Tiling in Interactive Panoramic Video: Approaches and Evaluation

Vamsidhar Reddy Gaddam, Michael Riegler, Ragnhild Eg, Carsten Griwodz, Pål Halvorsen

***Abstract*—Interactive panoramic systems are currently on the rise. However, one of the major challenges in such a system is the overhead involved in transferring a full-quality panorama to the client when only a part of the panorama is used to extract a virtual view. Thus, such a system should maximize the user experience while simultaneously minimizing the bandwidth required. In this paper, we apply tiling to deliver different quality levels for different parts of the panorama. Tiling has traditionally been applied to the delivery of very high-resolution content to clients. Here, we apply similar ideas in a real-time interactive panoramic video system. A major challenge lies in the movement of such a virtual view, for which clients' regions of interest change dynamically and independently from each other. We show that our algorithms, which progressively increase in quality towards the point of the view, manage to (i) reduce the bandwidth requirement and (ii) provide a similar QoE compared to a full panorama system.**

***Index Terms*—Multimedia system, tiling, user studies, video, panorama.**

I. Introduction

The role of videos on the Internet has become increasingly important in the last few years. YouTube and Netflix are alternately cited as the source of most Internet traffic [1], although other large companies, such as Facebook, integrate videos and video sharing in their services [2]. The commercial use of video streaming on the Internet has not only led to a proliferation of videos but also led users to expect high-quality videos—and the service providers are fulfilling these expectations. YouTube users can already watch videos in 4k. The adoption of such high resolutions means that the classical video streaming challenge, the availability of bandwidth, persists in spite of ever-growing bandwidth capacities [3].

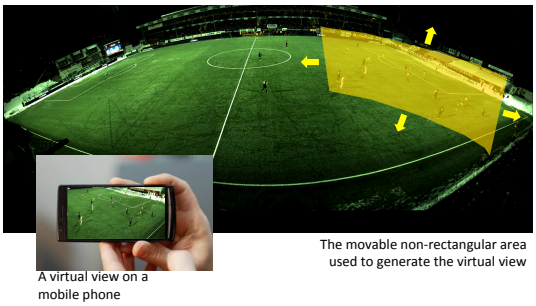


Fig. 1: The re-projected virtual view and the panorama video with the marked region of interest.

Panoramic videos are a special case among high-resolution videos. They have uses in video surveillance, sports analysis, robotics, and so on and differ from other video applications in the way users interact with them because, most of the time, users watch only a sub-section of the entire video. A large number of panorama solutions exist, including research prototypes and commercial products, but their potential is still largely unexplored and their delivery techniques are not yet perfected (for example, [4–7]).

Panoramic videos are usually created from the output of multiple cameras that cover a wide field of view and are stitched together into one high-resolution frame. Users are then commonly given the opportunity to access narrower views extracted from the panorama using pan-tilt-zoom (PTZ) operations. This means that each user controls one or more virtual cameras interactively to create their view. Panoramas created by stitching multiple camera views together are typically cylindrical panoramas, as shown in Figure 1. A cylindrical panorama provides a roughly uniform distribution of pixels over all angles recorded in the panorama; consequently, even panning operations are more complex than standard cropping operations.

Furthermore, in systems with a high number of users, the virtual view is usually generated on the client side to allow the system to scale and keep interaction latency low. The downside of this approach is that the entire panorama must be delivered to the client at all times because the user can perform PTZ operations at any time. Obviously, the delivery of a full-quality high-resolution panorama video is (excessively) costly in terms of bandwidth. For example, when we generated a 4096x1680 x264-encoded panorama video for an example installation, the average bandwidth requirement was approximately 9.5 Mbps. Moreover, because only parts of the panorama are used to extract the virtual view (see Figure 1), sending the full-quality panorama for each user wastes bandwidth.

To reduce the waste but preserving the user's high quality of experience (QoE), we present an analysis of options and propose a solution that combines tiling and HTTP adaptive streaming (HAS). The solution requires three main steps: (i) Using ideas from region-of-interest streaming and retrieving higher quality from the areas of the full panorama that comprise the virtual view, we analyse and discuss the trade-offs. (ii) We conduct a subjective study to validate the objective quality metrics for our scenario. Then, we use these metrics to investigate the trade-off between video quality and bandwidth for several adaptation strategies. (iii) Finally, we present a cost-effective solution for transferring a high-quality panoramic video to the user. Our experimental results show that our approach reduces the bandwidth requirement and provides a similar QoE compared to a full panorama system.

Manuscript received October 10, 2015; revised April 23 , 2016.

The rest of the paper is organized as follows. First, the current state-of-the-art is discussed in Section II. Section III provides a detailed overview of the system. Next, quality selection approaches are discussed in Section IV. We present the evaluation methods in section V and the evaluation results in Section VI. Section VII discusses the outcome of the evaluation, and conclusions are given in section VIII.

II. Related Work

Panoramas can be divided broadly into two groups, i.e., complete panoramas that span 360^o around at least one axis and partial panoramas with less angular coverage. When it comes to complete panoramas, a cube-panorama is a standard format. They consist of 4-6 images with 90^o field of view put next to each other. The format is convenient because it allows the creation of virtual views with only one linear transformation (homography) for each side of the cube. Its deficiency is that the number of pixels representing a viewing direction varies strongly between the center and the corner of a cube side. A cylindrical (spherical) panorama can also be used for a complete panorama. It reduces the pixel density problem, but requires the computation a homography for every column of pixels (every pixel) to project from the panorama onto a virtual view. For both cases, a compact representation is crucial wherever many users interact with the panorama video at the same time over the network. They affect storage space and memory usage as well, which may also be a performance concern.

Panorama Systems. Panorama systems that support user PTZ operations on a virtual camera have been developed in both research [4–12] and industry [13, 14].

PTZ cameras have existed for a long time for obtaining static panoramic images and are common in services such as street maps and photosynths. However, such systems deliver video experiences using similar interfaces. This adds numerous challenges to the systems and several additional dimensions to the problem of delivering an interactive video experience.

The articles [8] and [4] present a system for live/real time production of broadcast video using PTZ cameras. Specifically, [8] focuses on efficient interfaces to create a live virtual camera for a single producer. Carr *et al.* [10] discuss virtual PTZ cameras to control/steer a robotic PTZ camera. The advantage of using a robotic PTZ camera is that it can use optical zoom and, hence, provide high resolutions even at higher focal lengths. However, using a robotic PTZ camera exclusively limits the number of users who can control the camera to *one*.

In [11] used virtual PTZ cameras to follow a speaker when recording lectures. In addition, [15] focused specifically on lecture videos. At an abstract level, from an interactive experience point of view, indoor applications are similar to outdoor ones. However, the photometric challenges in outdoor applications are not comparable to those of indoor scenes due to variable lighting and greater depths, which can drastically affect the user experience. A good way of recording outdoor panoramas using HDTV cameras is provided by [16]. Some works [17, 18] investigate the system aspects of panorama capture systems, but only a few works [18, 19] discuss distributing the live panorama video—and even those lack a complete evaluation. Most industry projects work by transferring the entire panorama before starting the interaction; thus, they do not constitute a *true live* component. However, YouTube 360 has just released the first 360^o videos that deliver 4-sided cubic panorama videos stitched into a single video stream. These videos support pan and tilt operations (but not zoom) in the Chrome browser.

Streaming Options. Tiled video can be processed into an individual stream for each viewer on the server side [20], but this approach does not scale to a large number of concurrent viewers when those viewers can choose individual views. To support individualized views, the tiles that cover the user's desired view must be delivered from the server, with subsequent processing occurring on the client side. We are not aware of a discussion on the options for this approach.

All distributed tiling systems face the challenge of user interactions that require rapid changes to the user's view and, consequently, require new covering tiles to be delivered between two consecutive frames. Users are able to notice delayed reactions to their interactions even when the latency is only a few milliseconds [21]. To avoid this latency, tiling systems that extract views on the receiver side choose to retrieve all tiles (within interaction range) at all times, but they do so at a less than perfect quality to save bandwidth. HAS is well-suited for this multi-quality delivery because it can deliver multiple quality levels to large audiences with the help of standard Web caches to increase scalability. However, retrieval decisions can be made only on segment boundaries, which means that visual quality can be reduced for several frames after a user interaction affects the required tiles.

Faster quality improvement could be achieved by downloading a higher quality version of a segment that comes into visual range and decoding it, skipping frames that have already been displayed at low quality and continuing with the new higher-quality frames. However, this technique imposes sudden high demands on download bandwidth and decoding. Alternatively, Scalable Video Coding (SVC) and Mid Grain Scalability (MGS) could be combined with HAS [22], increasing quality by retrieving an enhancement layer. This approach places less load on bandwidth and allows the receiver to improve frame quality immediately after skipping to the correct frame in the enhancement layer. However, an H.264 SVC-encoded video has a 10% bandwidth overhead per enhancement layer compared to non-scaled video of the same quality [23].

Push-based streaming systems are an alternative because they can encode each tile as a continuous stream. Solutions that require multicast [24, 25] cannot be used on a large scale due to the lack of IP multicast. However, even in a unicast solution, a push server can respond to a receiver's request for higher quality within one Round Trip Delay Time (RTT) of a user request. One method works by updating the Session Description Protocol (SDP) [26], which can switch the unicast delivery of layers on and off—but of course, the SVC overhead mentioned above applies here as well. An even faster method is based on RTP [27], which can send a bit-rate request and instruct the server to send a new intra frame as soon as possible. This option is interesting because

it works with SVC (suffering the mentioned overhead), with non-layered codecs but live encoding (or transcoding) at the sender, or with a set of parallel streams where switching is supported through SI/SP frames [28]. The overhead of the SI/SP method lies between the other two approaches. However, these RTP-based methods all share the problem that packet loss can occur; therefore, today's systems usually use MPEG 2-TS packaging [29]. Unfortunately, this approach incurs a 20% bandwidth overhead [30].

Considering that all the approaches discussed above demand that the base-layer quality of all reachable tiles must be streamed at all times, the bandwidth overhead of the various alternatives to HAS seemed too large for our scenario. We have therefore chosen a HAS with 1-second segments. We discuss the QoE implications of the quality switching delay below.

Tiling Approaches Using HAS. Even though not directly related to the cylindrical/spherical panorama systems that provide free PTZ camera movement, there are some works [20, 31–33] that provide an approximate interaction. Tiling in interactive panorama video is discussed in [31]. However, they used a perspective panorama; consequently, the virtual camera merely performs cropping, a process that is identical to cropping from a high-resolution video. The most recent works by Sanchez *et al.* [34, 35] address the issues of scaling in panoramic systems and streaming. One of the drawbacks of tiling is the need to decode several streams in real time and then assemble them. In this regard, Sanchez *et al.*, using their approach along with H.265/HEVC/SHVC, can generate a single video bitstream of selected tiles in the compressed domain. Single hardware decoders are then used to decode the RoI in the video stream, thus improving the streaming and scaling performance. Liu *et al.* [20] provide zoomable playout on mobile devices for larger resolution videos, and [32] present an approach for zoomable video in which the tiles are optimally selected and sent from the server side. The user study performed by [36] investigated the effect of tiling on the zoomable video presentation. However, except for [31], these works do not support a completely random PTZ camera. A similar system was presented by [33], in which the tiles are encoded at multiple qualities and retrieved depending on the current view; however, they do not discuss smooth random movement. Their interface is similar to that of a zoomable video, where a user can pick a portion of the entire video presented in a thumbnail. Then, that part is cropped from the full resolution and presented to the user. Hence, to our knowledge, our work is the first to handle the problem of tiling and discuss its trade-offs in the context of a random PTZ camera for a cylindrical panorama texture.

Other Techniques for Interactive Panoramic Systems Apart from the technologies presented so far, there are several other techniques for interactive panoramic systems that should be mentioned. The tiled large-screen autostereoscopic display [37, 38] and the recent super multiview (SMV) displays [39] also offer interactive navigation capability for videos. In particular, multi-view based and view-plus-depth based techniques can be used to generate the required view using view synthesis techniques. Farid *et al.* present a panorama-based solution that can be used to enable interactive panoramic

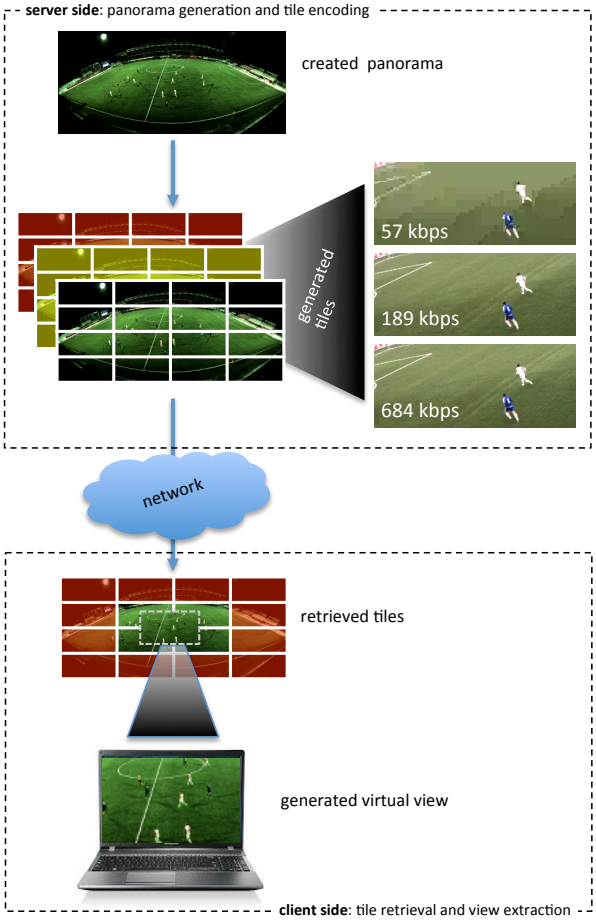


Fig. 2: At the server side, we divide the generated panorama video into 8x8 tiles and then encode each tile at different qualities. The client retrieves the appropriate quality tiles based on the current position of the virtual camera (full quality tiles for the virtual view and low quality (red) tiles outside the field of view).

video navigation using depth-aided inpainting [40] and spatiotemporal occlusion compensation [41]. Both are specifically designed for 3D video coding.

In [42], we present a system for real-time interactive zooming and panning that reduces the per-user bandwidth requirement by taking quality changes into account in different parts of the panorama video when moving the virtual camera. Depending on the actions of the user, the results show that there is a large potential for reducing the transfer cost. In this article, we present an extended version of the previous study's [42] preliminary results along with an evaluation of the QoE resulting from different tiling approaches.

III. System Overview

Our panorama system is currently running live at two different locations. The tiling generation and retrieval operations are highlighted in Figure 2. All components run in real-time; consequently, users can control the virtual camera during a live stream.

Server Side. Cylindrical panorama images are generated from five 2K cameras whose shutters and exposures are perfectly synchronized. The seams are calculated dynamically

for every frame. The frames are divided into 64 tiles (8x8), and one video stream is generated for each tile. Each video tile is encoded into 1-second segments at multiple qualities (and bit rates) using *libav* and *x264*. Each tile can then be requested individually by the client using HAS.

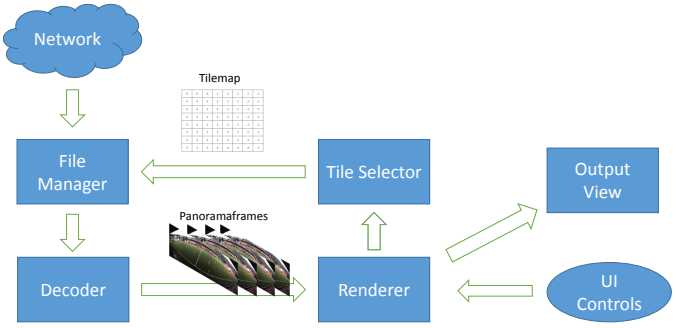


Fig. 3: The architecture of a client that supports tiling.

Client Side. After the different quality tiles are available on the server, the client fetches tiles and generates the virtual view from the retrieved panorama. The task of the client is to retrieve high-quality tiles for the virtual view and lower quality tiles for the surrounding areas. Thus, the client is able to supply the user with a high-quality virtual view video while at the same time saving bandwidth compared to the full-quality panorama retrieval approaches discussed in the previous section.

However, the system must fetch, spatially, every tile in the panorama video, whatever the quality might be. This way, the system can still provide data if the user interactively moves the virtual camera. The alternative would be to present black areas or a static image when none of the surrounding tiles had been retrieved at all and it is aesthetically undesirable. In order to provide the tiling functionality, the client is designed as shown in Figure 3. There are four major components in the client system, (i) a File Manager, (ii) a Decoder, (iii) a Renderer and (iv) a Tile Selector.

File Manager. The File Manager component is responsible for requesting the appropriate tiles at a given quality from the server (determined by the Tile Selector described below). The byte stream is transferred when fetched and forwarded straight to the Decoder module rather than saved as a local file, thus bypassing the disk.

Fig. 4: Sample output frame from the decoder module.

Decoder. After the tiles are available from the File Manager, the Decoder module begins decoding frames and pushing them to a common panorama texture. Because the tiles are spatially independent of each other, this process is heavily parallelizable. The operations are frame-synchronized to avoid placing tiles from two different frames at different time instants into the same panorama frame. Figure 4 shows an example frame[1] revealing how the panorama frame is reconstructed from different quality tiles.

Renderer. As soon as a panorama frame is decoded, it is pushed to the rendering module. This module is responsible for creating the virtual views using the PTZ parameters. In addition, it provides user interaction and virtual view controls. In most interactive systems, the functionality of the Renderer is limited to this. However, to support tiling, we need to save information about the parts of the panorama that are currently being viewed. This information is transferred to the Tile Selector module, which then uses that information to select the appropriate tile qualities for the next iteration.

Tile Selector. After a frame is displayed, the Renderer transfers the panorama location from which the current view is extracted to the Tile Selector. This information is crucial for selecting the next set of tiles. Finally, it is important to point out that all these modules need to perform in real-time to provide a smooth interactive experience to the user. Moreover, they must do so while keeping bandwidth consumption to the minimum required level. One can observe that this is a challenging task in the Decoder module, where several videos are expected to be decoded concurrently in real-time and also frame-synchronized.

IV. Quality Selection Approaches

As described earlier, the Tile Selector is responsible for determining the appropriate qualities (and bitrates) of the different tiles needed, and it must adapt according to the viewer movement. Let $Q = \{q_0, q_1, \ldots, q_{n-1}\}$ be the set of n available quality levels and T_i be the tile quality at tile i. Then, the problem can be written as a simple labelling problem in equation 1. The qualities are in decreasing order, where q_0 is the highest quality tile, as follows:

$$T_i = q \quad \text{where} \quad q \in Q. \tag{1}$$

There are several ways to perform this labelling. The selected method will ultimately influence the bandwidth consumed and the user experience of the system. A *binary tile occupancy map*, containing information on which tiles are currently used to generate the virtual view, is used in the labelling process. The binary occupancy map has $B_i = 1$ at tile i when the view needs pixels from tile i on the panorama. Even when using the same binary occupancy map, there are several ways to select the tile quality. Below, we briefly outline some of the algorithms evaluated in this study. The three first algorithms make a binary decision between a predefined, yet configurable, high or low quality. The last approach allows for a gradual (multi-level) decrease of quality depending on the importance of a tile.

[1]Due to the possibly limited resolution of printers, we recommend analysing the images on screen.

A. Binary

The binary approach is a straightforward approach in which high quality is assigned to the required tiles and low quality to the ones that are not required (Figure 5). Using the binary occupancy map described above, this choice becomes rather trivial. Hence, the binary approach can be formulated as follows:

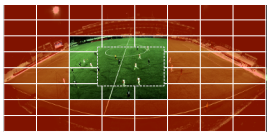


Fig. 5: Tiled binary.

$$T_i = \begin{cases} q_h & \text{if } B_i = 1 \\ q_l & \text{else.} \end{cases} \tag{2}$$

The only requirement in this case is to have $l > h$. However, the choice of the exact quality levels can be considered as tuning.

B. Rescaled

One approach commonly used in research in terms of tiling is to send a low-quality base thumbnail video and provide only the required high quality tiles [31, 32] (Figure 6). To create the thumbnail video, the source video is down-scaled and stored. During the process of virtual view generation, pixels from the available high quality tiles are used. For pixels where the high quality data are missing, the thumbnail video is up-scaled and used. These can be considered as low quality tiles.

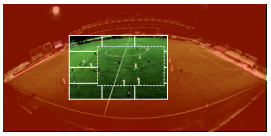


Fig. 6: Thumbnail.

C. Prediction

When a user moves the virtual camera, there is a chance that the view will be generated by some low-quality tiles because tile quality changes only at the segment boundary. To reduce the probability that this will occur, it is beneficial to try to predict future movements and retrieve higher quality tiles when there is a high probability that the user will move the view to this tile (Figure 7. This is similar to the tiled binary, but it enlarges the high quality area based on the prediction. In this respect, it is beneficial to predict the path across several frames into the future. Several prediction models are available; however, to keep the comparison to the state-of-the-art consistent, we used the Auto Regressive Moving Average (ARMA) prediction [31]. For this method, let θ_t be the position and $\delta\theta_t$ be the velocity of the view at the time instant t. The velocity at the current instant is estimated as

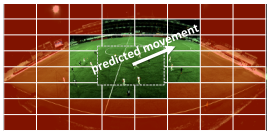


Fig. 7: Predicted.

$$\delta\theta_t = \alpha\delta\theta_{t-1} + (1-\alpha)(\theta - \theta_{t-1}). \tag{3}$$

Then, the future position $\hat{\theta}_{t+f}$ at $t+f$ is estimated as

$$\hat{\theta}_{t+f} = \theta_t + f\delta\theta_t, \tag{4}$$

where f is the number of frames predicted into the future. The result can be used immediately to construct a future binary occupancy map, and that map can be used in any of the approaches mentioned here. However, for the sake of comparison, we use the Predictive approach along with the Rescaled approach [31].

D. Pyramid

The pyramid is a complex scheme in which we chose qualities intelligently with a gradually decreasing quality according to the distance from the virtual camera (Figure 8). Here, we introduce the term *priority* (p_i) that varies within the range $[0, 1]$, where 0 is the most important and 1 is the least important. Depending on the importance, we fetch the corresponding quality. However, there is a catch. If we were to decide solely on the importance, we might end up fetching large numbers of high-quality tiles for a zoomed-out virtual view. Here, the maximum quality level (q_{max}) comes into play. This quantity depends on the number of high priority tiles. We select q_H as the quality level to be used when all the tiles are being used for the virtual view as shown below:

Fig. 8: Pyramid.

$$q_{max} = (\frac{\sum_{i \in T} b_i}{N}) q_H \tag{5}$$

$$T_i = \begin{cases} q_{max} & \text{if } b_i = 1 \\ q_{max} + p_i(n - q_{max} - 1) & \text{else.} \end{cases} \tag{6}$$

Here, T is the set of all tiles on the panorama and n is the number of quality levels. After calculating q_{max}, we count the occupancy of the neighbourhood ($\mathcal{N}_i$) of tile i and then assign that as its p_i, as shown in equation 7. As one can observe, there are several tunable parameters. One is q_H, which determines the quality at a given zoom level. A second is the selection of the neighbourhood itself, which can be determined by the weights of α_j. We can make the weights be either isotropic or anisotropic. Given the fact that users are more prone to pan than to tilt, anisotropic weights can lead to similar performance as isotropic ones while consuming less bandwidth.

$$p_i = 1 - \frac{\sum_{j \in \mathcal{N}_i} \alpha_j b_j}{\sum_{j \in \mathcal{N}_i} \alpha_j}. \tag{7}$$

V. Evaluation Methods

The problem of bandwidth reduction involves a strict trade-off of two conflicting constraints. One constraint is the bandwidth itself, which can be measured directly as the rate of data transferred. The second constraint is the quality of experience, which is not trivial to measure. When developing approaches, we need to consider how well the approaches perform with respect to these constraints and which approaches provide

the best trade-off. We compare the two different pipelines in Figure 9 and use the final output (the rendered virtual view) for comparison.

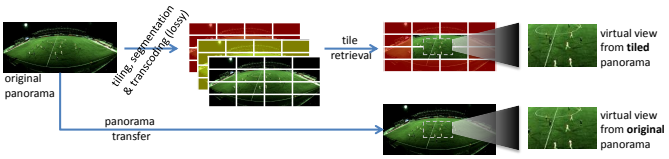


Fig. 9: Pipeline differences: tiled vs. original panorama.

We assessed the quality selection schemes described in Section IV by comparing their performances on the entire first half of a soccer game (approximately 47 minutes). We used five quality levels with increasing Constant Rate Factor (CRF) values as follows: $q_0(21)$, $q_1(24)$, $q_2(30)$, $q_3(36)$ and $q_5(48)$. For this evaluation, we compared the quality of four pre-generated sequences of PTZ operations, called *paths*. We created four paths whose pan/tilt operations follow the general soccer game flow but whose zoom varies as described and labelled in Table I. The quality selection methods were labelled as shown in Table II.

Label	Path
s1	The virtual camera is severely zoomed in
s2	The zoom is at a medium level
s3	An overview video where the view is zoomed out
s4	A dynamic zoom factor depending on the situation

TABLE I: Paths: sequences of PTZ operations

Label	Approach
l1	Binary with q_0 and q_4
l2	Binary with q_1 and q_3
l3	Rescaled with no prediction
l4	Rescaled with 100 frame prediction
l5	Pyramidal with isotropic weights
l6	Pyramidal with anisotropic weights
l7	Pyramidal with isotropic weights (different parameters)
l8	Full Panorama input (no tiling)

TABLE II: Labelling of approaches for analysis

A. Quality metric

The challenge for objective video quality metrics has so far been to match subjective viewing experiences for videos of finite duration (8–12 seconds). Objective methods that have tried to solve this challenge and that have undergone rigorous independent testing [43] are intended for constant-quality videos (with uniform disturbances). These methods can estimate QoE when degradation in a video spans several frames and work well for individual HAS segments. However, they may not be suitable when the user is presented with a view that is stitched from several independently adapting HAS video tiles. In this scenario, only parts of the video suffer from distortion, and as updates occur, they can instantly change the degradation.

To compare the quality selection schemes in our paper, we have therefore conducted a user study to assess whether the image similarity metric SSIM [44] or OpenVQ, an independent implementation of a perceptual video quality metric described in ITU-T J.247 [45, Annex B], provides good estimates of subjective quality assessment. All user study experiments are performed using 12-second excerpts[2] from the 4 sequences mentioned in Table I. In Figure 10, one can observe the number of tiles, out of the 64 possible per frame, used during the virtual camera operation.

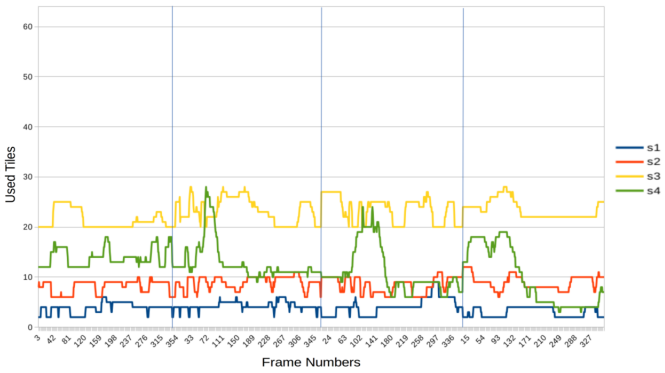


Fig. 10: Total number of tiles used, out of 64 possible, per frame in sequences of 12-second durations at 4 predetermined time instants.

OpenVQ: Perceptual Evaluation of Video Quality (PEVQ) is a full-reference algorithm that outputs mean opinion scores (MOS) as an objective video quality metric. After evaluation by the Video Quality Experts Group (VQEG), it has become part of the standard ITU-T J.247 [45]. Out of all the candidates, PEVQ achieved the best error rate with respect to subjective studies conducted by two independent institutes.

Unfortunately, PEVQ is not freely available for researchers, so we used OpenVQ, which is based on J.247 Annex B but is not a one-to-one implementation of PEVQ. The *patented* consideration of temporal alignment has been dropped because neither HAS nor RTP-based streaming suffers from temporal misalignment. Furthermore, flaws in the formulas in J.247 Annex B force a rather loose interpretation. The dataset used to evaluate J.247 candidates is not publicly available, but using the ground truth of ICCRyN datasets [46–48], OpenVQ achieves results close to those reported for PEVQ in J.247. We provide OpenVQ as open-source software under the terms of the GNU Affero General Public License (AGPL) version 3, as published by the Free Software Foundation [3].

SSIM: The Structured Similarity Index (SSIM) [44] is a metric for assessing differences between images. It is supposed to model human subjective experience quite well, but [49] have demonstrated that this fails for a variety of possible image degradations. Nevertheless, SSIM is often used to estimate video quality. For example, $x264$ makes encoding decisions based on this metric, and [50] constructed a video quality assessment tool based on it.

PSNR: One commonly used measure for evaluating video quality is PSNR. As [51] explains, it is solely a pixel difference metric and quite unrelated to subjective experience. [52] already explained its limits, while [53] have clarified that it can predict human preference in one particular case: when the

[2]The same sequences are attached along with the paper, but compressed due to space limitations.

[3]https://bitbucket.org/mpg_code/openvq

same content has been encoded with different compression strengths.

B. Assessing objective metrics

The measure that we require for this study differs from the ground truths that have been used in assessing other current video quality metrics. As mentioned in Section V-A, we have tiled videos that follow a HAS model. An adaptation decision for each tile is made once a second. We do not aim at generating a single quality value for an entire 47-minute test case because we have not found any valid basis for doing so in the literature. Instead, we verify how well the above objective metrics describe user experience on a second-by-second basis.

1) Study design: We compared the results of the objective metrics with subjective evaluations across a range of tiling approaches. The user study was designed to investigate two aspects of the subjective perception of quality. We consider the noticeability of quality distortions and the experienced annoyance to be related but distinct. We ran two consecutive experiments, one to address the detection of tiling distortions and one to address the annoyance resulting from those distortions. In addition, we included five-point absolute category rating scales for subjective assessments of overall video quality, adhering to ITU-T P.911 [54].

In both experiments, participants watched sequences with durations of 12 seconds extracted from the sequences described at the beginning of Section V. These sequences were originally chosen as representations of different football scenarios and, hence, provided variety to participants and served to increase generality. Because all the sequences included pre-recorded camera panning and zooming movements, our final stimulus collection contained sequences with frequent tile shifts and varying changes in compression rates and video quality. In the detection experiment, we instructed participants to pay attention to the quality of the presented sequences and to push the spacebar down the moment they noticed a change for the worse, holding it down for the entire duration of the quality drop. The annoyance experiment followed the same procedure, only the instructions were changed to ask participants to push down the spacebar while they experienced annoyance due to low video quality. At the end of each sequence, participants rated the overall video quality on a 5-point scale ranging from ”bad” to ”excellent”.

To secure a sufficient number of participants, we recruited the assistance of crowdsourcing workers. This approach requires some extra methodological considerations due to challenges that concern lack of task adherence and comprehension and, in turn, reduced data consistency [55, 56]. Thus, we initially conducted 3 pilot studies to ensure that the experiments were presented in a succinct but understandable format. The first was completed by colleagues and students and the following two on the crowdsourcing platforms Microworkers and Crowdflowers. Following each pilot, we adapted the experiments according to the received feedback. For the final study, we used Microworkers and collected data from a total of 200 different participants. Although we implemented quality measures such as gold samples and majority votes, the highly subjective nature of the task did not allow more than the most basic automatic filtering to exclude non-complying individuals. We excluded only participants who failed to complete the experiment. Later, on manual inspection, we also removed participants who had obviously attempted to circumvent the experimental tasks—altogether 15%. All other potential exclusion criteria were found to potentially exclude valid human perceptions as well.

We then calculated the agreement between participants' quality ratings for each sequence using Fleiss's Kappa statistic [57]. Because this statistic depends on the number of raters and comparisons [58], we consider it in the context of the possible minimum and maximum values, which are established at -0.80 and 1. For the detection experiment, inter-rater agreements varied between 0.22 and 0.37 across the different sequences and quality conditions. The annoyance experiment yielded values between 0.24 and 0.39. With respect to the arguably subjective and variable measures of detection and annoyance, we judge these positive agreement scores as indications that participants adhered to the task at hand.

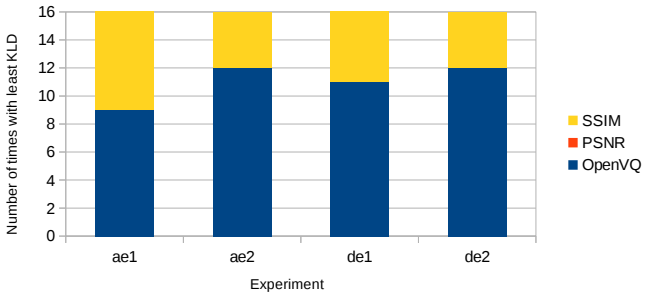
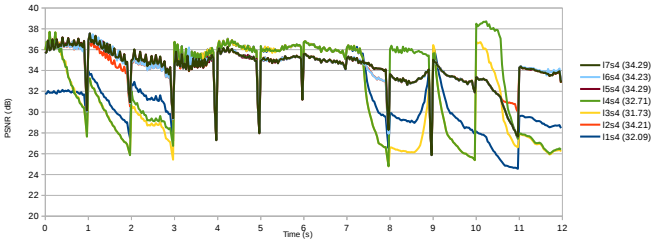


Fig. 11: Number of times a metric had the least divergence from the user input in each task of the experiments among OpenVQ, PSNR and SSIM.

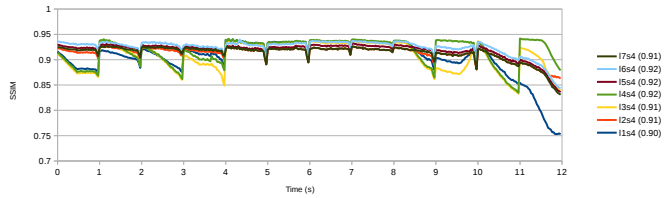
2) Performance of Evaluation Metrics: The analysis of user studies for perception is always challenging, especially when the users are expected to provide time-varying input. For example, our study aims at recording differences in perception among users but also records response time differences between them. It may be possible to counteract this by assuming that users' opinions correlate and maximizing the cross correlation by time shifting all inputs. However, due to the weakness of this assumption, we ignored response times and averaged user inputs across all users.

The results of our user study show a reasonably strong relationship between the user input and OpenVQ, as well as SSIM. We used Kullback-Leibler divergence (KLD) [59] to estimate the information loss in approximating subjective results with the objective metrics. KLD stays below 0.05 for path $s1$ and below 0.01 for the other paths. Figure 11 shows that both OpenVQ and SSIM can be closer to the average subjective ratings[4]. In Figure 12, one can see various evaluation metrics over a 12-second segment used in the user study.

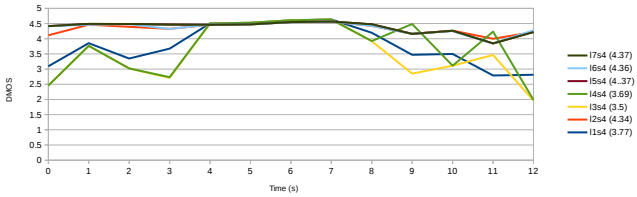
[4]Standalone HTML/JS plots for each study are attached to the paper for reference. Any recent web browser can be used to explore them.



(a) PSNR.



(b) SSIM.



(c) OpenVQ.

Fig. 12: Different variations of the quality metrics across the 12 second clips. For reference, the average of each metric across the 12-second duration is presented in parentheses for each approach.

(a) l3s4 - PSNR: 30.39 (b) l7s4 - PSNR: 30.89

Fig. 13: Example of severe differences within a frame (319), leading to similar PSNR values.

Although PSNR is unsuitable as a metric of visual quality (also quite easily shown to fail in the case where high- and low-quality tiles are merged into a single view, as in Figure 13), we also present PSNR results because they expose unexpected properties of the 1-second video segments. The PSNR results in Figure 12a exposed regular severe degradation of the last frame in each 1-second segment. Although this is not noticeable to a human observer even when single-stepping through the video, it is clear evidence of problems in *ffmpeg* or the way in which we used it.

VI. Evaluation Results

In this section, due to space limitations, we mostly discuss the $s4$ sequence, which is representative of the real-life virtual camera operation. However, using the other sequences, we can observe zoom-specific results. For example, $s1$ consumes the least amount of bandwidth due to the low number of required tiles. We can also observe from $s3$ that when the user is interested solely in an overview of the field, there is no need to fetch the highest quality tiles.

A. Bandwidth

A simple way to determine the cost of network delivery is to measure the bandwidth. The most commonly used measurement is the average bandwidth over the entire run. Figure 14 presents a box plot of each approach using different paths for the first half of the game. However, the interactive services can have different bandwidth requirements at different instances. Therefore, we use a running bandwidth profile to evaluate the performance of the approaches. Figure 15 shows the bandwidth profile for all approaches for a 90-second segment starting 1000 seconds into the game. We can observe that there is some correlation with the number of tiles used at that time instant, which can be seen in Figure 16.

Tiles	Total	out21	out24	out30	out36	out48
2x2	17G	7.5G	5.0G	2.2G	1.2G	528M
4x4	18G	7.7G	5.3G	2.5G	1.5G	821M
8x8	23G	8.7G	6.4G	3.7G	2.4G	1.8G

TABLE III: Data size for a $6,297$-second soccer video using different tile granularities when compressing each tile with CRF values of 21, 24, 30, 36 and 48 on 1 second segments. In comparison, the size of the non-tiled panorama using the same segment length is 7.3 GB.

The methods are tuned to provide similar bandwidth with slight variations depending on the number of tiles used. However, it is quite evident that the approaches that use the highest quality tiles wherever required will require high bandwidth when many tiles are used in the view. This can be seen in the large bandwidth requirements for $l1$, $l3$ and $l4$. However, over the long run of the random zoom sequence ($s4$), which is probably most representative of a real scenario, the bandwidth consumption for all approaches is quite similar. For an estimate of the costs on the server side, we present the total disk space occupied by the tiled segments in Table III. Irrespective of the approach, it can be observed that the bandwidth savings are quite high, sometimes reducing requirements to only 25% of the requirements for the full panorama. Hence, it becomes important to evaluate the approaches for quality.

B. Quality

So far, no approach exists that can simultaneously provide both the best visual quality and low bandwidth usage at the same time during the entire virtual view operation. However, some approaches, especially the pyramidal ones, can provide both decent bandwidth savings and acceptable quality most of the time. As shown in Figure 17, all the methods suffer from quality degradation at times. The predictive approach is functional and provides improvement only when the actual positions match with the predicted positions. However, with a completely random operation, this can be challenging even

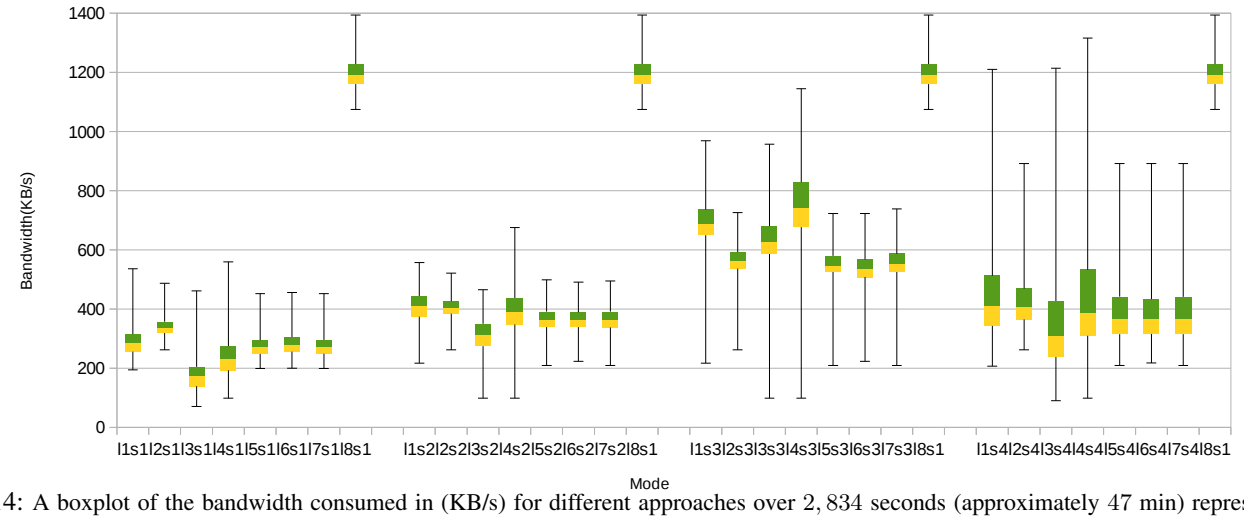


Fig. 14: A boxplot of the bandwidth consumed in (KB/s) for different approaches over $2,834$ seconds (approximately 47 min) representing the first half of a game.

Fig. 15: Bandwidth profile for a 90-second segment in the middle of $s4$.

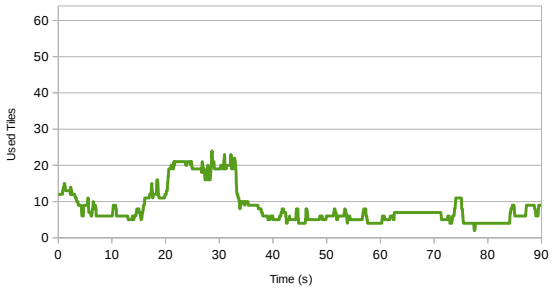
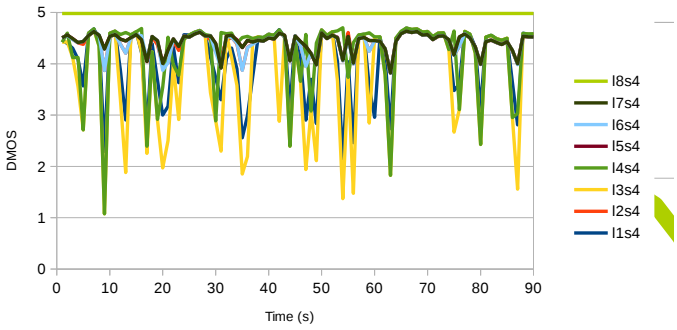


Fig. 16: Number of used tiles during 90 seconds at $1,000$ seconds into the soccer game for $s4$.



(a) OpenVQ variation

(b) SSIM variation

Fig. 17: Measured variation during 90 seconds at $1,000$ seconds into the soccer game for $s4$

with sophisticated algorithms. Moreover, the prediction algorithms seem to be the most expensive in terms of bandwidth. This occurs because not all high quality tiles fetched in the predicted areas are used for extraction of the virtual view; the user might not move the virtual view to the areas predicted by the algorithm.

In any tile-based fetching approach, we can safely assume that the changes in quality and bandwidth occur when the virtual camera is moving, not when it is static. This behaviour can be observed both in the bandwidth consumption plot (Figure 15) and in the quality plot (Figure 17). The quality is usually close to the original but drops when the movement of the virtual camera is large. The different tiling approaches tend to perform differently only in the parts where the quality drops. One of the main aims of the approaches should be to not reduce the quality significantly even with movement.

In Figure 17, we can see that a value of 0.93 for SSIM and 4.5 for DMOS runs along the time, with drops depicting the quality changes during the virtual camera movement. These values imply that the visual quality of the tiled virtual view output is on par with the original. Even during the drops, we can observe that the pyramidal approaches perform better than the others. However, SSIM and OpenVQ are full-reference quality measures, which implies that the evaluation can only be carried out in the presence of the high-quality virtual view. Still, there are ad hoc measures that one can collect in the background without much resource consumption and that can provide some insight into the quality of a virtual view.

Furthermore, Mavlankar *et al.* [31] introduce the notion of missing-pixel percentage to evaluate the accuracy of their prediction and thus the quality of the virtual view. A *missing pixel* is a pixel in the virtual view where the corresponding high quality panorama data are not available for rendering. The percentage of missing pixels can then be calculated against the total number of pixels in the virtual view. The average percentage of missing pixels across several seconds is used to evaluate various approaches in [60]. In the previous tiling approaches [32, 60], where the selection is mostly a binary process using either a high quality tile or a low quality thumbnail, the missing pixel percentage can contain considerable information about the quality. However, we also include pyramidal approaches in the evaluation that use multiple quality levels. Hence, we propose using a *Tile Histogram*. In a frame of the output virtual view, we count the percentage of pixels fetched from each tile.

[60] also provides evaluation results for the average percentage of missing pixels for a 480×240 cropped view of a $2,560 \times 704$-pixel panorama. This 6.7% ratio is equivalent to using 4 tiles in a 64 tiled panorama ($s1$ from Figure 10), in which case the average percentage of missing pixels (approxi-

Label/Sequence	s1	s2	s3	s4
l3	20.22	10.20	2.60	9.44
l4	18.20	8.56	1.94	6.53

TABLE IV: Average percentage of missing pixel measurements over the entire first half of the game

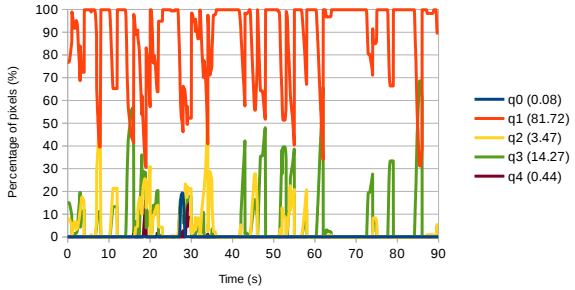


Fig. 18: Pixel histogram during 90 seconds. The average percentages for each level appear in parentheses.

mately 20%, from Table IV) is coherent with their results. As Table IV shows, we can also observe that the missing pixel percentage varies depending on the zoom level. An example profile of a pixel histogram is plotted in Figure 18. One can observe a certain correlation between the pixel histogram profile and the variations in quality observed from OpenVQ or SSIM. Ad hoc metrics such as these can be used as a reference to check quality on the fly during the process. However, full-reference metrics provide the most accurate insights into the quality variations.

The results we provide come from operating the interactive virtual view in a manner that reflects an actual operation. However, there are several factors influencing the strategies in a real world scenario. Some of these are the speed of the virtual camera movement, the number of key frames in the encoded stream, individual preferences for the zoom factor and the original panorama video properties. Our evaluation results demonstrate that tiling approaches can be designed to save bandwidth and provide good quality at the same time. However, the fine tuning is dependent on the

VII. Discussion

The study presented in this paper uses HAS as the delivery method for tiled panorama video, although this is not the only option. The QoE provided by systems that can adapt visual quality within a single RTT should be explored as well. However, as we discussed in Section II, there is a considerable bandwidth penalty associated with push-based unicast solutions. Nevertheless, multicast is not widely available, and we have therefore investigated a HAS-based approach.

From the analysis of quality metrics and bandwidth profiles for different movement paths of a virtual camera, we make several observations for the various approaches. We find that *pyramidal* approaches provide a stable quality across different zoom factors and random movements; they are a good trade-off between bandwidth savings and perceived video quality. When only a small portion of the panorama is used, we find that the *rescaling* approaches require the least bandwidth, but only at a significant loss in quality. The *prediction* results when using a general prediction algorithm are not particularly impressive, and [60] found that even context-based prediction does not provide much improvement.

The adaptation strategies evaluated in this paper try to adapt the quality of a tile according to the movement of the view to provide the best quality possible in the area of the panorama used by the virtual camera to generate the view. There are, however, numerous works that similarly try to optimize the quality of traditional (non-panoramic) HAS streaming according to available resources. Clients of all the major HAS variants (Apple HLS, Microsoft Smooth Streaming and DASH) use algorithms that attempt to have a high, stable quality. Additionally, researchers have presented approaches that try to optimize the segment retrieval, for example, according to buffer occupancy [61] and consistent visual quality [62]. However, pursuing this topic is out of the scope of this paper, but it would be an interesting topic to pursue in the future, providing optimal tile quality according to a combination of both the virtual view and the available resources.

We have also explored and analysed the effect of different segmented streaming approaches on quality for an arena sport scenario in which the on-field movement is small compared to the entire field. It would definitely be interesting to explore the effects in different scenarios such as ones that are *detailed but static* and *detailed with large movements*. However, these scenarios may require different treatment to achieve a good trade-off between quality and bandwidth usage.

VIII. Conclusion

This paper presented multiple approaches for tiling that can exploit the coding efficiency of H.264 to reduce bandwidth requirements for an interactive live PTZ system. We evaluated the approaches using several different methods and compared these methods for their agreement with subjective perception.

Based on our experimental results, we can provide several conclusions. Overall, our results show that pyramidal approaches reduce the bandwidth requirement and simultaneously provide a QoE similar to that of a full-quality, non-tiled panorama system. Furthermore, utilizing the CRF parameter of H.264 provides better bandwidth savings and better visual quality compared to up-scaling a thumbnail video when the panoramic system is static and the movement in the scene is small compared to the scene itself. This is a rather common scenario for arena sports such as rugby, soccer, hockey, and cricket. Because a subjective study is a time-consuming and expensive way to evaluate the approaches, there has been a rise in objective evaluations. We conclude that traditional evaluation methods will fail to correlate well with subjective assessments of the experience and that a new metric, OpenVQ, closely captures subjective ratings.

Both the approaches and the evaluation methods described here can be used with other interactive live PTZ camera systems; however, the tiling approaches, especially the quality levels, will require some parameter tuning specific to the application to achieve an optimal performance. Finally, we provide an open-source implementation of the OpenVQ estimation tool for further use by researchers.

REFERENCES

[1] Statista, "Top 10 Internet Traffic Services," 2015, http://www.statista.com/chart/1620/top-10-traffic-hogs/.

[2] Hypebot, "Video Wars Youtube versus Facebook," 2015, http://www.hypebot.com/hypebot/2015/03/the-video-wars-youtube-vs-facebook.html.

[3] J. Sanchez-Hernandez, J. Garcia-Ortiz, V. Gonzalez-Ruiz, and D. Muller, "Interactive streaming of sequences of high resolution jpeg2000 images," *Multimedia, IEEE Transactions on*, vol. 17, no. 10, pp. 1829–1838, Oct 2015.

[4] W.-K. Tang, T.-T. Wong, and P.-A. Heng, "A system for real-time panorama generation and display in tele-immersive applications," *Multimedia, IEEE Transactions on*, vol. 7, no. 2, pp. 280–292, April 2005.

[5] S. Tzavidas and A. Katsaggelos, "A multicamera setup for generating stereo panoramic video," *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 880–890, Oct 2005.

[6] H.-Y. Shum, K.-T. Ng, and S.-C. Chan, "A virtual reality system using the concentric mosaic: construction, rendering, and data compression," *Multimedia, IEEE Transactions on*, vol. 7, no. 1, pp. 85–95, Feb 2005.

[7] Q. Zhao, L. Wan, W. Feng, J. Zhang, and T.-T. Wong, "Cube2video: Navigate between cubic panoramas in real-time," *Multimedia, IEEE Transactions on*, vol. 15, no. 8, pp. 1745–1754, Dec 2013.

[8] E. Foote, P. Carr, P. Lucey, Y. Sheikh, and I. Matthews, "One-man-band: A touch screen interface for producing live multi-camera sports broadcasts," in *Proc. of ACM MM*, 2013, pp. 163–172.

[9] C. Fehn, C. Weissig, I. Feldmann, M. Muller, P. Eisert, P. Kauff, and H. Bloss, "Creation of high-resolution video panoramas of sport events," in *Proc. of IEEE ISM*, Dec. 2006, pp. 291 –298.

[10] P. Carr, M. Mistry, and I. Matthews, "Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording," in *Proc. of ACM MM*, 2013, pp. 193–202.

[11] X. Sun, J. Foote, D. Kimber, and B. Manjunath, "Region of interest extraction and virtual camera control based on panoramic video capturing," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 981–990, 2005.

[12] W.-K. Tang, T.-T. Wong, and P.-A. Heng, "A system for real-time panorama generation and display in tele-immersive applications," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 280–292, April 2005.

[13] Camargus, "Premium Stadium Video Technology Infrastructure," https://www.youtube.com/watch?v=SO32pEgCeDI.

[14] Teamcoco, "Conan 360: The New Angle on Late Night," 2014, http://teamcoco.com/360.

[15] T. Yokoi and H. Fujiyoshi, "Virtual camerawork for generating lecture video from high resolution images," in *Proc. of IEEE ICME*, July 2005.

[16] W. Xu and J. Mulligan, "Panoramic video stitching from commodity hdtv cameras," *Multimedia Systems*, vol. 19, no. 5, pp. 407–426, 2013.

[17] P. Carr and R. Hartley, "Portable multi-megapixel camera with real-time recording and playback," in *Proc. of DICTA*, 2009, pp. 74–80.

[18] H. Kimata, M. Isogai, H. Noto, M. Inoue, K. Fukazawa, and N. Matsuura, "Interactive panorama video distribution system," in *Proc. of ITU Telecom World*, Oct 2011, pp. 45–50.

[19] O. Niamut, J. Macq, M. Prins, R. Van Brandenburg, N. Verzijp, and P. Alface, "Towards scalable and interactive delivery of immersive media," in *Proc. of NEM Summit*, 2012, pp. 69–74.

[20] F. Liu and W. T. Ooi, "Zoomable video playback on mobile devices by selective decoding," in *Proc. of PCM*, 2012.

[21] K. Raaen, R. Eg, and C. Griwodz, "Can gamers detect cloud delay?" in *Proc. of NetGames*, 2014, pp. 1–3.

[22] I. Unanue, I. Urteaga, R. Husemann, J. D. Ser, V. Roesler, A. Rodriguez, and P. Sanchez, "A tutorial on H.264/SVC scalable video coding and its tradeoff between quality, coding efficiency and performance," in *Recent Advances on Video Coding*. Intech, 2011, pp. 3–26.

[23] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP," in *Proc. of ACM MMSys*, 2015, pp. 213–218.

[24] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," *ACM CCR*, vol. 26, pp. 117–130, 1996.

[25] K. A. Hua, Y. Cai, and S. Sheu, "Patching: a multicast technique for true video-on-demand services," in *Proc. of ACM MM*, 1998, pp. 191–200.

[26] J. Rosenberg and H. Schulzrinne, "An offer/answer model with session description protocol (SDP)," RFC 3264 (Proposed Standard), Jun. 2002.

[27] S. Wenger, U. Chandra, M. Westerlund, and B. Burman, "Codec control messages in the RTP audio-visual profile with feedback (AVPF)," RFC 5104 (Proposed Standard), Feb. 2008.

[28] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE TCSVT*, vol. 13, no. 7, pp. 637–644, 2003.

[29] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," RFC 2250 (Proposed Standard), Jan. 1998.

[30] H. Riiser, P. Halvorsen, C. Griwodz, and D. Johansen, "Low overhead container format for adaptive streaming," in *Proc. of ACM MMSys*, 2010, pp. 193–198.

[31] A. Mavlankar and B. Girod, "Video streaming with interactive pan/tilt/zoom," in *High-Quality Visual Experience*, ser. Signals and Communication Technology, 2010, pp. 431–455.

[32] R. Guntur and W. T. Ooi, "On tile assignment for region-of-interest video streaming in a wireless LAN," in *Proc. of NOSSDAV*, 2012, pp. 59–64.

[33] H. Kimata, D. Ochi, A. Kameda, H. Noto, K. Fukazawa, and A. Kojima, "Mobile and multi-device interactive panorama video distribution system," in *Proc. of GCCE*, Oct 2012, pp. 574–578.

[34] Y. Sanchez, R. Skupin, and T. Schierl, "Compressed domain video processing for tile based panoramic streaming using hevc," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2244–2248.

[35] Y. Sanchez de la Fuente, R. Skupin, and T. Schierl, "Compressed domain video processing for tile based panoramic streaming using shvc," in *Proceedings of the 3rd International Workshop on Immersive Media Experiences*. ACM, 2015, pp. 13–18.

[36] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan, "Mixing tile resolutions in tiled video: A perceptual quality assessment," in *Proc. of NOSSDAV*, 2014, pp. 25:25–25:30.

[37] Y. Takaki and N. Nago, "Multi-projection of lenticular displays to construct a 256-view super multi-view display," *Optics express*, vol. 18, no. 9, pp. 8824–8835, 2010.

[38] Y. Takaki, K. Tanaka, and J. Nakamura, "Super multi-view display with a lower resolution flat-panel display," *Optics express*, vol. 19, no. 5, pp. 4129–4139, 2011.

[39] M. P. Tehrani, T. Senoh, M. Okui, K. Yamamoto, N. Inoue, T. Fujii, and H. Nakamura, "Proposal to consider a new work item and its use case-rei: An ultra-multiview 3d display," *ISO/IEC JTC1/SC29/WG11/m30022*, 2013.

[40] M. S. Farid, M. Lucenteforte, and M. Grangetto, "A panoramic 3d video coding with directional depth aided inpainting," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3233–3237.

[41] ——, "Panorama view with spatiotemporal occlusion compensation for 3d video coding," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 205–219, 2015.

[42] V. Reddy Gaddam, H. B. Ngo, R. Langseth, C. Griwodz, D. Johansen, and P. Halvorsen, "Tiling of panorama video for interactive virtual cameras: Overheads and potential bandwidth requirement reduction," in *Picture Coding Symposium (PCS), 2015*. IEEE, 2015, pp. 204–209.

[43] K. Brunnstrom, D. Hands, F. Speranza, and A. Webster, "VQEG validation and ITU standardization of objective perceptual video quality metrics," *IEEE Signal Processing Magazine*, vol. 26, no. 3, 2009.

[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[45] ITU-T, "J.247: Objective perceptual multimedia video quality measurement in the presence of a full reference," 2008.

[46] Y. Pitrey, U. Engelke, M. Barkowsky, R. Pépion, and P. L. Callet, "Aligning subjective tests using a low cost common set," in *Proc. of EuroITV QoEMCS*, 2011.

[47] F. Boulos, W. Chen, B. Parrein, and P. Le Callet, "Region-of-interest intra prediction for H.264/AVC error resilience," in *Proc. of ICIP*, 2009, pp. 3109–3112.

[48] S. Pechard, M. Carnec, P. Le Callet, and D. Barba, "From SD to HD television: Effects of H.264 distortions versus display size on quality of experience," in *Proc. of ICIP*, 2006, pp. 409–412.

[49] R. Dosselmann and X. Yang, "A comprehensive assessment of the structural similarity index," *Signal, Image and Video Processing*, vol. 5, no. 1, pp. 81–91, 2011.

[50] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, Feb. 2004.

[51] S. Winkler, *Digital Video Quality: Vision Models and Metrics*. Wiley, 2005.

[52] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE T. Inform. Theory*, vol. 20, no. 4, pp. 525 – 536, 1974.

[53] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.

[54] ITU-T, "P.911: Subjective audiovisual quality assessment methods for multimedia applications," Geneva, pp. 1–46, 1998.

[55] F. Ribeiro, D. Florencio, and V. Nascimento, "Crowdsourcing subjective image quality evaluation," in *Prof. of ICIP*, 2011, pp. 3097–3100.

[56] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proc. of CHI*, 2008, pp. 453–456.

[57] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[58] J. Sim and C. C. Wright, "The kappa statistic in reliability studies: use, interpretation, and sample size requirements," *Physical therapy*, vol. 85, no. 3, pp. 257–268, 2005.

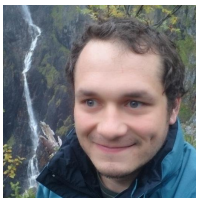
[59] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, pp. 79–86, 1951.

[60] A. Mavlankar and B. Girod, "Pre-fetching based on video analysis for interactive region-of-interest streaming of soccer sequences," in *Proc. of ICIP*, Nov 2009, pp. 3061–3064.

[61] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM TOMCCAP*, vol. 8, no. 3, 2011.

[62] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over HTTP with consistent quality," in *Proc. of ACM MMSys*, 2014, pp. 248–258.

Vamsidhar Reddy Gaddam is an engineer in the Media Processing Group at ARM. He was a PhD student at the Department of Informatics, University of Oslo and an external student member of the MPG at Simula. His research primarily focuses on the computer vision and signal processing algorithms for the Bagadus project. His PhD focused on the development of next-generation broadcasting technologies for football scenarios. He obtained his bachelor's degree in electrical engineering from the Indian Institute of Technology – Delhi with a major in communication technology in 2009.



Michael Riegler is a PhD student at Simula Research Laboratory. He received his master's degree from the Klagenfurt University with distinction. His master's thesis was about large-scale content-based image retrieval and written at the Technical University of Delft under the supervision of Martha Larson. His research interests are endoscopic video analysis and understanding, image processing, image retrieval, parallel processing, gamification and serious games, crowdsourcing, social computing and user intentions. Furthermore, he is involved in several initiatives such as the MediaEval Benchmarking initiative for Multimedia Evaluation.

Ragnhild Eg is an associate professor at Westerdals Oslo School of Arts, Communication and Technology. She obtained her bachelor's degree in psychology and journalism from the University of Queensland and completed her master's in cognitive and biological psychology at the Norwegian University of Science and Technology. Her previous work focused on the integration of auditory and visual speech cues and the consequences of losing perceptual information due to quality distortions. Her research interests are related to human perception, multisensory integration and multimedia quality. Future research directions include spatial and depth perception in audiovisual 3D-scenarios and further explorations of video coding artefacts.

Carsten Griwodz is a senior researcher at the Norwegian research company Simula Research Laboratory AS, Norway, and a professor at the University of Oslo. He received his Diploma in Computer Science from the University of Paderborn, Germany, in 1993. From 1993 to 1997, he worked at the IBM European Networking Center in Heidelberg, Germany. In 1997 he joined the Multimedia Communications Lab at Darmstadt University of Technology, Germany, where he obtained his doctoral degree in 2000. He joined the University of Oslo in 2000 and Simula Research Laboratory in 2005. Carsten's research interest lies in the performance of multimedia systems, particularly streaming media, which includes all types of media that are transported over the Internet with a temporal demand, including stored and live video as well as games and immersive systems. To achieve this, he wants to advance operating system and protocol support, parallel processing and the understanding of the human experience. Currently, he is working on a freely available full-reference video quality assessment tool inspired by the work of the ITU-T's VQEG working group and on a new evaluation of caching strategies in a DASH world. He is currently editor-in-chief of the ACM SIGMM Records.

Pål Halvorsen is a senior researcher at Simula Research Laboratory and a professor at the Department of Informatics, University of Oslo, Norway. He received his his doctoral degree (Dr. Scient.) in 2001 from the Department of Informatics, University of Oslo, Norway. His research focuses mainly on distributed multimedia systems, including operating systems, processing, storage and retrieval, communication and distribution.