Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP

Design, Implementation, and Evaluation

Mario Graf
Bitmovin Inc.
301 Howard Street, Suite 1800
San Francisco, California 94105
mario.graf@bitmovin.com

Christian Timmerer
Alpen-Adria-Universität Klagenfurt
/ Bitmovin Inc.
Universitätsstraße 65-67
9020 Klagenfurt, Austria
christian.timmerer@itec.aau.at

Christopher Mueller
Bitmovin Inc.
301 Howard Street, Suite 1800
San Francisco, California 94105
christopher.mueller@bitmovin.com

ABSTRACT

Real-time entertainment services such as streaming audio-visual content deployed over the open, unmanaged Internet account now for more than 70% during peak periods. More and more such bandwidth hungry applications and services are proposed like immersive media services such as virtual reality and, specifically omnidirectional/360-degree videos. The adaptive streaming of omnidirectional video over HTTP imposes an important challenge on today's video delivery infrastructures which calls for dedicated, thoroughly designed techniques for content generation, delivery, and consumption.

This paper describes the usage of tiles — as specified within modern video codecs such HEVC/H.265 and VP9 — enabling bandwidth efficient adaptive streaming of omnidirectional video over HTTP and we define various streaming strategies. Therefore, the parameters and characteristics of a dataset for omnidirectional video are proposed and exemplary instantiated to evaluate various aspects of such an ecosystem, namely bitrate overhead, bandwidth requirements, and quality aspects in terms of viewport PSNR. The results indicate bitrate savings from 40% (in a realistic scenario with recorded head movements from real users) up to 65% (in an ideal scenario with a centered/fixed viewport) and serve as a baseline and guidelines for advanced techniques including the outline of a research roadmap for the near future.

CCS CONCEPTS

•Information systems →Multimedia streaming; Multimedia content creation; •Networks →*Network experimentation; Network measurement;*

KEYWORDS

MPEG-DASH, Omnidirectional Video, 360-degree video, immersive media, HEVC/H.265 tiles, tiled streaming

ACM Reference format:
Mario Graf, Christian Timmerer, and Christopher Mueller. 2017. Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP. In *Proceedings of MMSys'17, Taipei, Taiwan, June 20-23, 2017,* 11 pages.
DOI: http://dx.doi.org/10.1145/3083187.3084016

1 INTRODUCTION

Universal media access [12] as proposed in the late 90s, early 2000 is now reality. It is very easy to generate, distribute, share, and consume any media content, anywhere, anytime, and with any device. These kind of real-time entertainment services — specifically, streaming audio and video — are typically deployed over the open, unmanaged Internet and account now for more than 70% of the evening traffic in North American fixed access networks. It is assumed that this number will reach 80% by the end of 2020 [17]. A major technical breakthrough and enabler was certainly the adaptive streaming over HTTP resulting in the standardization of MPEG-DASH [19, 20].

One of the next big things in adaptive media streaming is most likely related to virtual reality (VR) applications and, specifically, omnidirectional (360-degree) media streaming, which is currently built on top of the existing adaptive streaming ecosystems.

Omnidirectional video (ODV) content allows the user to change her/his viewing direction in multiple directions while consuming the video, resulting in a more immersive experience than consuming traditional video content with a fixed viewing direction. Such video content can be consumed using different devices ranging from smart phones and desktop computers to special head-mounted displays (HMD) like Oculus Rift, Samsung Gear VR, HTC Vive, etc. When using a HMD to watch such a content, the viewing direction can be changed by head movements. On smart phones and tablets, the viewing direction can be changed by touch interaction or by moving the device around thanks to built-in sensors. On a desktop computer, the mouse or keyboard can be used for interacting with the omnidirectional video.

MMSys'17, Taipei, Taiwan

978-1-4503-5002-0/17/06...$15.00
DOI: http://dx.doi.org/10.1145/3083187.3084016

The streaming of ODV content is currently deployed in a naive way by simply streaming the entire 360-degree scene/view in constant quality without exploiting and optimizing the quality for the user's viewport. This approach is referred to as *monolithic* streaming of ODV content. Region of interest (ROI) based coding has been proposed as a promising candidate to be adopted for adaptive streaming use cases but lacks of native support in state-of-the-art video codecs such as AVC/H.264 or VP8 but, fortunately, is fully supported within HEVC/H.265 or VP9 and referred to as *tiles*. Tiles divide a video picture/frame into regular-sized, rectangular regions which are independently decodable, enable efficient parallel processing, and provides entry points for local access. However, encoding and streaming options utilizing tiles for adaptive HTTP streaming are not yet adequately described in the literature. In this paper, we describe basic principles of adaptive tile-based streaming of omnidirectional video services over HTTP, available encoding options, and evaluations with respect to bitrate overhead, bandwidth requirements, and quality aspects.

The remainder of this paper is structured as follows. Related work and background information is described in Section 2. The system architecture and options for tile-based adaptive streaming of omnidirectional video over HTTP is described in Section 3 and implementation details used for the evaluation are briefly highlighted in Section 4. The evaluation and its results are described in Section 5. Finally, conclusions and future work items are provided in Section 6.

2 BACKGROUND AND RELATED WORK

2.1 Background Overview

The basic system architecture including major interfaces of an omnidirectional video ecosystem is shown in Figure 1. It typically starts with multiple videos being captured including various metadata ①, stitched together, and further edited before entering the encoding process ②. The encoding -- typically a single video – considers projection and interactivity metadata and utilizes an appropriate storage and/or delivery format (including possibly encryption) ③ before it will be decoded on the target device. After decoding – again, typically a single video –, various projection and interactivity metadata ④ will guide the rendering process which interacts with the corresponding input/output technology (such as HMDs) ⑤. The focus of this paper is on the encoding and adaptive streaming.

In the past, various projection formats have been proposed (e.g., equirectangular, cube maps, pyramid maps, frustum maps, equal-area projection) [5, 25] while currently, in practice, mainly equirectangular projection is used. Equirectangular projection adopts a constant spacing of latitudes and longitudes which allows for simple, efficient processing but introduces horizontal stretching in the projected panorama near the poles. It is supported in most of the available content generation tools (i.e., camera hardware + stitching software) which explains its popularity.

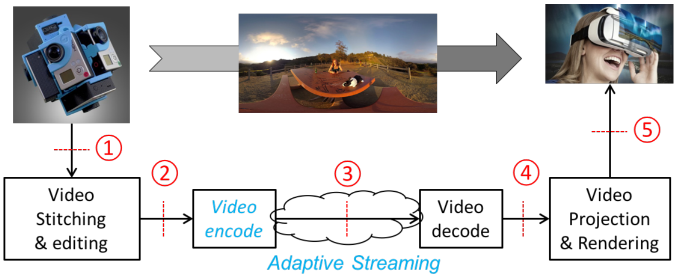

Figure 1: Basic System Architecture and Interfaces for Omnidirectional Video Streaming.

As of today, no special techniques for coding in the spherical domain of the video exists and, thus, the video is projected to the rectangular domain. Therefore, state of the art video codecs (AVC/H.264, HEVC/H.265, VP8, VP9) and delivery formats (DASH/HLS) can be used to deploy a basic adaptive streaming service of omnidirectional video content. However, this is very inefficient as the typical Field of View (FoV) of many VR devices is limited and a lot of content is delivered, decoded, and rendered for nothing (e.g., what is happening outside of the users' FoV). Viewport adaptive streaming has been introduced to overcome this limitation but requires multiple versions of the same content for each view. That is, it adopts a similar strategy as in adaptive media streaming (DASH/HLS) but the number of versions of the same content significantly increases which impacts (cloud) storage and (content delivery) network costs (see further details in Section 2.3). A novel approach in this domain utilizes the concept of video tiles - as part of the HEVC standard [21] - and the Spatial Relationship Descriptor (SRD) of the MPEG-DASH standard [13] to enable efficient adaptive streaming of omnidirectional media services. Please note that replacing a tile with a tile of a different quality requires the usage of HEVC tiling tools with constrained motion. We will further discuss and evaluate possible options for the encoding and streaming in this paper.

2.2 Overview of Standardization Activities

JPEG started an initiative called *Pleno* [4] focusing on images but our focus is on video and, thus, we will concentrate on standards related to video. In this context, MPEG started a new work item related to *immersive media* officially referred to as ISO/IEC 23090 which – at the time of writing of this paper – foresees five parts.

The first part will be a technical report describing the scope of this new standard and a set of use cases and applications from which actual requirements can be derived. Technical reports are usually publicly available for free. The second part specifies the omnidirectional media application format (OMAF) [2] addressing the urgent need of the industry for a standard is this area. Part three will address immersive video and part four defines immersive audio. Finally, part five will

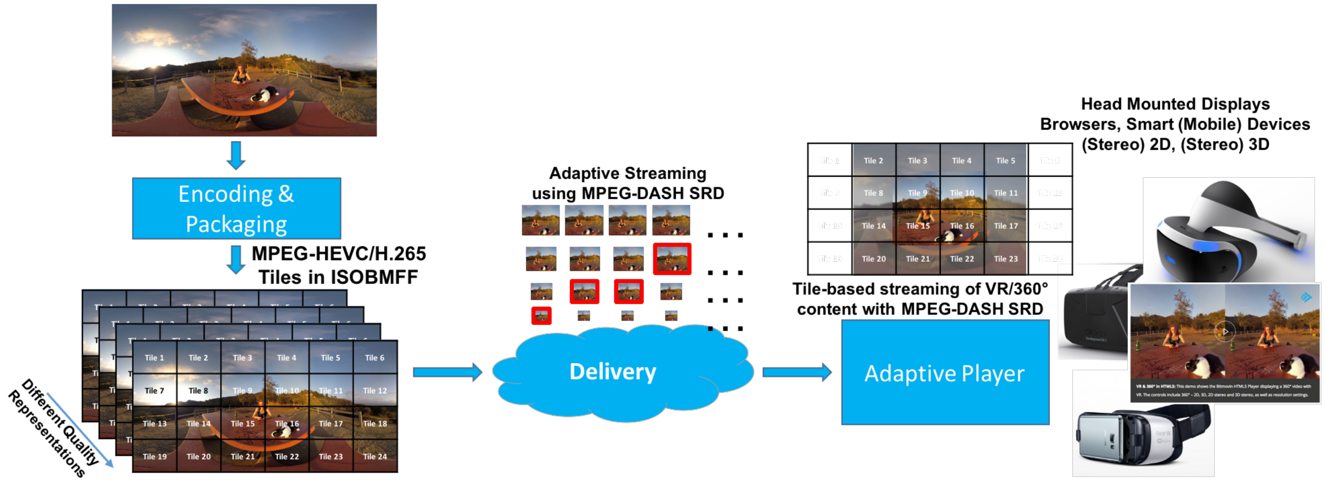

Figure 2: System Architecture for Bandwidth Efficient Tiled Streaming.

contain a specification for point cloud compression for which a call for proposals is currently available. OMAF is part of a first phase of standards related to immersive media and should finally become available by the end of 2017, beginning of 2018 while the other parts are scheduled at a later stage around 2020. The current OMAF committee draft comprises a specification of the *i)* equirectangular projection format (note that others might be added in the future), *ii)* metadata for interoperable rendering of 360-degree monoscopic and stereoscopic audio-visual data, *iii)* storage format adopting the ISO base media file format (ISOBMFF/mp4), and *iv)* the following codecs: MPEG-H High Efficiency Video Coding (HEVC) and MPEG-H 3D audio.

The Spatial Relationship Descriptor (SRD) of the MPEG-DASH standard [13] provides means to describe how the media content is organized in the spatial domain. In particular, the SRD is fully integrated in the media presentation description (MPD) of MPEG-DASH and is used to describe a grid of rectangular tiles which allows a client implementation to request only a given region of interest — typically associated to a contiguous set of tiles. Interestingly, the SRD has been developed before OMAF and how SRD is used with OMAF is currently subject to standardization.

Finally, WebVR [23] defines an API which provides support for accessing virtual reality devices, including sensors and head-mounted displays on the web. It is currently available in Firefox nightly builds, in Chrome 56+ for Android and experimental builds of Chromium for Windows, and in the Samsung Internet Browser for Gear VR.

2.3 Related Work

The description of basic tiled streaming can be found in [13] and a demo is described in [9]. In both cases basic principles are discussed which served as a motivation for this paper focusing on various encoding and streaming options including evaluations thereof. Earlier work in this domain adopted MPEG-4 video or AVC/H.264 as HEVC/H.265 was not yet available [3, 6, 8, 14, 15]. In AVC/H.264 slices where used to implement tiles as they share the same coding principles except that slides do not have to be regular-sized/rectangular. In our work, we focus on HEVC/H.265 and utilize the built-in tile feature.

In 2016 Facebook proposed a pyramid geometry and applied it to 360-degree video which shall reduce file size by 80% [7]. However, no further details or scientific evaluations have been provided and it seems to be impractical as it requires multiple versions, i.e., a total of 150 different versions of the same video, and the impact on storage and network requirements is unknown. Skupin et al. [18] demonstrated the use cube maps utilizing tiles whereas our approach is based on the equirectangular projection format. In [16] authors adopted scalable extensions of HEVC/H.265 for the streaming of 360-degree content. Zare et al. [26] is closely related to our approach. They also adopt HEVC/H.265 tiles and proposed different tiling schemes including a preliminary evaluation. In this paper, we investigated additional tiling patterns with a more detailed evaluation of different streaming scenarios including a realistic deployment setup and various evaluation parameters leading to a better encoding and streaming performance.

Finally, existing objective metrics such as PSNR are known for their limitations as QoE metrics and are even more controversial for omnidirectional video. However, in the past, spherical PSNR (S-PSNR) and viewport PSNR (V-PSNR) [25] have been proposed which can be used with Bjøntegaard Delta [1] known from traditional video applications. In this paper, we adopt V-PSNR as an evaluation metric and further details are provided in Section 5.

3 BANDWIDTH EFFICIENT TILED STREAMING

The system architecture enabling bandwidth efficient tiled streaming is depicted in Figure 2. The video will be encoded and packaged in the rectangular domain utilizing tiles from HEVC/H.265 and is available in multiple quality representations. The MPEG-DASH SRD is used to describe the tile structure which enables the client to request tiles and quality representations depending on the context conditions including the Field of View (FoV). That is, individual tiles can be requested from different quality representations (e.g., those within the FoV with highest possible quality and neighboring, adjacent tiles with lower quality) or not at all as indicated in Figure 2. For the actual decoding, the individual tiles need to be re-/transmultiplexed into a single, standard-compliant HEVC/H.265 bitstream as typically only a single decoder is available at today's client device platforms. Note that this might change in the (near) future.

In the following we describe basic streaming strategies which provide the basis for our evaluation. The streaming strategies can be divided into two categories, namely *full delivery* and *partial delivery*. Full delivery provides all tiles of a frame to the client resulting in a full frame without any holes whereas partial delivery allows that some tiles (of a frame) are not delivered at all (i.e., those outside the current viewport). Based on that we can define a variety of streaming strategies as depicted in Figure 3. The figure shows three possible strategies based on two segments which are downloaded over time. Each frame is divided according to a given tiling pattern (4 × 3 in this example) and the viewport is indicated using a red rectangle.

Full Delivery Basic: All tiles which are visible in the user's current viewport are requested in the *highest possible quality representation* (green tiles) while all other tiles, which are not visible at the moment, are requested in the *lowest available quality representation* (red tiles). In general, the bitrate of the highest possible quality representation for the tiles within the current viewport depend on the available bandwidth and the bitrate of the tiles outside of the current viewport. In the best case it is the same as the highest available quality representation and in the worst case it is the lowest available quality representation. This strategy shall serve as a basis (benchmark) for all advanced strategies.

Full Delivery Advanced: This strategy is an improvement of the *full delivery basic* approach with various options. One possibility could be to request all tiles around the visible viewport in an lower (but not lowest) quality since these are the parts of the video which are visible when the user's viewport starts to move in any direction. A more sophisticated approach would be to predict the user's viewport movement and to request the corresponding tiles in a higher quality than others as shown in Figure 3. For example, yellow tiles are requested in a higher quality as it is expected that the user's viewport will move into that direction. Details and further variations of this advanced strategy are subject to future work.

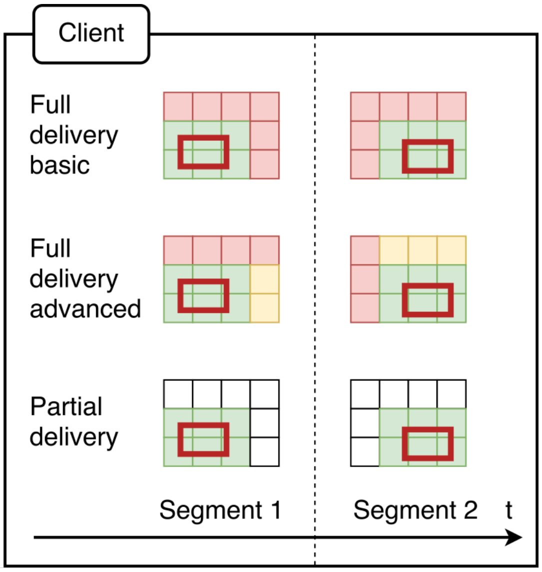

Figure 3: Tiled Streaming Strategies at the Adaptive Streaming Client requesting two Segments over Time.

Partial Delivery: This strategy reduces the amount of the delivered data even more. In particular, the streaming client requests the tiles within the user's current viewport in the highest possible quality representation (green tiles) and all other tiles (i.e., those outside the current viewport) are not requested at all, remain on the server (white tiles) and, thus, the available bandwidth is consumed solely by those tiles within the user's current viewport. However, (fast or unexpected) user head motions could lead to the rendering of blank areas (or corresponding tiles are rendered with delay) which expectedly decreases the Quality of Experience (QoE) for the user. Thus, this strategy is merely considered as impractical but could serve as a benchmark to show what can be achieved at most.

All above strategies have pros and cons, specifically when taking into account different network characteristics (available bandwidth but most importantly delay) and user interactivity (fast versus slow viewport movement/interactivity). In this paper, the main focus is on multimedia systems integration to enable the above strategies using today's device platforms and perform a benchmark evaluation enabling advanced options in the future.

4 IMPLEMENTATION

This section briefly highlights the tools developed in the course of this paper and adopted for the evaluation: *libVR*, *tileMuxer*, *tileTools*, and *tiled player* as native Android app and for web/HTML5 environments.

4.1 lilbVR

The *libVR* is a library offering multiple functionalities in the context of ODV such as vector and matrix operations, identifying the visible tiles for a given viewport configuration, and determining the current viewing direction. One of the core functionalities is the multithreaded calculation of PSNR and viewport PSNR respectively. Therefore, a pinhole camera model is used to calculate the projection from spherical coordinates to viewport coordinates. It is implemented as a Java library including a command-line interface wrapper and is used as a library in the Android-based tiled player.

4.2 tileMuxer

The *tileMuxer* allows for preprocessing of tiled HEVC/H.265 streams, i.e., splitting network abstraction layer units (NALUs) containing multiple tiles into NALUs where each contains one tile. The main functionality is *i)* extracting individual tiles from tiled HEVC/H.265 files and storing them as separate files and *ii)* combining tiles with potentially different qualities into a single tiled HEVC/H.265 file. The tileMuxer supports plain HEVC/H.265 streams and ISOBMFF-packaged HEVC/H.265 files. It is implemented as a Java library including a command-line interface wrapper and is used as a library in the Android-based tiled player.

4.3 tileTools

The *tileTools* comprise a set of tools for visualizing tiles: *a)* visualize tiles for a given viewport; *b)* visualize recorded head movements; and *c)* the generation of a video according to the recorded head movements.

4.4 Android-based Tiled Player

The Android-based tiled player runs natively on Android and supports MPEG-DASH SRD. It utilizes the tileMuxer to provide single HEVC/H.265 segments — composed from multiple tiles — into a single HEVC/H.265 decoder instance to exploit the hardware-accelerated decoding of the underlying platform. It implements a simple adaptation logic using libVR where all tiles visible in current viewport are requested in the highest possible quality and all other tiles are requested in lowest available quality which in total matches the available bandwidth. Please note that the adaptation logic is not (yet) optimized and adopts only a very basic buffer management and throughput measurements.

4.5 Web-based Tiled Player

Finally, we have implemented a web-based tiled player in Javascript using the HTML5 Media Source Extensions (MSE). The implementation is a proof-of-concept as implementing adaptive HTTP streaming in HTML with HEVC/H.265 has one major drawback. It requires explicit support of HEVC/H.265 within the web browser and the underlying hardware that is currently only available within the Microsoft Edge browser on a limited set of end user devices. We expect this to change in the future.

5 EVALUATION

This section describes the evaluation setup and evaluation results comprising quality metrics used in the context of OVD, a basic dataset and how it has been generated, and the actual results in terms of *bitrate overhead* (due to various tiling patterns), *bandwidth requirements*, and *quality* (adopting viewport PSNR).

5.1 Quality Metrics for ODV

In principle, objective quality metrics like standard PSNR or SSIM can be also used for ODV content by applying it directly on ODV frames in the rectangular domain as long as the same projection method is used. In case difference project formats are used (e.g., equirectangular and cube maps), metrics like PSNR cannot be used anymore which calls for a metric to be applied in the spherical domain. Therefore, *spherical PSNR (S-PSNR)* was introduced by Yu et al. [25] which defines a uniform grid of sampling points on the sphere. For each of these sampling points, the corresponding pixels in the rectangular domain are calculated which allows the usage of the standard PSNR calculation.

Unfortunately, S-PSNR has some limitations when using tiled streaming adopting the streaming strategies as proposed in Section 3 which requires to take into account the viewport for the PSNR calculation. In particular, when the tiles outside the viewport have a lower quality (or are not available at all), the PSNR cannot be compared with non-tiled, monolithic ODV content. Therefore, Yu et al. [25] proposed viewport PSNR (V-PSNR) adopting the same principles as for S-PSNR but taking only the pixels of a given viewport into account. Finally, with V-PSNR we can also use Bjøntegaard Delta [1].

5.2 Dataset

For the evaluation we have defined a dataset of ODV content with various parameters which influences the performance and quality of streaming and playback of tiled ODV content. The parameters and its instantiation for this paper are described in the following.

Segment size / intra period. Segment size is an important parameter for adaptive HTTP streaming and is typically measured in seconds worth of video content. Shorter segments allow for faster quality adaptation to changing context conditions including the user's viewport but reduces coding efficiency due to a higher frequency of intra (only) frames. In combination with small video buffers — allows for better interactivity — it may lead to more buffer underruns and, thus, stalls which impacts the Quality of Experience (QoE). For our evaluation of tiled streaming we selected a short segment size of 1s (at 25fps) to account for interactivity and compared it with non-tiled, monolithic streaming using 1s, 2s, and 4s segment sizes (at 25fps). In the literature, 4s segment size has been reported as a good tradeoff between streaming and coding efficiency [10].

Table 1: BD-PSNR for different Tiling Patterns w.r.t. Tiled Monolithic (1×1).

		BD-PSNR [dB]			
Sequence	Resolution	3 × 2	5 × 3	6 × 4	8 × 5
A.-Creed	1920x960	-0.328	-0.887	-1.260	-1.842
A.-Creed	3840x1920	-0.163	-0.504	-0.726	-1.091
A.-Creed	7680x3840	-0.064	-0.258	-0.376	-0.577
E.-World	1920x960	-0.185	-0.451	-0.607	-0.875
E.-World	3840x1920	-0.131	-0.265	-0.374	-0.513
E.-World	7680x3840	-0.092	-0.170	-0.239	-0.320

Table 2: BD-BR for different Tiling Patterns w.r.t. Tiled Monolithic (1×1).

		BD-BR [%]			
Sequence	Resolution	3 × 2	5 × 3	6 × 4	8 × 5
A.-Creed	1920x960	8.542	24.456	36.101	55.833
A.-Creed	3840x1920	4.034	12.833	18.929	29.515
A.-Creed	7680x3840	1.529	6.115	9.027	14.099
E.-World	1920x960	4.459	10.970	14.974	21.983
E.-World	3840x1920	2.925	5.965	8.493	11.778
E.-World	7680x3840	1.885	3.497	4.942	6.645

Tiling pattern. The tiling pattern also impacts the coding efficiency of HEVC/H.265 as tiles are encoded independent of other tiles. That is, larger tiles provide a better coding efficiency but less flexibility for viewport selection and smaller tiles provide a better match to a given viewport but, consequently, reduce coding efficiency. In this paper, we adopt the following tiling patterns (columns × rows): 1 × 1 (i.e., tiles monolithic), 3 × 2, 5 × 3, 6 × 4, and 8 × 5.

Spatial resolution. We adopted the following spatial resolutions ranging from high- to ultra high-definition which is suitable for ODV content: 1920 × 960, 3840 × 1920 and 7680 × 3840. In practice, however, for achieving a resolution of 4K for a viewport with 120° Field of View (FoV), the resolution of the entire frame would be in the range of 12K × 4K which is impractical to realize with currently available coding tools.

Map projection. Although many projection formats are available, we selected the equirectangular format as it is the only format which is widely supported and deployed.

Encoding parameters. We used quantization parameter (QP) ranging from 22 to 42 in steps of five leading to five different bitrate versions allowing for sufficient Bjøntegaard Delta calculations.

Source (SRC) video sequences. Due to the lack of freely available ODV content we simply downloaded two sequences from YouTube: *ExploreTheWorld*[1] and *AssassinsCreed*[2]. The latter comprises computer-generated content whereas the former is a documentary. Together, they represent a broad range of content genres. The total number of content configurations is 210 sequences. The encoding is done with the Kvazaar encoder [22].

Viewport-based ODV content. For each video sequence (ExploreTheWorld and AssassinsCreed) we recorded the viewing directions (head movements) from three different users consuming the content using a HMD. In particular, three head movements were recorded for 1920 × 960 and one head movement for the other two resolutions. No special instructions were given when users consumed the content resulting in natural user interactivity (head movements). These recordings were used to generate videos simulating the streaming strategy *full delivery basic* as described in Section 3, i.e., tiles within the viewport (based on the viewing direction of the actual users) are encoded with higher quality (QP={22,27,32,37,42}) whereas tiles outside the viewport are always encoded with the lowest available quality (QP 42). The total number of video configurations using the recorded head motions is 140.

5.3 Results: Bitrate Overhead

The goal of the first evaluation was to analyze the coding overhead introduced by using HEVC/H.265 tiles to encode ODV content. HEVC/H.265 tiles [11] divides the video frame into rectangular regions where each tile is independently coded and, thus, a frame could be composed by tiles from different quality representations when used in the context of adaptive HTTP streaming. For this evaluation we used all three spatial resolutions and the Bjøntegaard-Delta (BD) [1] (PSNR, Bitrate (BR)) to compare the monolithic tiling (i.e., 1 × 1) with all other tiling patterns. The results of the BD-PSNR and BD-BR are shown in Table 1 and Table 2 respectively. A negative BD-PSNR means a lower PSNR than using monolithic tiling and a positive BD-BR means a higher bitrate compared to monolithic tiling. As expected, the bitrate overhead growths with an increasing number of tiles but differences between the tiling patterns decrease with higher resolutions. Thus, we show the rate-distortion (RD) curves of both sequences for 1920 × 960 only which are depicted in Figure 4 and Figure 5 respectively. The RD curves look different for the two sequences due to the different genre of the content, i.e., the computer-generated content achieves higher PSNR at lower bitrate than the documentary.

The tiling pattern 3 × 2 provides the lowest overhead but obviously has less flexibility with respect to viewport selection whereas 8 × 5 results in a very high overhead, specifically at lower resolutions. For example, when comparing 1 × 1 and 8 × 5 in Figure 4, it shows a difference of approximately 1.2 dB at 1,500 kbps and approximately +37.3% bitrate at 45 dB. The tiling patterns 5 × 3 and 6 × 4 are somewhat comparable but the latter offers more flexibility due to a higher number of tiles. Thus, we will further investigate 6 × 4 and also 3 × 2 regarding bandwidth requirements.

[1] https://www.youtube.com/watch?v=1_ifgJqLqTY, first 120s.

[2] https://www.youtube.com/watch?v=g0AYnMPkg2k, 18s-128s as first 18s comprises very dark content.

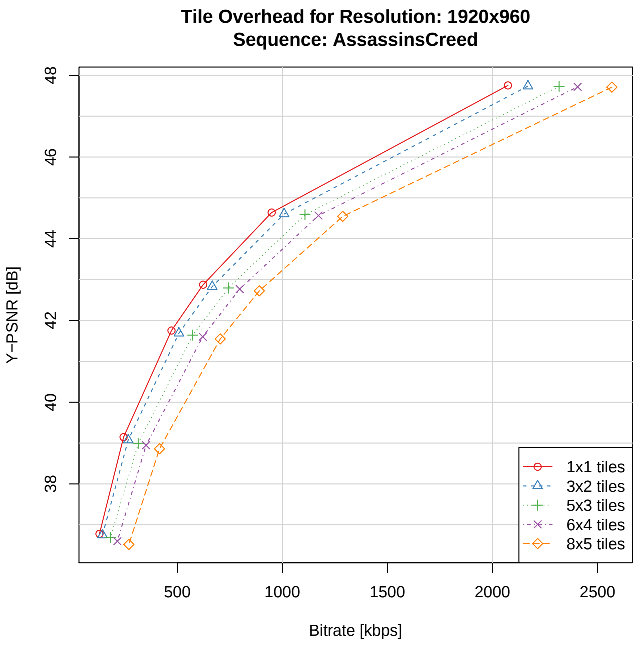

Figure 4: Tile Overhead *AssassinsCreed* 1920 × 960.

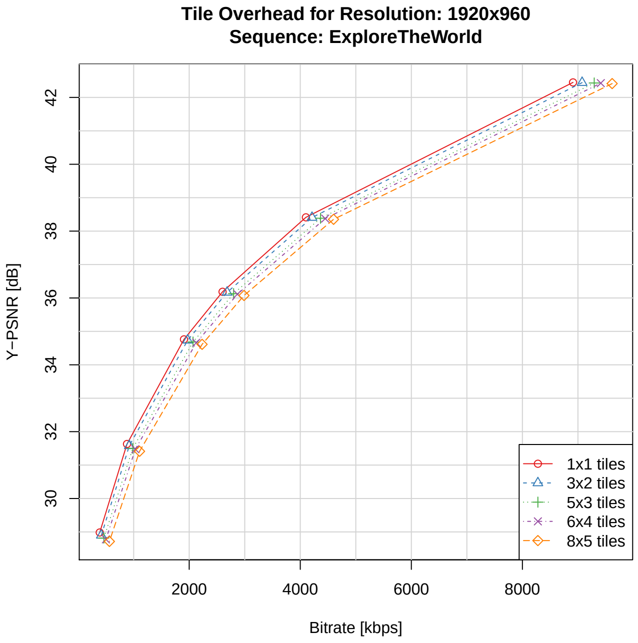

Figure 5: Tile Overhead *ExploreTheWorld* 1920 × 960.

5.4 Results: Bandwidth Requirements

In this evaluation we compare monolithic streaming to tiled streaming based on the bandwidth requirements. For the monolithic streaming approach we evaluate all three segments sizes / intra periods, namely 1s, 2s, and 4s. For the tiled streaming we investigate two of the streaming strategies introduced in Section 3: *full delivery basic* and *partial delivery*.

In a first evaluation, the viewport is predefined at a pitch and yaw angle of 0° and a horizontal FoV of 96°. Using this configuration the viewport is centered at the equator of the equirectangular video frame which provides the lowest distortion. Furthermore, the content at high quality is encoded with QP 27 whereas at low quality QP 42 is used. For monolithic streaming, only high quality is used and for tiled streaming, the high quality is used only for tiles visible in the defined viewport and remaining tiles use low quality (full delivery basic) or no tiles at all (partial delivery).

Since we use a static viewport for this evaluation, the setup represents an ideal scenario. All results obtained in this evaluation can be seen as the upper bound of what is possible by using the described tiled streaming approaches.

The results for the *ExploreTheWorld* sequence using tile patterns 3 × 2 and 6 × 4 are shown in Figure 6. Obviously, the bandwidth requirements for tiled streaming with partial delivery strategy provide the best results but are impractical except probably in a low/zero-delay environment. On the other hand, monolithic streaming requires more bandwidth but the bandwidth requirements decreases with increasing segment sizes due to the coding efficiency of larger intra periods. The bare tiling overhead can be seen when comparing *monolithic 1sec* with *tiles monolithic* which uses the high quality (QP 27) for all tiles. It increases slightly with increasing spatial resolutions. Interestingly, the streaming strategy full delivery basic significantly reduces the bandwidth requirements, specifically for higher spatial resolutions and with tiling pattern 6×4. Thus, we will further investigate this tiling pattern, also for our subsequent viewport PSNR analysis.

Before discussing V-PSNR we will specifically compare monolithic streaming (4s segment size) with the streaming strategies *partial delivery* and *full delivery basic* as shown in Figure 7. It clearly shows that partial delivery can reduce the bandwidth requirements by more than 75% compared to the current, state-of-the-art deployments in omnidirectional video streaming. However, as partial delivery is somewhat impractical, the purpose of this result is mainly to show the potential of tiled streaming. The streaming strategy full delivery basic achieves a bandwidth saving by approximately 65% and can be seen as a benchmark for any advanced mechanisms which are subject to future work.

A summary of the results for all resolutions and tiling patterns can be found in Table 3. Note that 'positive' values indicate an overhead whereas 'negative' values show actual savings. The results clearly identify that the tiling pattern 6 × 4 is a promising configuration for such use cases as it shows highest bitrate savings across all resolutions whereas the tiling pattern 3 × 2 has the lowest bitrate savings.

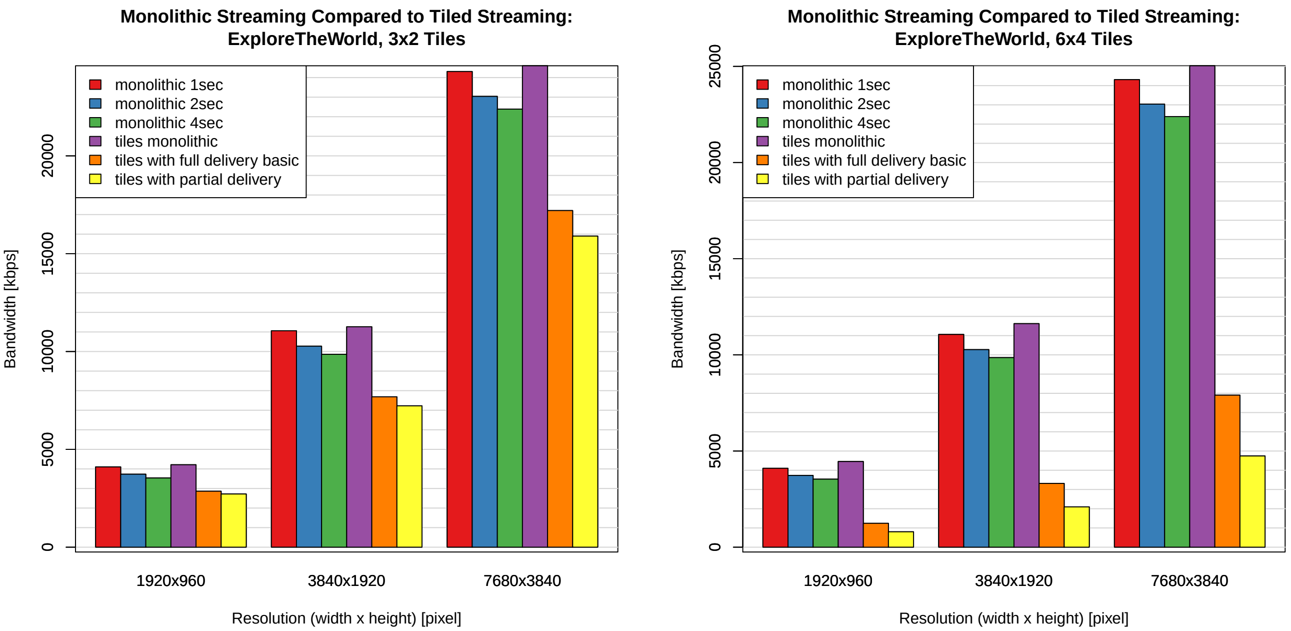

Figure 6: Bandwidth Requirements for Monolithic Streaming compared to Tiled Video Streaming for the Sequence *ExploreTheWorld* and for Tiling Patterns 3×2 (left) and 6×4 (right).

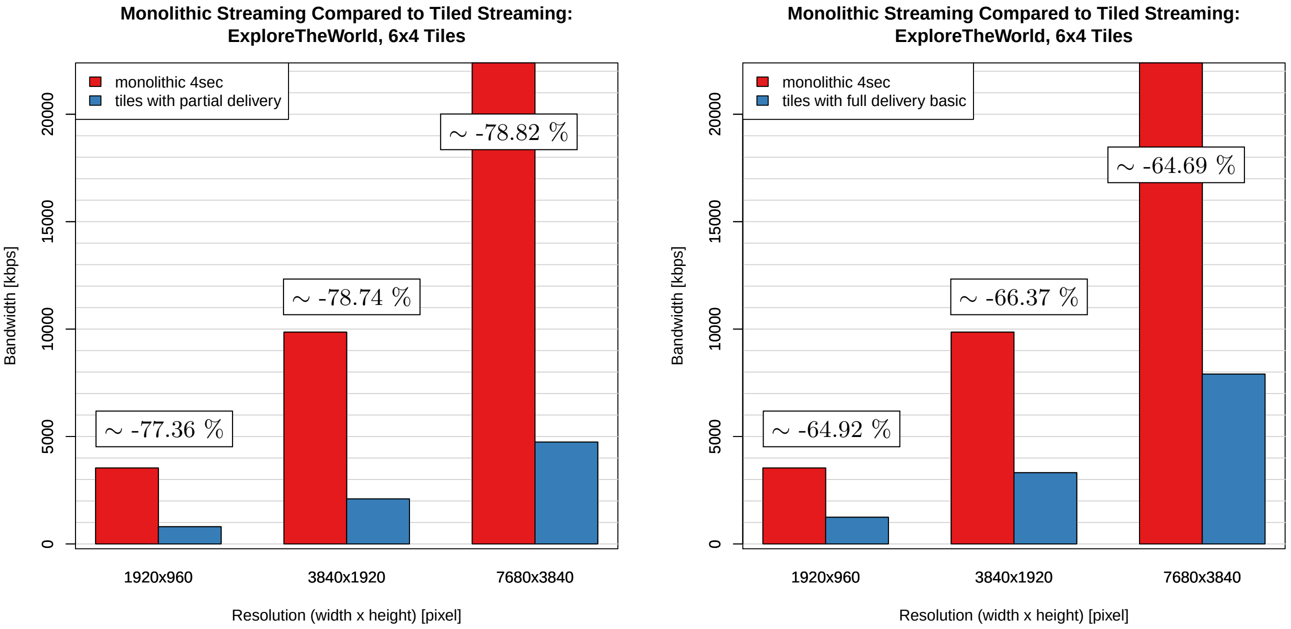

Figure 7: Comparison of Monolithic Streaming with a Segment Size / Intra Period of 4sec to Tiled Streaming using *Partial Delivery* (left) and *Full Delivery Basic* (right) strategy for *ExploreTheWorld* and Tiling Pattern 6×4.

Table 3: Bitrate Savings in Percent Relative to Monolithic Video for Different Resolutions and Tiling Patterns for the Sequence *ExploreTheWorld*.

		Monolithic [kbps]	Tiles, Bitrate Saving [%]		
Resolution	Tiling	Monolithic 4s	Tiles Monolithic	**Full Delivery Basic**	Partial Delivery
1920x960	3x2	3,537.32	19.13	**-19.14**	-23.12
1920x960	5x3	3,537.32	23.51	-42.81	-50.73
1920x960	6x4	3,537.32	25.93	**-64.92**	-77.36
1920x960	8x5	3,537.32	30.28	-45.47	-57.55
3840x1920	3x2	9,857.76	14.28	**-22.04**	-26.72
3840x1920	5x3	9,857.76	16.25	-45.91	-54.36
3840x1920	6x4	9,857.76	17.92	**-66.37**	-78.74
3840x1920	8x5	9,857.76	19.92	-49.25	-60.21
7680x3840	3x2	22,390.45	9.89	**-23.16**	-28.99
7680x3840	5x3	22,390.45	10.90	-46.20	-56.36
7680x3840	6x4	22,390.45	11.80	**-64.69**	-78.82
7680x3840	8x5	22,390.45	12.77	-50.36	-62.48

5.5 Results: Viewport PSNR

The final evaluation compares the viewport PSNR (V-PSNR) based on the recorded head movements to demonstrate a realistic environment. We used again the sequence *ExploreTheWorld* but at different resolutions and generated a set of tiled videos using *tileMuxer* which takes into account the recorded head movements. Since we evaluate tiled streaming using adaptive HTTP streaming (DASH/HLS), the client can only adapt the quality of the tiles on segment boundaries. That is, when the user moves her/his head directly at the beginning of a new segment, it takes minimum one segment length until the client is able to switch to a higher quality for the tiles within the concerned viewport. As a consequence, potentially lower quality tiles or blank areas are presented to the user depending on the tiled streaming strategy.

The results of the V-PSNR of a given head movement comparing monolithic streaming with tiled streaming (with 6×4 tiling pattern) at resolutions 1920×960 and 3840×1920 are shown in Figure 8. The results for *tiles monolithic* (i.e., 6×4 tiling pattern with constant quality) is at the lower end due to the tile overhead as already discussed above. It can be seen as a lower threshold for this content in the given configuration. Obviously, larger segment sizes provide better results due to increased coding efficiency as shown in the RD curves for *monolithic 1s, 2s, and 4s.* Interestingly, the streaming strategy using tiles with full delivery basic shows best results, specifically more than 40% (at QP 27 for tiles visible within the viewport) compared to monolithic 4s which corresponds to a practical, state-of-the-art deployment. For the sequence AssessinsCreed reaches more than 55% (at QP 27 for tiles visible within the viewport) but only for the 8K resolution (not shown here). Please note that the streaming strategy with partial delivery is not shown here as we believe it is impractical as V-PSNR might be calculated using potentially blank tiles depending on the user's head movement.

Zare et al. [26] report bitrate savings from 30% to 40% depending on the tiling scheme. However, they adopted a different tiling schemes assuming a common user behavior which requires subjective studies not yet conducted. Additionally, they have a reduced set of QPs (22, 26, 30, 34) compared to our configuration (22, 27, 32, 37, 42). Therefore, we excluded QP 42 from the BD-BR calculation and the summary of the BD-BR from all head movement recordings, spatial resolutions, and tiling patterns is shown in Table 4. FrameLog 1-3 denote the different head movement recordings from the three users. Similar as Zare et al., our results confirm a bitrate saving up to 40%, specifically for the tiling pattern 6×4 and it seems that using different tilling schemes (e.g., as suggested by Zare et al.) has only a small impact (if any at all).

6 CONCLUSIONS AND FUTURE WORK

Adaptive streaming of omnidirectional/360-degree video content in a virtual reality (VR) setting is a challenging task which requires smart encoding and streaming techniques to cope with today's and future application and service requirements. In this paper, we explored various options enabling the bandwidth efficient adaptive streaming of omnidirectional video over HTTP. We presented a system architecture and implemented basic tools to facilitate the evaluation of different encoding and streaming options utilizing tiles within HEVC/H.265. A dataset for ODV streaming is defined which serves as a basis for the evaluation comprising different segment sizes, tiling patterns, spatial resolutions, map projection, quantization parameters, source video sequences, and even viewport-based ODV content generated from recorded head movements.

The actual evaluation is performed with respect to bitrate overhead (due to tiling), bandwidth requirements, and viewport PSNR. Based on the results we can conclude that the

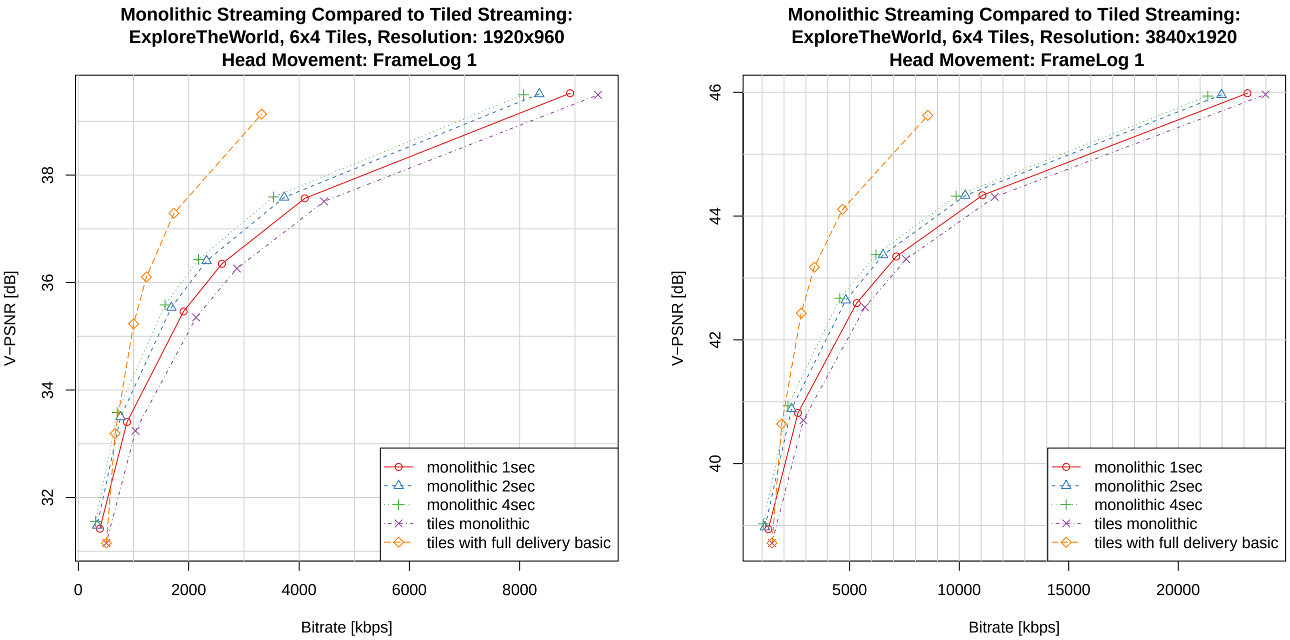

Figure 8: Viewport PSNR of a given Head Movement for User 1 (i.e., FrameLog 1) for Monolithic Streaming compared to Tiled Video Streaming for the Sequence *ExploreTheWorld*, Tiling Patterns 6×4, and different Resolutions: 1920×960 (left), 3840×1920 (right).

Table 4: BD-BR of Tiled Content over Monolithic Content with a Segment Size / Intra Period of 4 seconds using V-PSNR for the Sequence *ExploreTheWorld*.

			BD-BR [%]	
Head Movements	Resolution	Tiling	Tiles Monolithic	Tiles With Full Delivery Basic
User 1	1920x960	3x2	30.538	-9.008
User 1	1920x960	5x3	34.732	-35.427
User 1	1920x960	6x4	38.680	**-35.433**
User 1	1920x960	8x5	45.682	-35.360
User 1	3840x1920	6x4	25.874	**-38.982**
User 2	1920x960	3x2	30.779	-15.075
User 2	1920x960	5x3	34.513	-28.976
User 2	1920x960	6x4	38.501	**-40.896**
User 2	1920x960	8x5	45.748	-29.970
User 3	1920x960	3x2	31.042	-11.317
User 3	1920x960	5x3	34.926	-31.786
User 3	1920x960	6x4	38.884	**-38.389**
User 3	1920x960	8x5	46.439	-32.282

tilling pattern 6×4 provide the best tradeoff between viewport selection flexibility, bitrate overhead, and bandwidth requirements. We have formulated a variety of streaming strategies and provided a baseline evaluation to demonstrate the feasibility of tiled streaming achieving bitrate savings up to approximately 65% when applied in a realistic scenario and compared with state-of-the-art techniques. Finally, we conducted an evaluation adopting viewport PSNR based on recorded head movements achieving bitrate savings up to 40%.

Finally, we highlight potential future work items defining the outline of a roadmap for the near future in this domain and comprises various aspects including — but not limited to — the *i)* provisioning of a publicly available dataset enabling reproducible research in this area; *ii)* investigation of advanced adaptation/streaming strategies; *iii)* incorporation of machine learning techniques to predict head movements (possibly including eye tracking in combination with HMDs); *iv)* usage of different projection formats (e.g., cube map) reducing the limitations of the equirectangular project format; *v)* utilization of HTTP/2 push mechanisms as requesting each tile individually drastically increases the number of HTTP requests which may impact the overall streaming performance discussed in [24]; and *vi)* subjective quality assessments as a prerequisite for QoE models including various tilling patterns taking into account those suggested in [26].

REFERENCES

[1] Gisle Bjøntegaard. 2001. Calculation of average PSNR differences between RD-curves, ITU-T VCEG-M33. (April 2001).

[2] Byeongdoo Choi, Ye-Kui Wang, Miska M. Hannuksela, and Youngkwon Lim. 2017. *ISO/IEC CD 23000-20 Part 20: Omnidirectional Media Application Format (OMAF).* Committee Draft. W3C. Work in Progress.

[3] F. Dai, Y. Shen, Y. Zhang, and S. Lin. 2007. The Most Efficient Tile Size in Tile-Based Cylinder Panoramic Video Coding and its Selection Under Restriction of Bandwidth. In *2007 IEEE International Conference on Multimedia and Expo.* 1355–1358. DOI:http://dx.doi.org/10.1109/ICME.2007.4284910

[4] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens. 2016. JPEG Pleno: Toward an Efficient Representation of Visual Reality. *IEEE MultiMedia* 23, 4 (Oct 2016), 14–20. DOI:http://dx.doi.org/10.1109/MMUL.2016.64

[5] C. W. Fu, L. Wan, T. T. Wong, and C. S. Leung. 2009. The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video. *IEEE Transactions on Multimedia* 11, 4 (June 2009), 634–644. DOI:http://dx.doi.org/10.1109/TMM.2009.2017626

[6] S Heymann, A Smolic, K Mueller, Y Guo, J Rurainsky, P Eisert, and T Wiegand. 2005. Representation, Coding and Interactive Rendering of High-Resolution Panoramic Images and Video using MPEG-4. In *Proc. Panoramic Photogrammetry Workshop (PPW).*

[7] Evgeny Kuzyakov and David Pio. 2016. Next-Generation Video Encoding Techniques for 360 Video and VR. (2016). Online: *https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/.*

[8] Peter Lambert and Rik Van de Walle. 2009. Real-time interactive regions of interest in H.264/AVC. *Journal of Real-Time Image Processing* 4, 1 (2009), 67–77. DOI:http://dx.doi.org/10.1007/s11554-008-0102-0

[9] Jean Le Feuvre and Cyril Concolato. 2016. Tiled-based Adaptive Streaming Using MPEG-DASH. In *Proceedings of the 7th International Conference on Multimedia Systems (MMSys '16).* ACM, New York, NY, USA, Article 41, 3 pages. DOI:http://dx.doi.org/10.1145/2910017.2910641

[10] Stefan Lederer, Christopher Müller, and Christian Timmerer. 2012. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference (MMSys '12).* ACM, New York, NY, USA, 89–94. DOI:http://dx.doi.org/10.1145/2155555.2155570

[11] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou. 2013. An Overview of Tiles in HEVC. *IEEE Journal of Selected Topics in Signal Processing* 7, 6 (Dec 2013), 969–977. DOI:http://dx.doi.org/10.1109/JSTSP.2013.2271451

[12] R. Mohan, J. R. Smith, and Chung-Sheng Li. 1999. Adapting Multimedia Internet Content for Universal Access. *IEEE Transactions on Multimedia* 1, 1 (Mar 1999), 104–114. DOI:http://dx.doi.org/10.1109/6046.748175

[13] Omar A. Niamut, Emmanuel Thomas, Lucia D'Acunto, Cyril Concolato, Franck Denoual, and Seong Yong Lim. 2016. MPEG DASH SRD: Spatial Relationship Description. In *Proceedings of the 7th International Conference on Multimedia Systems (MMSys '16).* ACM, New York, NY, USA, Article 5, 8 pages. DOI:http://dx.doi.org/10.1145/2910017.2910606

[14] Ngo Quang Minh Khiem, Guntur Ravindra, Axel Carlier, and Wei Tsang Ooi. 2010. Supporting Zoomable Video Streams with Dynamic Region-of-interest Cropping. In *Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems (MMSys '10).* ACM, New York, NY, USA, 259–270. DOI:http://dx.doi.org/10.1145/1730836.1730868

[15] Patrice Rondao Alface, Jean-François Macq, and Nico Verzijp. 2012. Interactive Omnidirectional Video Delivery: A Bandwidth-Effective Approach. *Bell Labs Technical Journal* 16, 4 (2012), 135–147. DOI:http://dx.doi.org/10.1002/bltj.20538

[16] Y. Sánchez de la Fuente, R. Skupin, and T. Schierl. 2016. Video Processing for Panoramic Streaming using HEVC and its Scalable Extensions. *Multimedia Tools and Applications* (2016), 1–29. DOI:http://dx.doi.org/10.1007/s11042-016-4097-4

[17] Sandvine. 2016. 2016 Global Internet Phenomena Report: Latin America & North America. (2016). Online: *http://sandvine.com/.*

[18] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl. 2016. Tile Based HEVC Video for Head Mounted Displays. In *2016 IEEE International Symposium on Multimedia (ISM).* 399–400. DOI:http://dx.doi.org/10.1109/ISM.2016.0089

[19] I. Sodagar. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia* 18, 4 (2011), 62–67. DOI:http://dx.doi.org/10.1109/MMUL.2011.71

[20] Thomas Stockhammer. 2011. Dynamic Adaptive Streaming over HTTP: Standards and Design Principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems (MMSys '11).* ACM, New York, USA, 133–144. Online: *http://doi.acm.org/10.1145/1943552.1943572.*

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

[22] Marko Viitanen, Ari Koivula, Ari Lemmetti, Arttu Ylä-Outinen, Jarno Vanne, and Timo D. Hämäläinen. 2016. Kvazaar: Open-Source HEVC/H.265 Encoder. In *Proceedings of the 2016 ACM on Multimedia Conference (MM '16).* ACM, New York, NY, USA, 1179–1182. DOI:http://dx.doi.org/10.1145/2964284.2973796

[23] Vladimir Vukicevic, Brandon Jones, Kearwood Gilbert, and Chris Van Wiemeersch. 2017. *WebVR.* Editors Draft. ISO/IEC JTC 1/SC 29/WG 11. Work in Progress.

[24] Mengbai Xiao, Viswanathan Swaminathan, Sheng Wei, and Songqing Chen. 2016. Evaluating and Improving Push Based Video Streaming with HTTP/2. In *Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '16).* ACM, New York, NY, USA, Article 3, 6 pages. DOI:http://dx.doi.org/10.1145/2910642.2910652

[25] M. Yu, H. Lakshman, and B. Girod. 2015. A Framework to Evaluate Omnidirectional Video Coding Schemes. In *2015 IEEE International Symposium on Mixed and Augmented Reality (ISMAR).* 31–36. DOI:http://dx.doi.org/10.1109/ISMAR.2015.12

[26] Alireza Zare, Alireza Aminlou, Miska M. Hannuksela, and Moncef Gabbouj. 2016. HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications. In *Proceedings of the 2016 ACM on Multimedia Conference (MM '16).* ACM, New York, NY, USA, 601–605. DOI:http://dx.doi.org/10.1145/2964284.2967292