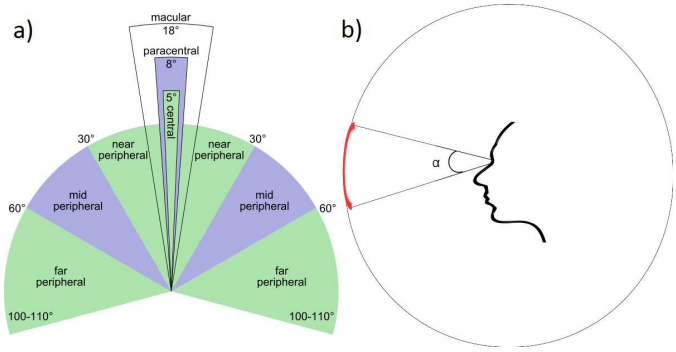


Fig. 2. a) Regions in the central and peripheral vision. Image from Wikimedia Commons, author Zyxwv99 (https://commons.wikimedia.org/wiki/File:Peripheral_vision.svg), https://creativecommons.org/licenses/by-sa/4.0/legalcode. b) Aperture α of the right circular cone identifying the foreground region.

Fig. 3. Radii used to discretize the intersection between the cone and sphere (left), selected tiles for the high quality foreground (right).

and aperture α, as shown in Figure 2b. In this paper α refers to the *foveal angle* and it is a key parameter of the system. The value of this angle needs to be carefully selected so that the background region falls at the periphery of a user's field of vision, making the lower quality not noticeable, or at least not disturbing, for the user. To simplify the computations to identify the tiles intercepted by the cone, a number of regularly spaced radii contained in a given selected α has been used, as shown in Figure 3a. In this example, as for the rest of the paper, a 360° video with equirectangular projection are used. The selected tiles for the foreground region are shown in Figure 3b.

Concerning the background region, it is important to remark that HEVC tiling introduces visual artifacts at the edges of each tile, when videos are encoded at low bitrates. To avoid these effects, potentially impacting in a negative way the users' experience, we decided not to display low QP tiles in the background. Instead a non tiled video is used for the background and it is sent separately from the foreground tiles, similarly to [23]. This implies that the client needs two decoders and it simply overlays the foreground tiles at the appropriate locations over each full-sphere wide background frame. The background video can be of either lower bitrate or lower resolution as compared to the foreground. Even if this approach adds decoding complexity and a portion of each background frame is "wasted" under the tiled foreground, it is likely to improve QoE at no extra storage cost. Following this scheme, both the tiled HEVC video for the foreground and the un-tiled one for the background can be prepared in advance at content provider servers.

5.2 Quality switching problem

The previous section did not address how rapidly a new quality of an individual tile can be activated. This is crucial for maintaining the user's experience after large saccadic movements. One limitation of the HEVC codec is that only I-frames can be used as RAPs. Thus, if the client requests a new tile to be added to the foreground and the upcoming frame is not an I-frame, the response will not come immediately. An illustration of the "random access problem" is shown in

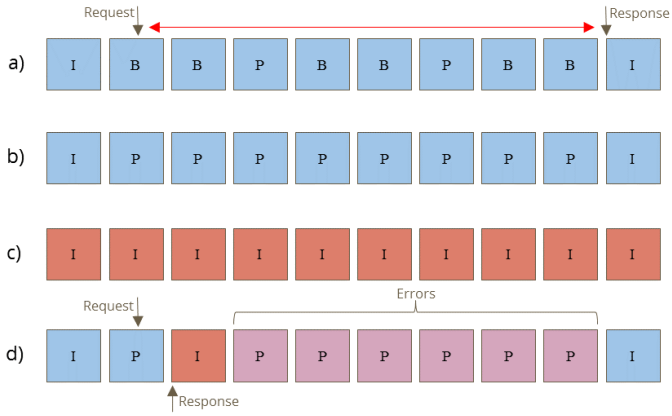


Fig. 4. a) Random access problem with standard HEVC video. b) Video encoded with I-frames and P-frames depending on the previous frame. c) I-frames only video, d) Approach with I-frame insertion and drift.

Figure 4a, where the client request arrives at the second frame, on a B tile. In this case the client needs to wait until the next I-frame to switch quality. The red arrow in Figure 4a represents this activation delay. A simple alternative, that allows immediate adaption of tile quality to new eye-gaze positions, is to encode the foreground tiles using only I-frames, as shown in Figure 4c. However, this approach would lead to much smaller compression ratios, dramatically increasing the bitrate of the video. Previous work on streaming for VR tried to mitigate this issue at the encoder, for example by forcing the insertion of random access pictures (I-frames) more frequently, e.g. every one [25] or two [14] seconds. While in our case frame activation delays of this magnitude are likely to disrupt the user experience, the aforementioned approaches focused on viewport centric solutions, where changes are triggered by head movements. These are typically both less frequent and much slower than saccadic movements, which can reach speeds up to 600°/s [36], implying that users' gaze can easily transition through the entire Field Of View (FOV) in a viewport in much less than a second. An alternative stream switching mechanism, based on distributed source coding, has been proposed and evaluated by Cheung et. al. [15] [12]. In order to effectively merge multiple SI frames, which are required to reconstruct the landing frame, they proposed a piecewise constant function. Their experimental results show some coding gains over H.264, with a reduction in decoder complexity. Essentially their work further extends the concepts of SP and SI frames that have been presented by Karczewicz and Kurceren in [24]. Both approaches are designed to support frame switching while eliminating decoding drift; however, from the user experience point of view, SI frames can be detectable to the human eye because of the processes of transformation and quantization performed. Moreover, these approaches are not supported in HEVC, thus the fact that existing decoders cannot be used is major drawback for the practical applications targeted by our work.

5.2.1 Proposed solution: I-frame insertion

In our proposed approach, instead, two versions of the source video are encoded and used for transmission, for a target bitrate:

- one using only I-frames (see Figure 4c) and
- one using I-frames and P-frames that depend only on the immediately previous frame (see Figure 4b).

The actual video delivered to the user is a mix between the aforementioned two versions. In particular, when a client, due to a change in user's eye-gaze, requests a new foreground tile and only a P-frame is available as the next frame, it is served the corresponding tile from the video composed of only I-frames. This approach is depicted in Figure 4d, where the third frame in the main video (Figure 4b) is replaced by the third frame of the video with only I-frames (in Figure 4c). This I-frame is then used to decode the next P-frame in the main video. However, since this was encoded using the replaced P-frame, its decoded output is certainly different from the corresponding picture in the main

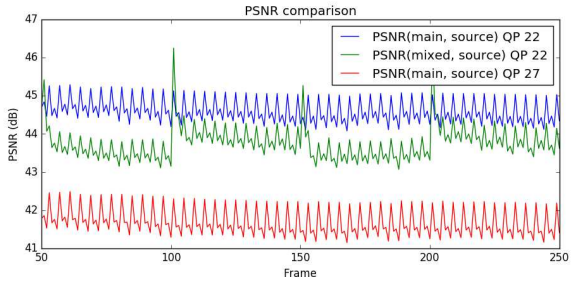


Fig. 5. Example of PSNR variations over consecutive frames for a p-video $PSNR(p(f),s(f))$, with QP 22 (top blue line) and 27 (bottom red line), and an m-video $PSNR(m(f),s(f))$ for QP 22 (middle green line).

video. The coding drift error resulting from this *I-frame insertion* is shown in Figure 4d, where the P-frames after the insertion are depicted with a different colour. This approach is similar to the S-frame concept adopted as reference in [24], however in this paper it is applied to each individual tile. To our knowledge, this has not been done before.

Decoding errors such as these can cause a wide range of visual artifacts, potentially disrupting the user experience. An important question is the extent and how noticeable these errors are to the end users. Further, this solution requires storing two videos, one of which is encoded only with I-frames. While this increases the number of bits stored at the content provider side, a key question is the extent of this increase. Moreover, since this method is likely to substantially reduce datarates over that of standard streaming, it is relevant to assess whether the benefits from capacity savings exceeds the potentially increased storage costs.

5.3 Experimental assessment procedure

Eight[1] videos, used for testing in standardization activities, were selected for our experiments. All videos have 4K resolution (3840x2160 or 4096x2160) and 10-bit color, with duration ranging between 5 and 12 seconds and either 50 or 60 fps framerate. All clips have been encoded using the Kvazaar encoder [33] with QP values of 22, 27, 32, and 37 and for different tiling layouts including 1x1, 3x3, 5x5, and 15x10. Tiling was performed by selecting Kvazaar's "motion constrained" settings to guarantee the possibility to decode each tile independently from the others. Also the "temporal motion vector prediction" feature of HEVC was deactivated, since experimentally we noticed that this setting reduced significantly the extent of the visual artifacts introduced by tile insertion. For all parameter combinations, each source video (**s-video**) has been encoded to generate three different versions[2]:

- an **i-video**, composed only of I-frames, similar to Figure 4c,
- a **p-video**, with a single I-frame followed by P-frames, with similar structure as the one in Figure 4b, and
- an **m-video**, obtained by arbitrarily inserting an I-frame every second in the corresponding **p-video**. This has a frame structure similar to that shown in Figure 4d.

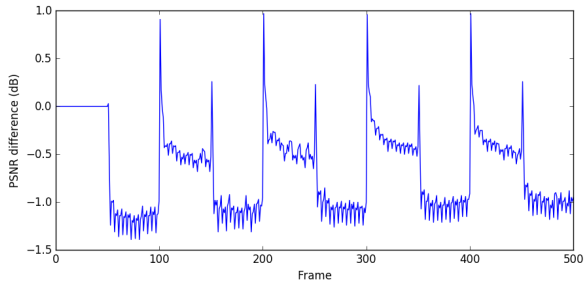
These periodic I-frames insertions characterizing the m-video have been included to scale up the number of data points available for analyzing the impact on video quality caused by drift following a frame insertion. Both visual inspection with several users and PSNR have been selected to assess performances. PSNR is a standard metric, commonly used to evaluate the quality of lossy codecs. Hereafter, the notation $PSNR(A(f),B(f))$ refers to the PSNR at frame f of video A, when video B is selected as the reference.

5.4 Results

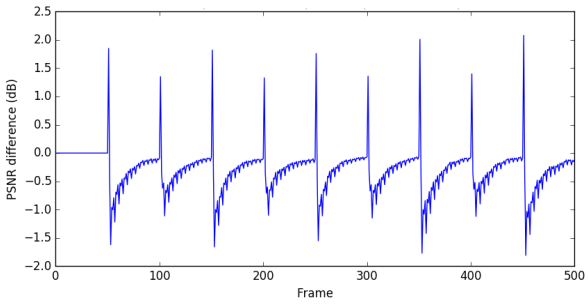
As expected, the average $\overline{PSNR(m,s)}$ over all frames, comparing an m-video to its source video s, is always lower than the corresponding av-

[1]The subset of tested videos, whose license allows redistribution, is available at [1] for different tiling and QP configurations.

[2]A total of 384 videos were prepared, i.e. 3 different encodings for 4 QP levels, 4 tiling cases, and 8 sources.



(a) The PSNR difference $\Delta_{m,p}(f)$ remains bounded



(b) The PSNR difference $\Delta_{m,p}(f)$ self-corrects

Fig. 6. Examples of different behaviors, corresponding to different source sequences, for the PSNR difference $\Delta_{m,p}(f)$ after frame insertion.

erage $\overline{PSNR(p,s)}$, comparing the p-video version with the same source. However, the interesting result is that this additional degradation is exceptionally modest for all considered video sources, ranging between 0.016 dB and 1.317 dB, for low QP values, i.e. with PSNR in the 35-45 dB range. Moreover, the worse performing frames across all QPs and tiling configurations lead to performance losses between 1.55 dB to 2.86 dB, depending on the specific video.

One of the most important result of this paper is the experimental validation that the coding drift caused by I-frame insertion at the tile level does not grow unbounded over consecutive frames. Unsurprisingly, the PSNR difference $\Delta_{m,p}(f) = PSNR(m(f),s(f)) - PSNR(p(f),s(f))$ peaks positively at the insertion frame, but then drops sharply, (see Figures 6a-b). After an I-frame insertion two types of behavior have been recorded throughout all the investigated videos:

1. the PSNR difference $\Delta_{m,p}(f)$ *remains bounded*, oscillating around a constant value, as shown in Figure 6a,

2. the PSNR difference *self-corrects*, as shown in Figure 6b.

It is important to remark that the performance shown in Figure 6a corresponds to one of the videos with the worst average PSNR decrease for all QP and tiling combinations considered in our experiments. Moreover, several users visually inspected the mixed videos and found that they could not detect any visual artifacts at QP 22. At QP 27 some minor artifacts could be predominantly noticed in only one video (the same one shown in Figure 6a), while three other sequences had very limited errors, spatially localized around a few tiles. In most high QP cases, i.e. QP 32 and 37, I-frame insertions lead to detectable errors, however these cases also show a number of visual artifacts introduced by tiling itself. Thus, we can conclude that QP 22 is suitable for tile replacement for all considered videos, while values between 22 and 27 are also acceptable for most videos. The per frame PSNRs for the video with the worst degradation recorded across our experiments are illustrated in Figure 5. Two cases are displayed for the main video p, QP 22 and 27. As it is clearly visible, the mixed video m is performing in between the main video cases and it is very close to the curve with the lowest QP. Similar performance occurred with all investigated videos, thus, based on the collected experimental observations (see [1]), it can be safely concluded that tile replacement causes less degradation than increasing QP by 5 points, at least when operating in low QP regions.

Tiles	Storage factor ν (P-frame)			Storage factor θ (I-frame)		
	Min	Mean	Max	Min	Mean	Max
3x3	0.9009	1.0092	1.0989	1.1881	2.9767	5.3491
5x5	0.9258	1.0732	1.2150	1.1900	2.9895	5.3645
10x5	0.9385	1.1375	1.3401	1.1924	3.0080	5.3865
8x8	0.9392	1.1804	1.4317	1.1931	3.0150	5.3930
10x10	0.9422	1.2506	1.5779	1.1964	3.0345	5.4146
15x10	0.9461	1.319	1.7238	1.2003	3.0600	5.4452

Table 1. Bounds on the increase of storage factors ν and θ for various tiling configurations and across all considered videos.

5.5 Storage requirements

Based on the results presented in the previous section we know that when the targeted foreground quality is high enough, I-frame insertion allows us to control the video quality in individual video tiles, thus achieving fast spatial random access with minimal visual degradation. While in principle this could enable substantial bandwidth savings when distributing content to the end-users, it increases storage costs at the server size. The focus of this section is to quantify the overall storage costs for supporting our proposed content delivery.

As described in Sections 5.1-5.2, three different video versions need to be prepared and stored at the server to support our content delivery operations: (1) a tiled version of the video that uses only I-frames (i-video), (2) a tiled version that uses I-frames and P-frames, the latter only depend on the immediately preceding frames (p-video) and (3) an un-tiled low quality version of the video to be used in the background.

To assess storage performances, the cost of each video version was compared to a *reference* video encoded using Kvazaar's [33] "very slow" preset, to get maximum compression. The same preset and default encoder settings were also used to encode both i-videos and p-videos, with default QP of 22 for I-frames and 23-24 for P-frames.

The *storage factor* ν is defined as the ratio between the file sizes associated with p-videos and their corresponding reference video. This is illustrated in Table 1 for different tiling settings. There, the first three columns illustrate the minimum, mean, and maximum values, recorded over all processed videos. The results show that storage costs are proportional to the number of tiles but less than double the size of the reference video, even for the worst performing case.

In the same table we also report the corresponding results for the *storage factor* θ associated with i-videos. The trend in this case is only a slight increase with the number of tiles, almost constant for each tested video. A wide range of values were recorded in the tested cases, with increases between 1.2 and 5.5 times.

The current system description does not constraint the quality of the background, nor if it is encoded at a different bitrate or at a different resolution and then upscaled when rendering. However, the storage requirements of the background differ by an order of magnitude from the two aforementioned cost components of the foreground region. To estimate the background storage, 20 4K videos where encoded in five lower resolutions (426x240, 640x360, 960x540, 1280x720, and 1920x1080) using Kvazaar, without tiles or any other constraints. The resulting file sizes showed that, on average, all five background videos together have a file size about 75% of their reference. Moreover, many background versions are probably not needed in practice, thus it is possible to further reduce background storage costs.

Figure 7 provides insight into how the *total storage* for the foreground (p-video and i-video storage) varies as function of the video bitrates. It includes data from 20 4K videos, all with 15x10 tiles. The results show that the required storage is essentially inversely proportional to the bitrates of the original videos. In the worse case, the total storage increases by 30.16 times the size of the reference video; however, the median across all considered cases is only 5.36 times. This makes the storage needed by the system competitive with some of the viewport-aware solutions. In particular, the storage requirements of Facebook's pyramid encoding [25] can be estimated to be about 6 times the original videos. In [25] it is mentioned that 5 different bitrates are used to encode 30 different viewports, effectively requiring 150 video versions to be stored. Only considering the 30 viewport versions corresponding to the highest available bitrate, and assuming an 80% file size

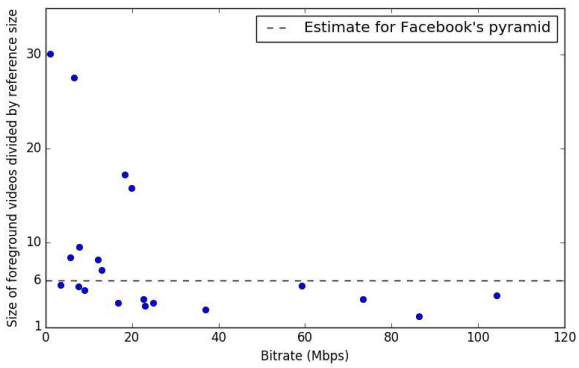


Fig. 7. Total storage cost increase for gaze-aware streaming. All videos have 15x10 tiling and are ordered based on their original bitrates.

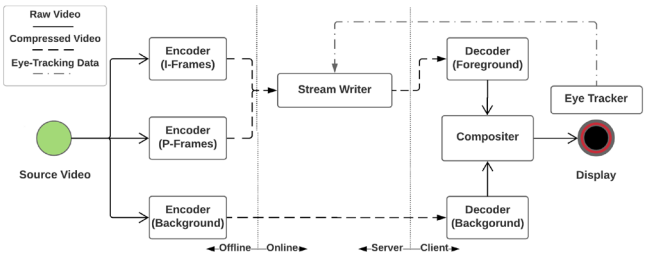


Fig. 8. Schematic of the proposed encoding/decoding pipeline.

reduction per viewport [25], the necessary storage should correspond to at least 6 times that of the original video. This storage factor is shown for reference in Figure 7, with a dashed line. Our proposed solution seems to outperform the estimated storage requirements for Facebook's pyramid solution, at least for videos in the medium-high bitrate region. An important observation is that *if storage is a limiting factor, low bitrate videos should not be served according to our proposed model*. In fact, some of the low bitrate videos used in this comparison have such small file sizes that they could be directly served without using gaze-aware content provision.

5.6 Proposed pipeline

A summary of the complete encoding/decoding pipeline for supporting gaze-aware streaming is shown in Figure 8. There the different entities with their individual units are represented, together with their interconnections and exchanged data types. At the server side 360° equirectangular videos are pre-encoded offline into low quality background, high quality tiled I-frames, and high quality tiled P-frames, as discussed in Section 5.2.1. When streaming is initiated, a *stream writer* in the foveal server uses the latest available eye-gaze measurements to select the appropriate tiles to compose the foreground video. This is transported to the client together with the background video. Once both are received and decoded, a *Compositer* overlays the high quality tiles over the background frame, to finalize the complete frame to be rendered in the HMD.

6 User Experiments

In order to explore the trade-off between QoE and the network requirements for gaze-aware streaming, a testbed that implements our proposed solution was developed and used in extensive testing with real users. In this section we describe its components and present the obtained results.

6.1 Testbed

To prepare the content to be used in the tests, Equi-Rectangular Projection (ERP) formatted videos are pre-processed and tiled using *Kvazaar* and *ffmpeg* open-source tools. Starting with a YUV raw video, tiled HEVC streams and then mp4 files are created. The resulting mp4 files

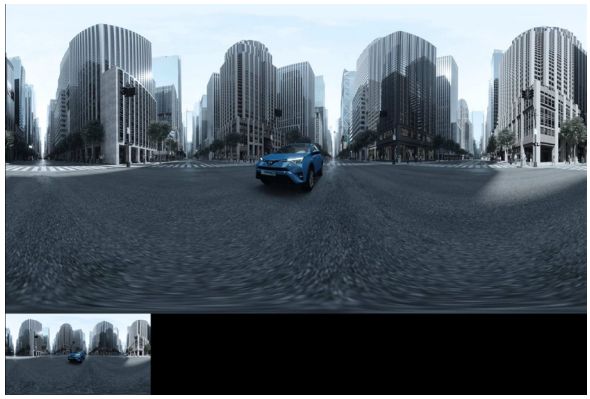


Fig. 9. Example of frame stacking for the *Toyota* video. The top image has 4K resolution while the bottom one is in 540p.

are then converted to MPEG-DASH format using *GPAC* open-source software, as described by Concolato et al. in [13]. This software provides support for creating tiled videos and playing them using DASH. While it can create DASH streams up to one frame per segment, it is unable to play configurations below 5-6 frames per segment. Thus, per frame DASH segments are created but not used for transmission: they are only used for estimating more realistic network bandwidth requirements, at the end of each user test. Since the experimental results concerning I-frame insertion clearly showed that users cannot detect the visual errors introduced by inserting low QP I-frames (see Section 5.4), we focused on a testbed that would explore the relationship between achievable QoE and network latency, rather than implementing a video decoder and transport mechanisms. In our view, the user experience assessment is the key pre-requisite for any practical implementation of gaze-aware streaming and it is a major contribution of our work.

In the testbed, I-frame **tile insertion** is implemented by using a version of the video in which each frame is composed from two frames of different resolutions *stacked* together, one from a low resolution version and one from the full 4K video. An example of this approach is shown in Figure 9 and several stacked videos are available at [1]. In this approach, the frame areas corresponding to the positions of foreground tiles are filled with the corresponding content from the high resolution part of the stacked video, while the low quality background is obtained from an upscaled version of the low quality frame. This has been implemented using the open-source toolbox *OpenFrameworks*, which provides an API for OpenGL programming.

An important consideration involves the specific shape and position of the tiles. Since tiling has been applied to equirectangular videos, the distribution of the tiles on the surface of the projection sphere is uneven, with fewer tiles activated when looking along the horizon than towards the poles, for a given foveal angle. This could lead to pronounced fluctuation of required datarate as function of gaze variations, but extreme variations in instantaneous bitrate have not been noticed in our experiments. This can be explained by the fact that very few users focus on the poles for prolonged amounts of time and that most of the variations in the videos are spatially located in areas close to the horizon, essentially leading to larger file sizes for tiles closer to the horizon than towards the poles[3]. Even so, it is worth exploring alternative approaches to this spatial placement of tiles, e.g. to reduce video bitrate fluctuations or to dynamically adapt the shape and position of tiles to specific objects in the images. However, this is outside the scope of the paper and constitutes a relevant direction for future works.

The testbed uses an early prototype of Tobii's *VR4 for Vive Developer Kit*, featuring an HTC Vive with embedded eye-trackers. This HMD has a refresh rate of 90Hz, while the eye-trackers operates at 120Hz. The *OpenVR API* is used to program the HMD, while *Tobii Stream Engine SDK* is used to access eye-gaze data. Detailed information on the various software components used in the testbed is available at [1].

[3]In our set of videos the average size of tiles at the horizon is about 2-2.5 times that of the tiles located in proximity of the poles.

To identify the tiles to be included in the foreground, the procedure described in Section 5.1 and illustrated in Figure 3 was utilized, considering the specific values of *foveal angle* α being investigated.

6.2 System parameters

As mentioned in Section 2.1, the end-to-end latency Δ_t is one of the most critical parameters constraining the performances of gaze-aware streaming. Out of the four system latency components previously mentioned, the eye-tracker latency *EL* and the display latency *DL* components of the testbed are based upon Tobii's VR4 HMD. Specifically, *EL* is approximately equal to 10 ms [4], representing the time needed by the eye tracker to process the acquired eye images and output the corresponding eye-gaze estimates. The HTC Vive has a *DL* of about 22 ms, corresponding to the duration of the two frames at 90 Hz (11.11 ms each), that are needed to render and send the image to the HMD panels [34]. To explore the network and QoE performances of gaze-aware streaming under various network configurations, a parameter for emulating different *RTT*s has been included in our testbed. This has been used to control a queue of eye-gaze measurements from the eye-tracker, making them available with a delay equal to the RTT. Values in the range 10-50 ms have been selected to represent realistic latencies for both 4G and 5G networks. These delays represent the sum of the "over-the-air" and "backend" latency components that we assume to occur over the communication links between user devices and content provider caches. It can be pointed out that 5G networks are being designed to target latencies over the air in the order of 1 ms [6].

One of the major differences between our testbed and the actual pipeline for gaze-aware streaming presented in Figure 8 is that the foreground and background streams are not decoded independently in the testbed. Instead, the "Compositer" operates on the aforementioned stacked frames. The compositer needs (a) to decode a stacked frame, (b) to separate its foreground and background, (c) to upscale the background image into 4K resolution, (d) to select the specific portions of the foreground corresponding to the activated tiles, and (e) to copy those onto the upscaled background image to obtain the frame to be rendered. To quantify the specific costs and latencies associated with the compositer, the *Frame Timing* software tool, available with Steam VR, has been used to record the performances of GPU and CPU during playout. In this way, the processing latency *PL* has been estimated to be approximately 16 ms. While most of this latency is introduced by various image processing operations, a non negligible amount is due to the software platform used (OpenFrameworks and OpenVR). To validate the latter, a set of experiments has been performed, where our selected videos has been played using both the popular "Virtual Desktop" video player app on Steam and our testbed, the latter with foveated content provisioning disabled. By comparing the logs provided by the *Frame Timing* software we have been able to estimate an average increase in total rendering time of 4 ms when using our testbed. This means that about 4 ms of the value estimated for *PL* is introduced by the high level development environment used for implementing the testbed, hence it could be eliminated in a more optimized realization.

Taking into account all aforementioned components, the end-to-end latency can be computed as $\Delta_t = EL + RTT + PL + DL = 48\text{ ms} + RTT$, i.e. between 58 and 98 ms. While this delay is important in the end-user experience, it does not define the achievable frame rate. Specifically, the frame rate is limited by the value of *PL*=16 ms, corresponding in our case to approximately 60 fps. While this represents the upper bound on the currently achievable performance with our testbed, it does not seem to be an intrinsic limitation of the technology *per se*.

6.2.1 Foveated parameter space

The foveal angle α parameter has been explored in user testing. The subset of values chosen for the experiments were 20°, 30°, 40°, and 50° degrees; with the larger values than the actual foveal region introduced as "safety margins", to intercept on the foreground tiles adjacent saccadic movements even before these are registered by the system (as would be necessary in the case of large end-to-end system latencies).

The video quality chosen for the background region can have a substantial impact on both QoE and overall bandwidth costs. In fact, for high system latencies, our gaze-aware solution can be too slow to appropriately react to users' eye-gaze variations, thus presenting users with content from the background region for a number of frames, after each saccadic movement. For large RTTs, the lower the resolution of the background the greater the probability that users will be exposed to significant fluctuations in video quality, therefore experiencing lower QoE levels. In order to explore these potential effects, while the foreground has been fixed to a full 4K resolution in all cases, different background qualities (Q_b) have been chosen, including 240p, 360p, 540p, 720p, and 1080p.

240p, 20°	360p, 20°	540p, 20°	720p, 20°	1080p, 20°
240p, 30°	360p, 30°	540p, 30°	720p, 30°	1080p, 30°
240p, 40°	360p, 40°	540p, 40°	720p, 40°	1080p, 40°
240p, 50°	360p, 50°	540p, 50°	720p, 20°	1080p, 50°

Table 2. Representation of the foveated parameter space $\{Q_b, \alpha\}$ considered in the user experiments. RTT cases include 10, 30, and 50ms.

6.3 Testing methodology

Three different videos with a duration of 1 min were chosen for the user testing: **Factory** - a slowly changing highly textured video, **Toyota** - a fast paced single point of focus video, and **Elephants** - a multi focal video with sharp contrast features. All three videos are available at [1].

As suggested in ITU-T P.910 [22], a full reference Degradation Category Rating (DCR) scale was used for testing. A grade of 5 indicates that the user cannot distinguish the original and the impaired video. A grade of 4 indicates that some differences are detectable, but they are not annoying. A grade of 3 indicates that the modified version is perceived as slightly annoying. Finally, grades 2 and 1 indicate annoying and very annoying user experiences, respectively.

For each user one of the aforementioned videos was chosen and a *representative* image for that video was first rendered on the HMD. Since many participants were using VR for the first time, each of them was shown this 360° image at full 4K for a period of 30s, to familiarize them with both the headset and the VR experience. Thereafter, starting from the highest RTT (50 ms in this case), all acceptable settings (grade 4 or 5) for each user were identified by quickly traversing the foveated parameter space shown in Table 2. This exploration using representative still images was introduced to quickly obtain an indication of values of the foveal angle and background quality that might deliver acceptable QoE to a user, when subsequently displaying the actual video. Based on results of this image-based preliminary exploration, a set of candidate parameters for the videos were identified for each RTT value and used for the actual video testing. For each RTT case, the testing was terminated once all background quality and foveal angle settings for achieving a 4 or 5 grade were identified. In particular, as soon as a couple $\{Q_b, \alpha\}$ in the parameter space was assigned a grade of 3, all parameter combinations with smaller or equal foveal angle and lower or equal background quality were immediately discarded from the experiment for that RTT case. This is because those operating points at lower resolution backgrounds or smaller foveal angles should only achieve inferior or equal QoE than 3, which is below our target grade (≥ 4). This experimental design dramatically reduced the exploration of the entire parameter space, by lowering the test time of a complete experiment with a video from a theoretical duration of 1 hour to about 20 minutes. One hour is the minimum theoretical testing duration since the parameter space itself consists of 60 cases to be tested (5 Q_bs, 3 RTTs and 4 αs), each with video duration of a minute.

6.4 Results

Testing involved 20 users with normal 20/20 vision without correction, with diverse background and gender, and ages ranging between 16 and 55 years old. All were invited to our office and in exchange for their time they received a movie ticket worth 120 SEK.

Fig 10 shows the average bandwidth utilization achieved by gaze-aware streaming when serving with various parameter settings the three different 360° videos chosen for user testing. The bandwidth utilization is computed for each experiment by dividing the total number of bits needed by gaze-aware streaming by the file size in bits of the reference

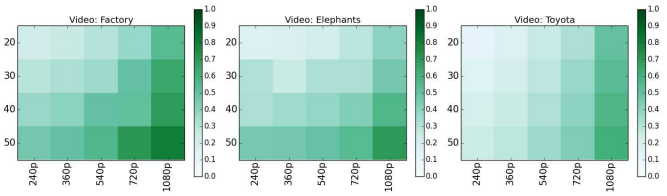
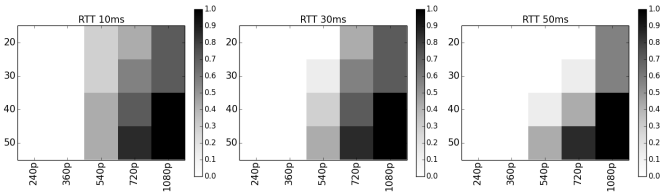
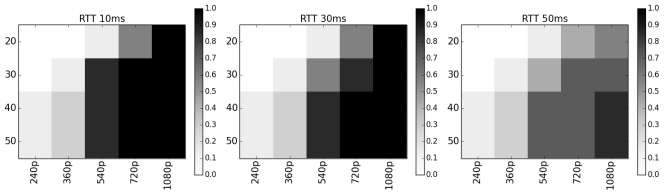


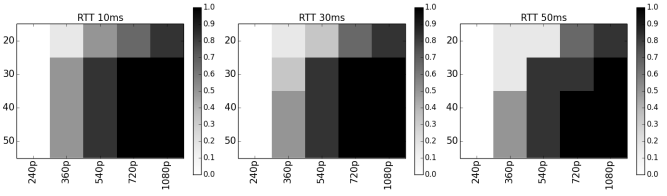
Fig. 10. Average bandwidth utilization as compared to the equirectangular reference case, for the various considered parameters and videos.



(a) Factory video (see [1]).



(b) Elephants video (see [1])



(c) Toyota video (see [1])

Fig. 11. Quality of experience expressed as fraction of users satisfied by a given parameter setting, for different videos and network RTTs.

video (equirectangular at full 4K). The horizontal axis in each subplot lists the different background qualities while the vertical axis lists the different foveal angles α. Each data point represents the average of all bandwidth savings obtained with the test subjects for that parameter combination. Due to our testing methodology not all combinations in the parameter space have been tested for some users. For those cases we used instead eye traces obtained by various test participants for the same video and same foveal angle but with different background quality. This approximation, including synthetic traces from real users, provides a good estimation for the bandwidth utilization in the missing cases. It has been validated using our experimental data, by comparing for different users the ratios between the total bytes required for the foreground and the size of the reference video, when the same video was served with identical foveal angle but different background qualities. The results showed that performances varied within only $\pm 4\%$, with even smaller spread for the cases with narrower foveal angles[4].

The results in Figure 10 show that the bandwidth utilization varies with the different videos served. Moreover, when looking at parameter combinations achieving similar order of magnitude in bandwidth utilization, i.e. specific regions with the same color gradation, it is possible to derive an equivalence for which lowering α by 10° corresponds to the same utilization as increasing the background quality by 1 or 2 levels (e.g. from 240p to 360p or 540p).

Figure 11 presents the users' QoE for the various points in the parameter space. There, the probability of the user evaluating a setting as acceptable (DCR rating 4 or 5) was introduced as a measure for computing the average QoE. For each user, the settings leading to acceptable quality were marked as 1 and those with lower grades were set to 0. The 1s and 0s supplied for each setting were summed up and divided by the number of test runs for that setting. The resulting values are numbers between 0 (represented by white) and 1 (represented by black) signifying the fraction of users in our panel finding those settings acceptable (with QoE grade ≥ 4). For each of the three videos, three RTT values were considered. It can be observed that as RTT increases, the acceptable region of points in the parameter space shifts towards the bottom right. When RTT increases the system is less responsive, thus when the user's eye-gaze changes it is more likely to end up in the frame background. By increasing the background quality this effect is somewhat mitigated, hence improving the overall experience. By widening the foveal angle instead, there is an increase in the probability that saccadic movements will land users' eye-gazes in locations already served by foreground tiles.

[4]This does not imply that similar QoE levels were experienced by the users, only that foreground costs, for a given video and α, are close across the users.

Video	SLA (%)	RTT 10 ms	RTT 30 ms	RTT 50 ms
Factory	*0.5*	50%	50%	46%
	0.7	46%	46%	31%
	0.9	31%	31%	31%
Elephants	*0.5*	71%	71%	67%
	0.7	68%	67%	67%
	0.9	67%	67%	40%
Toyota	*0.5*	79%	71%	71%
	0.7	71%	71%	71%
	0.9	62%	62%	60%

Table 3. Maximum Bandwidth Savings (%) in different RTT conditions and with foveal parameters set to achieve, at the lowest bandwidth cost, a target probability (SLA) for a user in the system to experience QoE≥ 4.

Video	Minimum	Maximum
Factory	25.66%	70.28%
Elephants	43.15%	81.34%
Toyota	48.46%	83.15%

Table 4. Bandwidth savings corresponding to parameter settings delivering acceptable QoE to all users (Minimum) or at least a user (Maximum).

While the videos Elephants and Toyota have similar performances, the case of Factory presents a very different QoE behavior. Factory has a lot of sharp features, in particular extensive overlays of computer generated white grids over black background and large text boxes, which often lead to detectable artifacts at the periphery of the users' field of view. This has been confirmed by the users during the experiments and it is part of the large body of subjective data collected during the experiments. Essentially, the portions of these large grids that are displayed in the background region are extremely down-sampled, leading to substantial variations in the color space. When users explore these large objects, they experience some color changing effects that attract their attention and degrade their overall experience.

By jointly analyzing the results for the bandwidth utilization in Figure 10 and the QoE regions in Figure 11, it is possible to quantify the bandwidth savings achievable by gaze-aware streaming for different Service Level Agreements (SLAs) and various network RTTs. In this case, we assume that an SLA is defined based upon the percentage of satisfied users, i.e. with a QoE grade ≥ 4, while the bandwidth savings are simply the complementary percentages of the utilizations in Figure 10. Table 3 shows the maximum bandwidth savings (%) achievable for different RTTs and for three SLA targets: half of the users (0.5), seventy percent of all users (0.7), and ninety percent of the entire tested panel (0.9). Except for the RTT settings associated with 5G (10 ms), the Toyota video shows very consistent performances across SLAs. However for RTT = 10 ms, 50% of the users can be served with settings leading to 79% bandwidth savings. A different behavior is instead seen for the case of the Elephants video, where for the worst RTT (50 ms) 27 percentage points of bandwidth savings are lost when selecting parameters settings satisfying 90% of the users, instead of aiming for an SLA of 0.7. Finally, the video Factory shows almost 20 percentage point difference in bandwidth savings between tuning the parameters for SLA = 0.5 and SLA = 0.9. This specific video polarizes our user

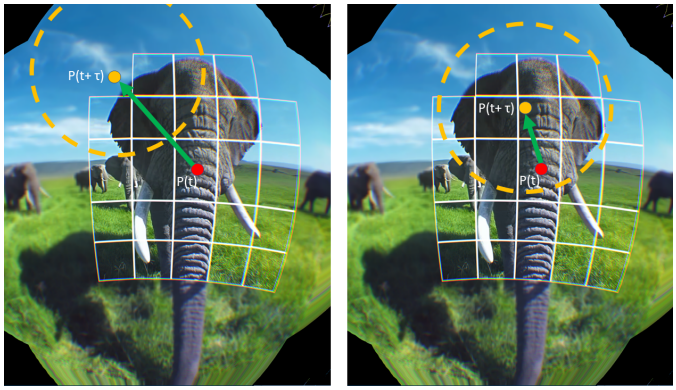


Fig. 12. *Miss* (left) and *Hit* (right) examples of saccadic movements.

panel: a subset of participants seem not to be bothered by the color changes at the periphery of their field of view, while another group is much more sensitive to these effects. In this respect, a clear result emerging from this study is that users have very different viewing patterns, very different visual acuity, and quality requirements. By looking at numbers in Table 4, we notice a dramatic variation between the performances associated with settings delivering acceptable quality (≥ 4) to at least a user, i.e. Maximum, and those required to satisfy all users with high grades, i.e. Minimum. In conclusion, designing a solution that is capable of high degrees of personalization is likely to achieve significantly better overall bandwidth saving performances. This could be achieved, for example, by creating a procedure for initializing user profiles, by providing users with an initial set of test videos to assess individual preferences and requirements.

7 Discussion

The previous section focused on assessing the levels of QoE perceived by individual users, when exposed to a number of videos with specific parameter settings. This section discusses the possibility of defining a simple indicator of user experience that is general enough to be applicable to different videos and video types while being robust across different users. An important pre-requisite for such an indicator is that it must be representative of the components of the human vision that have a greater influence on the formation of QoE. In our specific case, the perception of good or bad video quality is clearly influenced by the total system latency Δ_t. However, this is not the only factor, since our results clearly show that negative effects of large latencies on QoE can be compensated for by increasing either foveal angle or the background resolution (see Fig. 11). These results also imply that, for a given background quality, there are combinations of foveal angle and RTT delivering identical levels of QoE and that these are positively correlated, meaning that for increasing latency a wider foveal angle is needed to deliver the same user experience. Since user perceived quality decreases when users are exposed to lower resolution, one possibility would be to construct an indicator accounting for how often users are exposed to background quality and their tolerance to the various background video resolutions. Thus, a very simple approach could consider some of the statistics related to "hit" and "miss" events on the video foreground, throughout a video's playtime. Examples of hit and miss events are shown in Figure 12. There, two consecutive data points in a user's *scan path* are illustrated, $P(t+\tau)$ and $P(t)$, separated in time by τ ms. Since the foreground is updated at $t+\Delta_t$, a miss is defined as a saccadic movement that lands at $P(t+\tau)$ on the background region, when $\tau < \Delta_t$ (Figure 12 left). A hit is instead recorded for all the eye movements landing on high quality tiles composing the frame at that time instant. Based on these definitions, the hit probability $P_{hit}(Q_b)$ for a specific experiment with background quality Q_b can be computed as the percentage of hits events over the total number of eye-gaze measurements produced during the playtime. This quantity seems to closely match the aforementioned properties of the human visual system, since it does not explicitly depend on the individual values of RTT or foveal angle or on the specific scanning behavior of an individual user, but only on their combined effect. Processing the data

	360p	540p	720p
PCTL(90)	0.988	0.984	0.979
MEAN	0.982	0.973	0.968
PCTL(10)	0.972	0.958	0.949

Table 5. Summary of statistical properties of $P_{hit}(Q_b)$ for QoE≥ 4

collected in our experiments we have verified that, for a given value Q_b, $P_{hit}(Q_b)$ is positively correlated with the recorded QoE grades. This has been done for all the available data points in which a user has at least two entries with background quality Q_b. Note that due to our specific experimental design (see Section 6.3) not all the points in the parameter space have been tested with all users. To assess the existence of a threshold probability $P^*_{hit}(Q_b)$ for achieving good QoE and to describe its behavior for different Q_b values we have considered the subset, across all users and videos, of available $P_{hit}(Q_b)$ entries for which QoE≥ 4. The mean, 90th percentile, and 10th percentile of the hit probabilities leading to QoE≥ 4 are shown in Table 5 for $Q_b = \{360\text{p},540\text{p},720\text{p}\}$. In accordance with our intuition, the results show that the required hit probability decreases with increasing Q_b. Further, the dispersion across users and videos seems rather limited, given the modest distance between the 10th and 90th percentile values, especially for medium-low Q_bs. While these results provide an initial validation for this indicator, the complete characterization of its performances across different users and videos is outside the scope of this paper and is left for future investigations.

8 Conclusions

This paper presented gaze-aware streaming, a novel content delivery method for enabling future mobile video experiences in VR. Our approach exploits information from connected eye-trackers, expected to be embedded in the next generation of HMDs, to limit the provisioning of high video quality to the areas in proximity of users' fixations. The goal of gaze-aware streaming is to substantially reduce the bandwidth requirements for supporting video experiences while delivering high levels of user perceived quality. Pre-requisites to achieve these results are (1) mechanisms that can cope with different degrees of latency in the system and (2) solutions that support rapid spatial video adaptation within the frame, without requiring increases in the videos' bitrates.

A method based on I-frame insertion has been applied for the first time on tiled videos and it has been shown to be capable of allowing instantaneous video quality adaptation in specific portions of a video frame. The proposed method exploits built-in properties of HEVC encoders and while it introduces a moderate amount of coding drift (less than 1 db loss in PSNR), this is un-detectable by the users at the low QP levels (22-27) at which the system is expected to operate. Further, since multiple representations of the same video need to be stored at the server side, we explored the potential impact on content storage and found that our proposed solution has lower storage requirements than current viewport-based methods, especially for videos with medium-high bitrates. In those cases, only about 5 times the size of a equirectangular 360° 4K video is required to store all the different representations needed for gaze-aware provision. These results fulfil some of the most critical pre-requisites for the practical feasibility of gaze-aware streaming. A testbed implementing our envisioned gaze-aware streaming solution and including a prototype HMD with eye-tracker has been developed and tested with real users. The studies quantified the bandwidth savings achievable by the proposed approach and characterized the relationships between the various systems parameters, quality of experience and network latency. The results showed that up to 83% less bandwidth is needed to deliver high levels of quality of experience to the users, as compared to equirectangular 4K encoding. While the results vary for different users and different videos, bandwidth savings on the order of 60%-80% have been recorded for the majority of the users, with RRT ranges expected for upcoming 5G networks. For roughly half of the tested users similar bandwidth savings are already achievable for latency values that are typical of current 4G networks. In this respect, the personalization of foveal parameters and the definition of user profiles seem to be two promising ways forward for exploiting the differences in visual acuity within the user population.

9 Future works

While the results achieved with the current testbed are very promising, we are currently working on implementing an entire pipeline of foveated streaming, including the encoder/decoder presented in this paper together with a transport protocol. For the latter, we are assessing the possibility of using a modified version of the Spatial Relation Description (SRD) enhancement to MPEG-DASH.

Since our work has a strong focus on how VR services can be deployed in future 5G networks, a key direction for our research includes exploring the definitions of network APIs for supporting gaze-aware streaming requirements. A crucial component is therefore the definition of functions at the *network orchestration interface*, to optimize video quality selection as a function of available network bandwidth and RTT.

To effectively adapt the system's parameters to varying network conditions additional efforts will focus on developing novel metrics for measuring QoE in real time. These will be based on the hit probability described here and will also include aspects from objective metrics such as PSNR, but applied in the proximity of the eye-gaze estimates.

Additional research dimensions include both exploiting predictions of the users' fixations, to proactively control the activation of tiles in the foreground region, and the exploration of alternative shapes for the foveal region, with focus on solutions modulating the shape to match specific objects in the image located in proximity of the fixations points.

Acknowledgments

The research presented in this paper is part of the SEEN project, funded by Vinnova, with partners Tobii, Ericsson and KTH. The authors wish to thank Joakim Karlén and Mårten Skogö at Tobii for the fruitful discussions and access to Tobii's VR HMD. Special thanks go also to Calin Curescu and Johan Lundsjö at Ericsson Research for the many insights on cloud services and architectures for upcoming 5G networks. Our sincere gratitude goes to Professor Gerald Q. "Chip" Maguire Jr. for reviewing the manuscript and providing several valuable comments and insights.

References

[1] Github repository for the SEEN project, showcasing videos, results and code associated to the paper: https://github.com/MSL-EECS/SEEN.

[2] Akamai's [state of the internet], Q1 2017: https://content.akamai.com/gl-en-pg9135-q1-soti-connectivity.html. Technical report, Akamai Inc., 03 2017.

[3] Recommended upload encoding settings (bitrate). Technical report, YouTube Inc., June 2017.

[4] Specifications of Tobii Pro VR Integration, June 2017.

[5] Worldwide Quarterly Augmented and Virtual Reality Headset Tracker. Technical report, IDC Research Inc., 06 2017.

[6] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh. Realizing the Tactile Internet: Haptic Communications over Next Generation 5G Cellular Networks. *IEEE Wireless Communications - Networking and Internet Architecture*, 2017. doi: 10.1109/MWC.2016.1500157RP

[7] E. Akbas and M. P. Eckstein. Object detection through exploration with A foveated visual field. *CoRR*, abs/1408.0814, 2014. doi: abs/1408.0814

[8] E. Arabadzhiyska, O. T. Tursun, K. Myszkowski, H.-P. Seidel, and P. Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Trans. Graph.*, 36(4):50:1–50:12, July 2017.

[9] M. S. Banks, A. B. Sekuler, and S. J. Anderson. Peripheral spatial vision: limits imposed by optics, photoreceptors, and receptor pooling. *Journal of the Optical Society of America*, 11(8):1775–87, December 1991.

[10] J. A. Boluda, F. Pardo, T. Kayser, J. J. Perez, and J. Pelechano. A new foveated space-variant camera for robotic applications. In *Proceedings of Third International Conference on Electronics, Circuits, and Systems*, vol. 2, pp. 680–683 vol.2, Oct 1996. doi: 10.1109/ICECS.1996.584453

[11] G. Cheung, Z. Liu, Z. Ma, and J. Z. G. Tan. Multi-stream switching for interactive virtual reality video streaming. *CoRR*, abs/1703.09090, 2017.

[12] N.-M. Cheung, A. Ortega, and G. Cheung. Distributed source coding techniques for interactive multiview video streaming. In *Picture Coding Symposium, 2009. PCS 2009*, pp. 1–4. IEEE, 2009.

[13] C. Concolato, J. Le Feuvre, F. Denoual, E. Nassor, N. Ouedraogo, and J. Taquetet. Adaptive streaming of HEVC tiled videos using MPEG-DASH. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99), 2017. doi: 10.1109/TCSVT.2017.2688491

[14] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. Viewport-Adaptive Navigable 360-Degree Video Delivery. *CoRR*, abs/1609.08042, 2016.

[15] W. Dai, G. Cheung, N.-M. Cheung, A. Ortega, and O. C. Au. Merge frame design for video stream switching using piecewise constant functions. *IEEE Transactions on Image Processing*, 25(8):3489–3504, 2016.

[16] T. El-Ganainy and M. Hefeeda. Streaming virtual reality content. *CoRR*, abs/1612.08350, 2016.

[17] Y. Feng, G. Cheung, W. Tan, P. Le Callet, and Y. Ji. Low-cost eye gaze prediction system for interactive networked video streaming. *IEEE Transactions on Multimedia*, 15(8):1865–1879, Dec 2013. doi: 10.1109/TMM.2013.2272918

[18] G. Ghinea and G. M. Muntean. An eye-tracking-based adaptive multimedia streaming scheme. In *IEEE International Conference on Multimedia and Expo*, pp. 962–965, June 2009. doi: 10.1109/ICME.2009.5202656

[19] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3D Graphics. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2012*, 31(6), Nov. 2012. doi: 10.1145/2366145.2366183

[20] M. Hosseini and V. Swaminathan. Adaptive 360 VR video streaming: Divide and conquer! *CoRR*, abs/1609.08729, 2016.

[21] High Efficiency Video Coding. Standard, ITU, Dec 2016.

[22] Subjective video quality assessment methods for multimedia applications. Recommendation, International Telecommunication Union, 4 2008.

[23] K. Kammachi-Sreedhar, A. Aminlou, M. Hannuksela, and M. Gabbouj. Standard-compliant multiview video coding and streaming for virtual reality applications. In *Proceedings of the IEEE International Symposium on Multimedia*, ISM '16, San Jose, CA, USA, December 11–13, 2016.

[24] M. Karczewicz and R. Kurceren. The SP-and SI-frames design for H.264/AVC. *IEEE Transactions on circuits and systems for video technology*, 13(7):637–644, 2003.

[25] E. Kuzyakov and D. Pio. Next-generation video encoding techniques for 360 video and VR, Jan. 2016.

[26] G. M. Muntean, G. Ghinea, and T. N. Sheehan. Region of interest-based adaptive multimedia streaming scheme. *IEEE Transactions on Broadcasting*, 54(2):296–303, June 2008. doi: 10.1109/TBC.2008.919012

[27] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. Optimizing 360 Video Delivery over Cellular Networks. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ATC '16, pp. 1–6. ACM, New York, NY, USA, 2016. doi: 10.1145/2980055.2980056

[28] J. Ross, D. Burr, and M. Morrone. Suppression of the magnocellular pathway during saccades. *Behavioral Brain Researh*, 80:1–8, 1996.

[29] J. Ross, M. Morrone, M. Goldberg, and D. Burr. Changes in visual perception at the time of saccades. *Trends in Neuroscience*, 24(2):113–121, 2001. doi: 10.1016/S0166-2236(00)01685-4

[30] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, Dec 2012. doi: 10.1109/TCSVT.2012.2221191

[31] P. J. A. Unema, S. Pannasch, M. Joos, and B. M. Velichkovsky. Time course of information processing during scene perception: The relationship between saccade amplitude and fixation duration. *Visual Cognition*, 12(3):473–494, 2005. doi: 10.1080/13506280444000409

[32] G. Van der Auwera, M. Coban, F. Hendry, and M. Karczewicz. AHG8: Truncated Square Pyramid Projection (TSP) For 360 Video. *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*.

[33] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Hämäläinen. Kvazaar: Open-source hevc/h.265 encoder. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pp. 1179–1182. ACM, New York, NY, USA, 2016. doi: 10.1145/2964284.2973796

[34] A. Vlachos. Advanced VR Rendering. In *Game Developers Conference*. GDC, San Francisco, CA., 2015.

[35] F. Volkmann, L. Riggs, K. White, and R. Moore. Contrast sensitivity during saccadic eye movements. *Vision Research*, 18:1193–1199, 1978.

[36] S. J. Wilson, P. Glue, D. Ball, and D. J. Nutt. Saccadic eye movement parameters in normal subjects. *Electroencephalography and Clinical Neurophysiology*, 86(1), 1993.

[37] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj. HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications. In *Proceedings of the ACM on Multimedia Conference*, MM '16, pp. 601–605. ACM, NY, USA, 2016. doi: 10.1145/2964284.2967292

[38] W. Zhou and A. C. Bovik. Foveated image and video coding. *Digital Video, Image Quality and Perceptual Coding*, Dec. 2006.