Delay Impact on MPEG OMAF's Tile-Based Viewport-Dependent 360° Video Streaming

Yago Sánchez de la Fuente, Gurdeep Singh Bhullar, Robert Skupin, Cornelius Hellge, and Thomas Schierl, *Senior Member, IEEE*

***Abstract*— With the continuously increasing attention that 360° video streaming is drawing, several solutions have been developed lately. They can be grouped into two main categories, with the first one being the *viewport-independent* category that consists of encoding the whole 360° video content using a particular projection, e.g., Equirectangular Projection or Cubemap Projection, without taking any viewport orientation into account. Such approaches waste resources since content not being watched by the user is encoded with the same visual fidelity as the content actually watched. The second category, the *viewport-dependent* category, relies on techniques that allow for viewport adaptivity. One approach is to apply viewport-dependent projections, wherein the spherical 360° video is mapped onto a rectangular frame in such a way that a specific viewport is mapped to a comparatively larger part of the rectangular frame than the rest of the content. Another approach, as specified in MPEG OMAF, is to use tile-based streaming, with the 360° video being offered in a tiled manner at various resolutions. Thus, each user can retrieve the tiles at different resolutions so that the high-resolution tiles match its viewport. Although viewport-dependent approaches allow providing better visual quality at the viewport, the end-to-end delay is critical, since it has an impact on the time needed by clients to adapt to user movements so that movements are reflected on the retrieved content. In this paper, we analyze the impact of the delay and introduce various means to reduce the impact on the observed fidelity.**

***Index Terms*— Tile, OMAF, prediction, viewport, HEVC.**

I. Introduction

IN THE last years, there has been a rise in the interest of the research community and industry in 360° video streaming. 360° video streaming, a.k.a. Virtual Reality (VR) streaming, is understood as omnidirectional video content consumed with Head Mounted displays (HMDs). For such applications, the head pose may change considerably within milliseconds and it is vital that the viewport, i.e. the portion of the content visible to the user, is adapted instantaneously to the momentary head pose.

The increasing interest becomes obvious with the continuously rising number of HMDs available in the market, such as of Oculus Rift [1], Google Cardboard [2], Daydream [3], HTC VIVE [4], PlayStation VR [5] or Samsung GearVR [6]. With the consumer grade VR headsets improving in quality and with the increasing number of manufacturers, a lot of attention is drawn to creating content for such devices, among which video can become an important service. In fact, [7] forecasts that virtual reality and augmented reality video traffic will increase at a compound annual growth rate of 82% between 2016 and 2021. The main obstacle of 360° content generation used to be means for capturing content, which has been overcome as numerous manufacturers have started producing 360° cameras to enable capturing 360° video content. Manufacturers include GoPro Omni [8], Google Odyssey [9], Samsung Project Beyond [10], Facebook Surround 360 [11]. At the same time, distribution platforms, such as, Facebook [12] and YouTube [13] are already supporting 360° video streaming for VR devices.

In order to avoid the fragmentation of these new market segments and to ensure interoperability of 360° video ecosystems, many relevant standard development organizations (SDOs) have been working in the context of 360° video. For example, 3GPP has specified operation points and media profiles [14] for VR streaming in 3GPP ecosystems. At the same time, the Video Coding Experts Group (VCEG, ITU-T Q6/16) and the Moving Picture Experts Group (MPEG, ISO/IEC JTC 1/SC 29/WG 11) have taken the lead, among all SDOs working on 360° video related standardization, in the areas of 360° video coding and delivery. The Joint Collaborative Team on Video Coding (JCT-VC), has specified an amendment of HEVC with additional Supplemental Enhanced Information (SEI) messages for 360° video [15] while the Joint Video Experts Team (JVET), formerly known as the Joint Video Exploration Team, has investigated various 360° video projection formats.

In parallel, MPEG Systems subgroups have been working on delivery and other system aspects with the effort known as Omnidirectional MediA Format (OMAF) [16], which has become Part 2 of the emerging ISO/IEC 23090 MPEG-I standards suite on Immersive Media.

In addition, an industry forum called Virtual Reality Industry Forum (VR-IF) has been set up to achieve interoperability and bring together a broad range of companies from different sectors including the movie, television, broadcast, mobile, and interactive gaming ecosystems. Relaying on the technology developed in JCT-VC, JVET and MPEG, VR-IF has developed guidelines on content capturing, generation and delivery [17].

Manuscript received August 13, 2018; revised November 30, 2018; accepted January 29, 2019. Date of publication February 14, 2019; date of current version March 11, 2019. This paper was recommended by Guest Editor T. Stockhammer. *(Corresponding author: Yago Sánchez de la Fuente.)*

The authors are with the Multimedia Communications Group, Video Coding and Analytics Department, Fraunhofer Heinrich-Hertz Institute, 10587 Berlin, Germany (e-mail: yago.sanchez@hhi.fraunhofer.de; gurdeep.sigh.bhuller@hhi.fraunhofer.de; robert.skupin@hhi.fraunhofer.de; cornelius.hellge@hhi.fraunhofer.de; thomas.schierl@hhi.fraunhofer.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2019.2899516

Current guidelines on content generation and delivery focus on providing a suitable quality for currently available HMDs. On the one side, available HMD still provide a relatively low screen resolution of around 1440x1280 per eye. On the other side, decoding capabilities in typical devices are constrained by HEVC decoder level limits that were designed with traditional broadcast use cases in mind. While in such use cases the complete decoded picture is presented to the user, in 360° video use cases, a resolution of 4K for the whole video leads only to an achievable viewport resolution of roughly 1Kx1K assuming a Field of View (FoV) of 90°x90° and ERP, which is considerably less than what current HMDs are capable of, i.e. $< 57\%$ of the pixels compared to 1440x1280. Therefore, the guidelines provide, in addition to a *viewport-independent* configuration, a *viewport-dependent* configuration that fully exploits the currently available HMD capabilities in terms of resolution.

However, it is expected that in order to provide a good user experience, a resolution of around 4K is required for covering the viewport of the user. Note that visual acuity of the Human Visual System is assessed to be around 60 pixel per degree in the fovea [18] which would lead to a resolution even slightly higher than 4Kx4K for a viewport with a Field of View (FoV) of $90° \times 90°$. Using a naive approach of transmitting the complete 360° video with such fidelity would require devices to decode up to 21Kx10K assuming ERP. Such vast decoder capability requirements do not seem feasible in a near future given that most deployed video hardware is and will be designed for the broad monopoly of traditional video formats. Even though it is expected that decoding capabilities will increase in a near future, it is still considered that decoding capabilities will not fully fulfil the requirements for 360° videos according to the above naïve approach, at least for constrained devices such as mobile platforms.

Therefore, viewport adaptive coding and transmission schemes, as explained in section II, are the only viable solution to consider nowadays in order to fully utilize device capabilities and achieve a desirable visual quality in the viewport of users. These schemes can achieve higher visual fidelity within the current user viewport by sacrificing fidelity of video areas that are not within the viewport, for instance in terms of effective resolution.

Thus, in addition to more pixel budget compared to the above naïve approach, bitrate can also be saved for the price of temporarily showing content to the user at lower quality until the content optimized for the current user viewport is retrieved by the client.

This paper considers a viewport adaptive coding and transmission scheme using HEVC tiles. However, since *viewport-dependent* solutions involve continuous change of the retrieved content (e.g., as the user changes its head orientation), it is crucial to reduce latencies toward enabling compelling interactive VR experiences. A high degree of responsiveness is required so that the retrieved content is optimized for the viewport of the user, as a high end-to-end delay would lead to users retrieving content not optimal for their viewport and of poor fidelity and thus, would lead to a poor user experience. The paper shows the impact of the delay on unequal resolution tile-based streaming and provides some strategies to improve the quality of such a scheme.

The remainder of this paper is organized as follows. Section II describes the state-of-the-art of 360° video streaming. In section III, the configuration of the analyzed tile-based streaming is described and the impact of the end-to-end delay on its performance is shown. Section IV shows the effect of retrieving tiles not belonging to the viewport not only at lower resolution but also with a higher QP than that of the tiles in the viewport. Section V describes two basic prediction models, presents the results and explores the impact of using the prediction models when using equal and different QPs for the lower and higher resolution tiles. In section VI an algorithm is introduced that combines viewport prediction with a velocity-based QP distribution. Section VII present the results of a particular encoding configuration that allows to use a single video decoder for decoding multiple tiles when using the algorithm proposed in section V. Finally, in Section VIII the conclusion is presented.

II. State-of-the-Art of 360° Video Streaming

The first step for encoding 360° video content is to use a sphere-to-plane projection mapping to represent the omni-directional content in a limited rectangular picture. The most basic and widely used projection is the Equirectangular Projection (ERP) [19]. It consists of mapping longitude and latitude lines to even straight lines in the projected rectangular picture. However, ERP is a non-equal area projection, i.e. it is not an area preserving projection. Furthermore, it suffers from severe oversampling toward the sphere poles, which means that areas in the vicinity of the sphere poles are represented with a much higher number of pixels in the projected picture than equally sized areas that are located closer to the sphere equator. Further geometric distortions introduced by ERP are also detrimental to the coding efficiency of many codecs that traditionally employ a translatory motion model. Another commonly used projection is the Cubemap Projection (CMP) [20]. In CMP, the camera surroundings are projected onto the six faces of a cube, where the sample value of each sample on a cube face stems from a rectilinear projection of the camera surroundings onto the position of that sample. The resulting pictures for each cube face are then arranged in the rectangular frame of traditional video. Although, CMP is also a non-equal area projection, the over-sampling and geometric distortion issues of ERP are sharply decreased and hence, gains in coding efficiency can be demonstrated compared to ERP.

However, since ERP and CMP are viewport-agnostic projections, they suffer from the problem that they sacrifice a substantial number of pixels for video areas that are not even presented to the user as they are located outside the current user viewport.

As already mentioned, a superior solution can be provided by viewport adaptive coding and transmission schemes. Sphere-to-plane projections that achieve this purpose are herein referred to as viewport-specific projections. In the case of viewport-specific projections, a target viewport is defined and a higher amount of pixels-per-degree is assigned to the

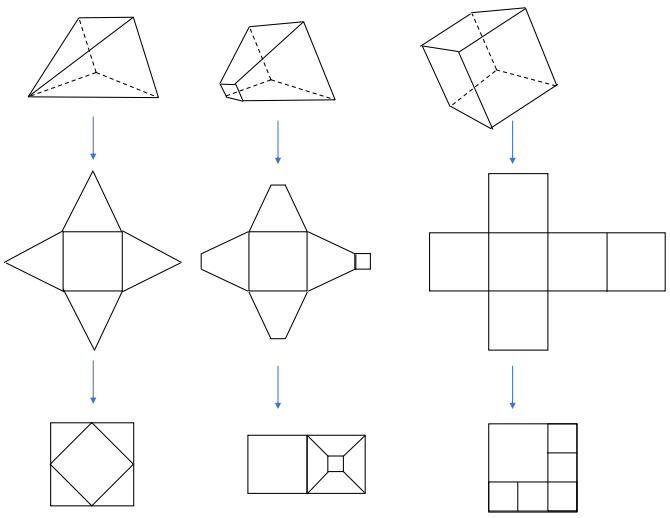

Fig. 1. Pyramid, truncated pyramid and multi-resolution CMP generation.

content closer to the target viewport than to content farther away from the target viewing orientation. Examples of such viewport-specific projections are illustrated in Fig. 1 from left to right: the pyramid projection [21], the truncated pyramid projection [22], and a multi-resolution CMP variant [23]. Fig.1 illustrates how these three viewport-specific projections are generated. While the top row illustrates the geometric primitives, the second row shows the unrolled surfaces and the bottom row gives possible arrangements of the polygon faces within a rectangular video frame. In the case of the pyramid or the truncated pyramid, the base of the polygon corresponds to the viewing direction of the user. Thus, the sampling density of the projected frame is highest in the area which is observed by the user. In the case of the multi-resolution CMP, faces of the polygon not corresponding to the viewing direction of the user are downsampled before being arranged within the rectangular frame.

One of the issues of viewport-specific projections is that, although the number of projections that need to be offered simultaneously for a service is configurable, typically a high number is required such as 30 or more in order to be able to match any given user orientation and provide a smooth quality transition when switching from one viewing direction to another. Corbillon *et al.* [24] describe such an approach on delivery using for 360° video with viewport-specific encodings where a fixed number of streams matching different viewports are offered. However, each of the viewport specific streams comes at considerable overhead cost for rendering at the content generation side, encoding and transmission (e.g. caching).

Tile-based streaming of 360° video can solve the issues inherent to streaming schemes based on viewport-specific projections. Tile-based streaming was presented in [25] for Region of Interest (RoI) panorama streaming, where only a subset of the video was transmitted. Although, only transmitting a subset of the video for 360° video streaming does not seem to be a valid approach since user poses can change very rapidly when using an HMD, tile-based streaming allows for a good trade-off to balance the pros and cons of viewport-adaptivity. The idea of tile-based streaming of 360° video is to offer tiled 360° video content at different resolutions so that the client can chose tiles at different resolutions depending on the user viewing direction, as explained for instance in [26]. Thus, in comparison to the viewport-specific projection-based streaming, less overhead for rendering and encoding compared to viewport-specific projections during content creation, as well as, less storage at the server and CDNs to store the content is required.

Tile-based streaming for panorama videos using MPEG-DASH has been already studied in [27]. Hosseini and Swaminathan [28] show the benefits of using a viewport-aware adaptation technique for tile-based streaming of 360° VR video. However, one critical aspect that needs to be taken into account for viewport-dependent schemes, and has not been studied yet, is the impact of the response time between user orientation changes and content retrieval on the visual fidelity. Obviously, a rapid response of the transmission system to changes in user orientation is required. In fact, the longer the response time of the system to user orientation change is, the longer will be the period in which the user is presented with low-quality content before the transmission and coding delay adapt to the new viewing orientation.

In order to achieve low latencies for DASH streaming, rate adaptation algorithms should work on a very small buffer size. However, Sanchez *et al.* [29] provide an analysis on how low buffers impact adaptive HTTP streaming. The results show that in order to avoid playback interruptions, caused by throughput variations in the network, DASH clients need to download content with a much lower bitrate than the measured throughput so that they can quickly react to network variations.

Therefore, it is crucial to predict future user viewing orientation to allow building comparatively larger buffers while showing high quality content most of the time. Several studies have been done lately on predicting the viewport of a user for 360° video streaming. In [30] and [31] prediction of a future fixation point has been studied with two methods discussed: constant velocity and constant acceleration based prediction. Based on these works, [32] applies prediction for a tile-based streaming scenario to transmit only a subset of tiles showing that by modeling the prediction error by a normal distribution with mean equal to 0 and taking the variance of the prediction error into account, it is possible to download the required tiles with a sufficiently high confidence level. More advanced prediction models have been proposed in the literature based on Linear Regression models (LRM) or Weighted Linear Regression Models (WLRM) as in [33]. Xie *et al.* [34] use an LRM model and based on the probabilities of the tiles as well as their bitrate and distortion an optimization is performed on which tiles to download at which of the given bitrate. The main issue with such an approach is that the rate distortion characteristics of each of the tiles for a 360° video needs to be known a priori. Further papers have presented different approaches to improve prediction, such as using saliency maps [35] or neural networks [36].

In general, to our knowledge one missing aspect in all considerations is an analysis of the effect of the delay on tile-based streaming, demonstrating the range of delays for which tile-based 360° video streaming may offer superior fidelity

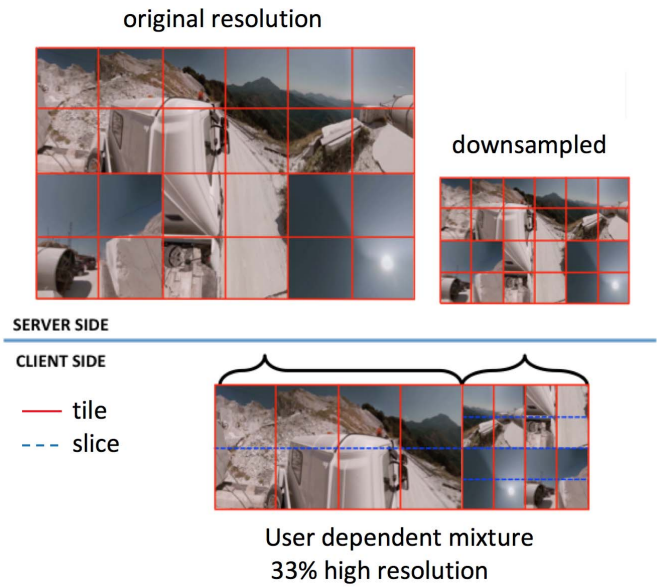

Fig. 2. Tiling of cubic video at different resolutions.

compared to the naïve state-of-the art viewport-independent approach and to what extent prediction can help improve a tile-based streaming system further.

III. Latency Effect on Tile-Based Streaming

A tile based streaming system based on DASH and HEVC was presented in [37]. The system uses CMP to represent the 360° video. The video is then sampled to various resolutions and tiled at a desired granularity (see the example in Fig. 2, where two resolutions are offered with a downsampling factor of 2x). The shown configuration is similar in spirit to configurations described in the Annex D of MPEG OMAF [38] as illustrated below.

On server side, the 360° video at several resolutions is tiled as e.g. illustrated at the top of Fig. 2 and each tile bitstream is offered separately, i.e. as separate bitstreams. Then, the client can select which tiles to download at a high resolution and which tiles at low resolution, as illustrated at the bottom of Fig. 2 with one such possible configuration.

In order to allow for a rapid response of the streaming system to user orientation changes, it is necessary that the transmitted content is offered with frequent Random Access Points (RAPs). However, very frequent RAPs are detrimental for coding efficiency of a video codec as these pictures have to be encoded using intra-prediction only and reset the inter-prediction chain. In the following, the concept of shifted-IDR as presented by Sanchez *et al.* [39] is used. The SIDR concept allows to mitigate the penalty of frequent RAPs by increasing the RAP frequency available to a client without increasing the RAP frequency of individual encoded streams. This is achieved by offering on server-side multiple streams or representations of a single content item such as a tile wherein the position of RAPs is different among the streams. A client can download and decode a content item at any of the different available RAP positions, but may at the same time benefit from an increased coding efficiency that stems from the larger RAP period in each individual representation when no RAP is required, as is for instance the case when a content item such as a tile does not undergo a resolution change. In the given system, the SIDR approach is used with a RAP period of 32 frames and 4 SIDR representations so that effectively, a RAP functionality at every 8 frames can be offered to a client.

10 sequences in CMP format with a resolution of 3072x2048 pixels have been used for the evaluation in this paper. The number of frames per sequence is varying from 300 to 900 frames with a framerate between 25 fps and 30 fps. The sequences have been encoded into 24 tiles of 512x512 pixels for high resolution and 256x256 for low resolution. The configurations available to the client consist of 8 high resolution tiles and 16 tiles low resolution tiles, effectively allowing a viewport to cover 1/3 of the 360° video in high quality.

In summary, the videos have been split into 24 tiles and each of the has been encoded at two resolutions with a downsampling factor 2:1. Each of the tiles at each resolution has been encoded at 4 QP values: namely 22, 27, 32 and 37. Besides, for each of these configurations, 4 bitstreams have been encoded with a RAP period of 32 frames each but with a shift of RAP of 8 frames corresponding to the previous SIDR version.

As aforementioned the performance of viewport-dependent approaches, i.e. the achievable user experience, decreases with the latency between user orientation change and downloading and presenting a tile setup that corresponds to the new actual viewing direction after an orientation change. For the duration of this latency, the user will be presented low-quality content affecting the user experience. Although no subjective evaluation has been performed assessing the effect of showing low-resolution content or mixed-resolution content for a short period of time, implementations of such solutions have been showcased, as e.g. [37], and it seems to be acceptable for the viewer to show such a lower fidelity content for a couple of hundreds of milliseconds.

In the following, the tile-based viewport-dependent approach is compared to the viewport-independent approach. The latter consists of sending the full sphere using CMP without adaptivity to the user viewport. This is carried out with a resolution of 2176×1448 pixel, which roughly corresponds to the number of pixels of the above described viewport-dependent approach with a resolution of 3072×1024.

In order to compare both approaches actual user traces for each of the test sequences have been recorded from 17 test subjects using an OculusRift CV1 HMD. Each of the test subjects was let to freely navigate through each of the 10 sequences and traces of their viewports have been recorded. Using the user traces, viewports have been generated and BD-rates [40] have been compared. For this purpose, the 17 traces collected for each of the sequences have been used to generate the corresponding viewport of $90° \times 90°$ for both the viewport-dependent and viewport-independent approach. Thus, the PSNR of these viewports has been calculated and with it the corresponding BD-rate has been computed.

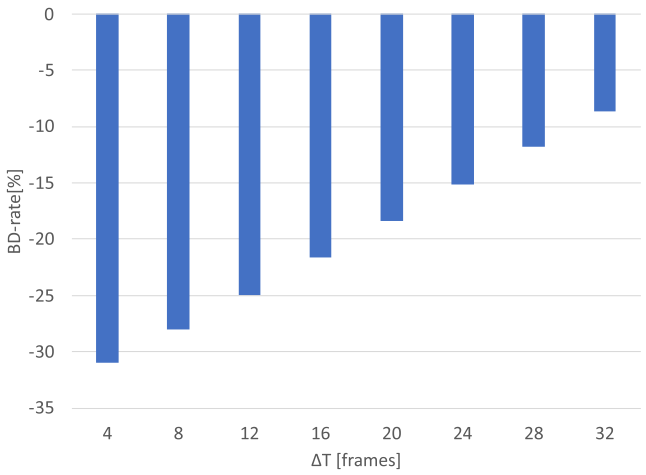

Fig. 3. BD-rate gains compared with the viewport-independent approach.

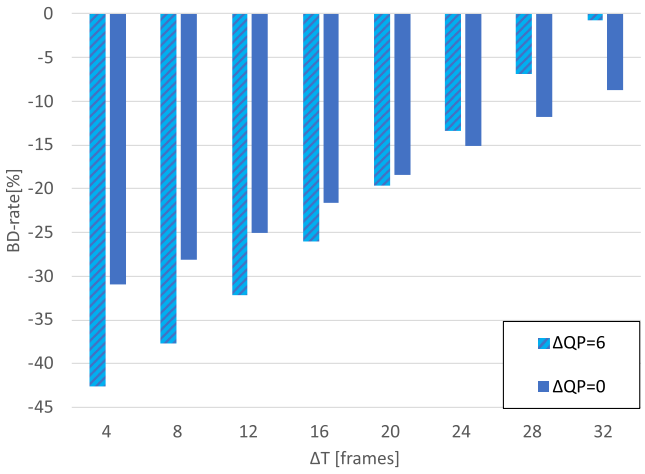

Fig. 4. BD-rate gains for a ΔQP of 6 compared with the viewport-independent approach.

Fig. 3 shows BD-rate performance of the viewport-dependent approach versus the viewport-independent approach over different latency values counted in frames. This means that the decision of which tiles to download at high-resolution and which to download at low-resolution for each segment is taken ΔT[frames] before starting to play back the corresponding segment and the viewing direction at that time is used for selecting the tiles at their respective resolutions. Values of ΔT[frames] between 4 and 32 have been evaluated.

As can be seen in Fig. 3, the viewport-dependent approach outperforms the viewport-independent approach. However, the larger the latency is, the worse becomes the performance. The latency, which is taken into account in this paper, referred to as end-to-end delay is defined as the time between the moment when the request is carried out for the tiles and the time at which they are played back. For very low end-to-end delays of 4 frames (i.e., around 160 ms), BD-rate gains of around 31% compared to the viewport-independent approach can be achieved. However, as the delay increases gains sharply decrease. It can be seen that gains drop up to 8% for a delay 32 frames.

IV. Unequal QP Assignment

While resolution adaptivity is a crucial step in efficiently using the available resources constrained by decoder level limits, the requirement to form a regular tile grid restricts flexibility and granularity of the resource management, e.g. with respect to the downsampling factor. Obviously, video codecs offer further means to adjust resource distribution, e.g. by means of adapting the quantization parameters (QP) of tiles. It can be shown that additional gains can be obtained when the fidelity of the tiles not corresponding to the viewport at the time that the request is performed is further reduced with respect to visual quality by means of coarser quantization. Hence, in addition to providing tiles at a lower resolution, tiles encoded with a higher QP, with $\Delta QP = QP_{LowRes}\text{-}QP_{HighRes}$, can be chosen for those tiles not in the viewport. In Fig. 4 results corresponding to using a ΔQP of 6 for the downloaded low-resolution tiles that do not correspond to the current user viewport are shown in comparison to using a ΔQP of 0, which are shown in dashed blue and solid blue respectively.

For this purpose, the low resolution bitstream described above have been instead encoded with QP values of 28, 33, 38 and 43.

For comparatively very short end-to-end delays, e.g. for a ΔT equal to 4 frames, gains of around 12% can be achieved in comparison to the previously presented results not differentiating low and high-resolution tiles in the quantization. With a ΔQP of 6, a maximum gain of around 43% BD-rate compared to the state-of-the-art viewport-independent approach can be achieved. However, as the end-to-end delay increases and more content of the lower fidelity tiles is shown to the user, the benefit of the ΔQP of 6 turns into a drawback as rate savings turn into a lower observed fidelity. Hence, under such conditions, downloading tiles at low resolution with a higher QP provides a worse solution.

One option would be to have a client deciding on whether to download the tiles with a different QP based on the observed end-to-end system latency. For instance, downloading tiles not belonging to the viewport with a ΔQP of 6 if the delay is equivalent to a value below 20 frames and with ΔQP equal to 0 if the delay is equivalent to a value larger than 20 frames.

V. Viewport Prediction

In order to tackle the issue of delay and associated loss in visual fidelity in the viewport of a user when using viewport-dependent streaming, this paper proposes a strategy based on two variants of head orientation prediction: namely angular velocity-based and angular acceleration-based prediction of the head pose. The prediction strategies are the same as presented in [30] but extended as follows to accommodate to a segment-based streaming.

More concretely, the future viewing orientation of a user needs to be predicted for a future segment, which consist of 8 frames in the given system design. Instead of predicting the viewing direction of a user with respect to the first frame of a segment, prediction is carried out for the frame in the middle of the segment and that viewing orientation is assumed to be constant for the whole segment so that

the high-resolution and low-resolution tiles corresponding to that viewport are downloaded. Thus, the client downloads content which will be most suitable for the whole duration of the respective segment. More precisely, this means that the prediction is carried out for a frame in the future that is $\Delta T + 4$ frames from the moment the prediction is carried out.

The goal of the prediction is to compute the extrinsic quaternion Qt_{pred} that corresponds to the head orientation of the user at the specified future time instant. The head orientation change is represented by a quaternion ΔQt defining a rotation by an angle θ about an axis vector v through the origin as follows:

$$\Delta Qt\,(v, \theta) = \left(\cos\left(\frac{\theta}{2}\right), v_x \sin\left(\frac{\theta}{2}\right), v_y \sin\left(\frac{\theta}{2}\right), v_z \sin\left(\frac{\theta}{2}\right)\right) \quad (1)$$

in which $\Delta Qt(v,\theta)$ denotes a unit-length quaternion that corresponds to a rotation of θ radians about a unit length axis vector $v = (v_x, v_y, v_z)$.

Given the quaternion representation of the predicted head orientation change, i.e. $\Delta Qt_{pred}(v,\theta)$, the predicted extrinsic quaternion Qt_{pred} can be computed by a quaternion multiplication of the $\Delta Qt_{pred}(v, \theta)$ and the extrinsic quaternion representation of the head pose at the moment of carrying out the prediction Qt(t).

$$Qt_{pred}\,(t + \Delta T + 4) = \Delta Qt_{pred}\,(v, \theta) * Qt(t) \quad (2)$$

A. Angular Velocity-Based Head Orientation Prediction

The first prediction approach for the head orientation change $\Delta Qt_{pred}\ (v, \theta)$ using the current head pose context is based on the momentary angular velocity $\omega(t)$ of the head at a given time instant.

As the angular velocity of the head orientation change is assumed to be constant during the timeframe between current time instant t and future time instant $(t + \Delta T + 4)$, with $\Delta T + 4$ being the timeframe to be predicted, both the rotation axis vector and the rotation angle can be written as a function of the angular velocity. More concretely, v(t) corresponds to the unitary vector with same direction as the angular velocity vector:

$$\omega(t) = (\omega_x, \omega_y, \omega_z) \quad (3)$$

$$v(t) = \frac{\omega\,(t)}{\|\omega\,(t)\|} \quad (4)$$

The rotation angle $\theta(t, \Delta T + 4)$ from t to $t + \Delta T + 4$ is linearly proportional to the magnitude of the angular velocity and can be computed as follows:

$$\theta\,(t, \Delta T + 4) = \|\omega\,(t)\| * (\Delta T + 4) \quad (5)$$

Therefore, the predicted extrinsic quaternion corresponding to the head pose can be computed by substituting in (2) the values of v(t) and $\theta(t,\Delta T + 4)$ from (4) and (5) respectively.

B. Angular Acceleration-Based Head Orientation Prediction

The second approach for prediction of the head orientation is based on angular acceleration of the head orientation change. Taking into account that the angular velocity $\omega(t)$ changes during the timeframe of the prediction, the angular acceleration is defined as follows, with Δt being the time difference for two frames, i.e. the frame at which the prediction is carried out and the previous frame:

$$a(t) = \frac{\Delta\omega\,(t)}{\Delta t} \quad (6)$$

Based on the angular acceleration, a constant average angular velocity $\omega_{avg}(t)$ can be computed that is equivalent to the non-constant velocity case, in the sense that it results in the same orientation change as when angular acceleration is considered.

$$\omega_{avg}\,(t) = \omega\,(t) + a(t)\frac{(\Delta T + 4)}{2} \quad (7)$$

When considering angular acceleration of the head orientation change, the computed $\omega_{avg}(t)$ might lead to sharply fluctuating values that magnify noisy angular velocity and angular acceleration measurements. Therefore, a Savitzky-Golay filter is used to smoothen the computed velocities.

Thus, the new $\omega_{avg}(t)$ can be used the computation of v(t) and $\theta(t,\Delta T + 4)$:

$$v(t) = \frac{\omega_{avg}\,(t)}{\left\|\omega_{avg}\,(t)\right\|} \quad (8)$$

$$\theta\,(t, \Delta T + 4) = \left\|\omega_{avg}\,(t)\right\| * (\Delta T + 4) \quad (9)$$

Then, the predicted extrinsic quaternion corresponding to the head pose can be computed by substituting in (2) the values of v(t) and $\theta(t, \Delta T + 4)$ derived using the acceleration-based approach, i.e. the values from (8) and (9) respectively.

However, as shown in Table I errors of the acceleration-based prediction are very high, which are similar to the velocity-based prediction. This indicates that the models based on [30] do not work properly, especially for high values of delay, without any modification.

TABLE I

MEAN ABSOLUTE ERROR IN DEGREES FOR THE ANGULAR ACCELERATION-BASED PREDICTION

ΔT	Yaw	Pitch	Roll
4	5.03	2.41	2.49
8	12.65	5.55	6.13
12	22.99	9.35	11.21
16	34.04	13.24	17.10
20	44.21	16.70	23.04
24	52.67	19.51	28.46
28	59.39	21.64	33.11
32	64.83	23.25	36.89

C. Alpha-Correction

The above equations for prediction of the future head orientation introduce a problem due to the fact that they do not incorporate valuable context such as the anatomical limits of head pose changes. As head orientation changes can be very rapid when regarded at such an instantaneous time scale, resulting predictions suffer from inaccuracies. As shown in Table I inaccuracies in the prediction increase

TABLE II
OPTIMAL ALPHA-CORRECTION (α_x) VALUES

	α	
ΔT	Velocity-based	Acceleration-based
4	0.85	0.75
8	0.7	0.60
12	0.6	0.5
16	0.55	0.45
20	0.50	0.4
24	0.45	0.35
28	0.40	0.3
32	0.40	0.25

TABLE III
MEAN ABSOLUTE ERROR IN DEGREES FOR THE ANGULAR ACCELERATION-BASED PREDICTION WITH ALPHA-CORRECTION

ΔT	Yaw	Pitch	Roll
4	3.88	1.67	1.69
8	7.93	3.09	3.15
12	12.24	4.48	4.61
16	16.53	5.83	6.01
20	20.71	7.01	7.24
24	24.70	7.94	8.22
28	28.23	8.90	9.22
32	32.23	9.65	10.00

with the delay, i.e. the further in the future the predicted head orientation, i.e. the larger ΔT, the higher is inaccuracy of the prediction.

Hence, a degree of error in the prediction was encountered, which needs to be accounted for. Taking inspiration from drift error correction method used in tracking methods for Oculus rift [30], a correction factor is added to the prediction.

Although the idea in [30] is to cope with a drift inherent from the measurements of the sensor, something similar can be applied to the prediction models presented in IV. The correction factor, used in the following, is based on the assumption that the larger the timeframe predicted, the less accurate is the hypothesis that the orientation change can be described by a constant velocity or constant acceleration that applies the whole predicted time. The assumption done for the alpha-correction presented below, is that the velocity and acceleration considered for prediction is not constant but will tend to a value of 0 as ΔT becomes bigger. This results in a computation of a corrected factor θ'(t,ΔT + 4) as:

$$\theta'(\mathrm{t},\,\Delta\mathrm{T}+4) = \left\|\vec{\alpha}(\Delta\mathrm{T}) * \omega(\mathrm{t})\right\| * (\Delta T + 4) \qquad (10)$$

In addition, for the angular acceleration-based head orientation prediction, the same correction factor is applied for the computation of the equivalent constant average angular velocity ω'$_{\mathrm{avg}}$(t) as follows:

$$\omega'_{\mathrm{avg}}(\mathrm{t}) = \omega(\mathrm{t}) + \vec{\alpha}(\Delta\mathrm{T}) * \mathrm{a}(\mathrm{t})\frac{(\Delta\mathrm{T}+4)}{2} \qquad (11)$$

The correction factor $\vec{\alpha}(\Delta\mathrm{T}) = (\alpha_{\mathrm{x}},\, \alpha_{\mathrm{y}},\, \alpha_{\mathrm{z}})$ is computed for the different values of ΔT so that the mean square error is minimized as follows:

$$\vec{\alpha}(\Delta T) = \underset{\vec{\alpha}}{min} \sum \left(\mathrm{Qt}_{pred} - \mathrm{Qt}_{real}\right)^2 \qquad (12)$$

The empirically derived optimal values for α_{x} for the 17 user traces for each of the sequences are listed in Table II exemplarily. As can be seen, the values of $\vec{\alpha}(\Delta\mathrm{T})$ decrease with increasing ΔT and the values for the angular acceleration-based approach are lower than those of the angular velocity-based approach.

The results shown in Table III show that the alpha-correction factor described above significantly improves the prediction as compared with not using the correction factor (see results in Table I). Fig. 5. shows the results of using the velocity-based prediction in orange and not using any prediction in blue for comparison. The results correspond to the case where ΔQP is equal to 0. It can be seen that in general gains from 1% to around 4% can be achieved when using the described velocity-based prediction compared to not using any prediction. Particularly, big gains (around 4%) can be shown for higher values of ΔT, since the amount of content shown in low-resolution is reduced when viewport prediction is applied. Similar results are shown in Fig. 6 for the case where ΔQP is set to 6. The velocity-based prediction, represented with the dashed orange bars, is able to improve the performance in particular for high values of ΔT. It can be seen that gains up to almost 10% can be achieved for ΔT = 32 over the case with no prediction, which is represented with dashed blue bars. Still for ΔT = 32 the results using ΔQP = 0 (see Fig. 5) show to be better, than when ΔQP = 6, with around 12% and 10% gains over the viewport-independent approach respectively.

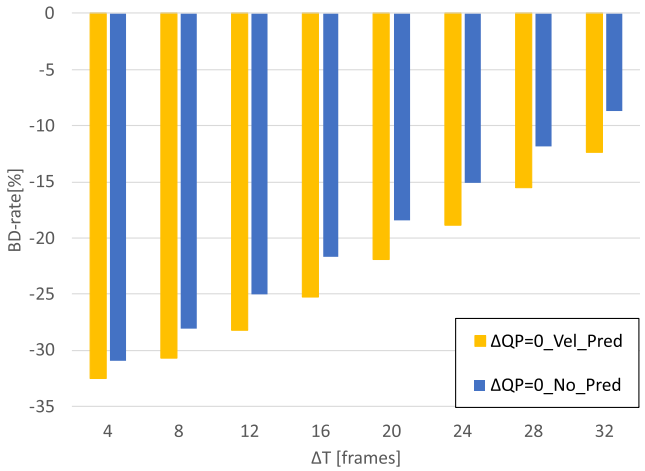

Fig. 5. BD-rate gains for a ΔQP of 0 compared with the viewport-independent approach for the velocity-based prediction.

Fig. 7. shows the results of using the acceleration-based prediction (in green) compared to the velocity-based prediction approach (in orange) when ΔQP is equal to 0. It can be seen that slightly higher gains of 1% can be obtained for all values of ΔT. The corresponding results for a ΔQP of 6 are shown in Fig. 8. Similar gains can be observed for the

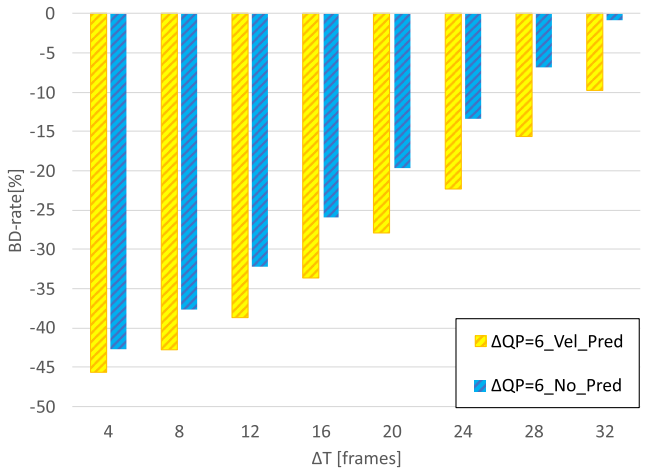

Fig. 6. BD-rate gains for a ΔQP of 6 compared with the viewport-independent approach for the velocity-based prediction.

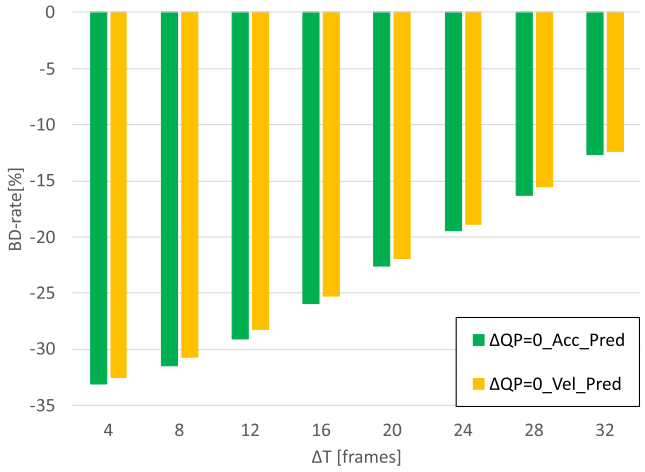

Fig. 7. BD-rate gains for a ΔQP of 0 compared with the viewport-independent approach for the acceleration-based prediction.

acceleration-based prediction in comparison to the velocity-based prediction for $\Delta QP = 6$.

VI. Viewport Prediction With Velocity Based QP Distribution

As already discussed before, there is a dependency on the system end-to-end delay (ΔT) or prediction time, that influences the effect of applying a ΔQP to low-resolution tiles. The smaller the values of ΔT, the more beneficial is to have a ΔQP greater than 0 but for higher values of ΔT, ΔQP values of 0 provide a more efficient tile-based streaming. This behaviour is logical, since the larger ΔT is, the larger are the errors that result from predicting the viewport and therefore, the more low-resolution/quality tiles are shown in the viewport. Therefore, for cases where low-resolution is shown often, a ΔQP of 0 results in a better quality shown to the user compared to downloading the low-resolution tiles encoded with a ΔQP of 6.

In a similar manner, an analysis of the user traces showed that the larger the values of the angular velocity and acceleration, the larger was the prediction error and, therefore,

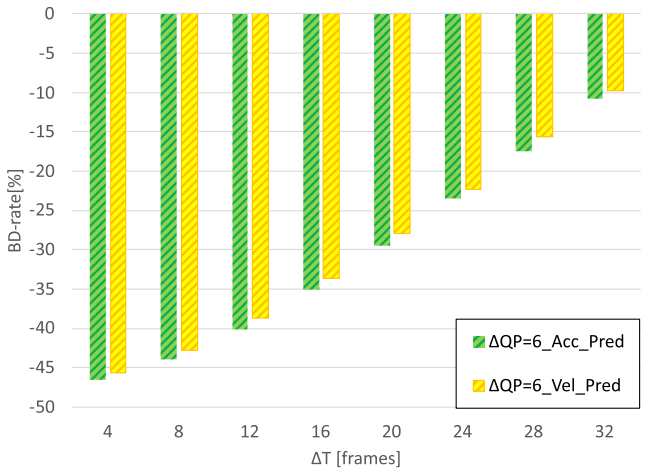

Fig. 8. BD-rate gains for a ΔQP of 6 compared with the viewport-independent approach for the acceleration-based prediction.

TABLE IV
Error Intervals for Yaw and Pitch in Degrees

	Yaw		Pitch	
j	$e_{j,min}$	$e_{j,max}$	$e_{j,min}$	$e_{j,max}$
0	0	7.5	0	5
1	7.5	12	5	10
2	12	17.5	10	15
3	17.5	30	15	30
4	30	45	30	90
5	45	60		
6	60	90		
7	90	180		

using a ΔQP of 6 lead to a worse performance than using a ΔQP of 0 for those large values of angular velocity and acceleration.

Consequently, an algorithm has been developed that tackles this issue. The idea is to define some confidence values or to derive some confidence values for the performed prediction to determine which tiles, if any, of the low-resolution tiles are downloaded with ΔQP 6 and which with ΔQP 0. Then, based on the $\omega'_{avg}(t)$ and ΔT, a mapping is carried out to a confidence-value interval that determines the QP distribution of the low-resolution tiles and with it the corresponding ΔQP of each of the tiles.

First, several error intervals have been defined $ei_j = [e_{j,min}, e_{j,max})$ for which a look up table is derived that maps a $\omega'_{avg}(t)$ and ΔT to a given ei_j with a confidence value of 95%. In other words, the probability that the error, which results from the viewport prediction, for a $\omega'_{avg}(t)$ and ΔT lies within the interval $[e_{j,min}, e_{j,max})$ is 95%. For that purpose, the traces and corresponding prediction errors have been statistically analysed and for each ΔT, velocity thresholds have been derived $\omega_j(\Delta T)$ so that any $\omega'_{avg}(t)$ that fulfils $\omega_{j-1}(\Delta T) < \omega'_{avg}(t) \leq \omega_{j-1}(\Delta T)$ leads to a prediction error $e \in ei_j$ with 95% probability.

The error intervals described above have been selected differently for yaw and for pitch as shown in Table IV.

TABLE V
EXTENDED FOV FOR YAW AND PITCH IN DEGREES

	Yaw	Pitch
j	$FoVExt_x$	$FoVExt_y$
0	0	0
1	2.5	2.5
2	7.5	5
3	15	7.5
4	22.5	12.5
5	30	
6	45	
7	50	

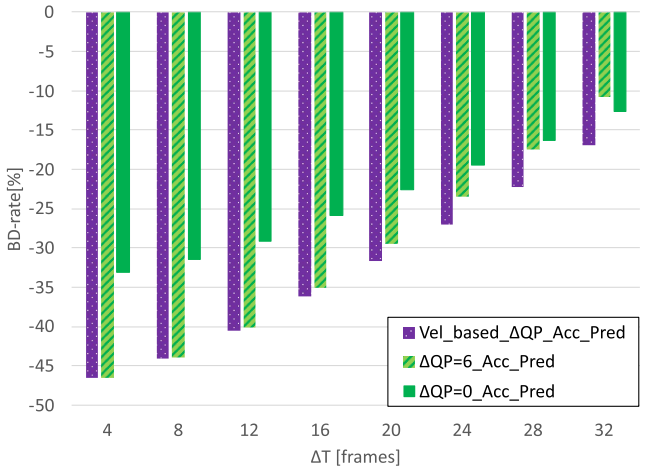

Fig. 9. BD-rate gains for the velocity-based QP distribution compared with the viewport-independent approach for the acceleration-based prediction.

The described algorithm defines a prefetch area, i.e. an extension of the FoV, ($FoVExt_x$, $FoVExt_y$) based on the error intervals in Table IV, for which the corresponding tiles are downloaded with a ΔQP of 0. Tiles outside the extended FoV, i.e. ($FoV_x + FoVExt_x$, $FoV_y + FoVExt_y$), are then downloaded with a ΔQP of 6. The values of ($FoVExt_x$, $FoVExt_y$) have been empirically found to be optimal and are shown in Table V.

For the results shown in the following the low-resolution tiles are encoded at 8 QP values: namely 22, 27, 28, 32, 33, 37, 38, and 43.

Fig. 9. shows the results of using the velocity-based QP distribution algorithm on top of the acceleration-based prediction, which is represented with purple bars. The results are compared with the acceleration-based prediction using a static ΔQP of 6 and 0. It can be seen that the proposed algorithm outperforms the previous results and provides BD-rate gains compared to the viewport-independent approach from 16% for $\Delta T = 32$ up to gains around 46% for $\Delta T = 4$. Besides, the figure shows that for high values of ΔT gains of almost 5% can be achieved compared to using a static ΔQP configuration.

VII. RESULTS FOR MOTION CONSTRAINT TILES

The results shown in the previous figures correspond to cases where tiles have been encoded independently,

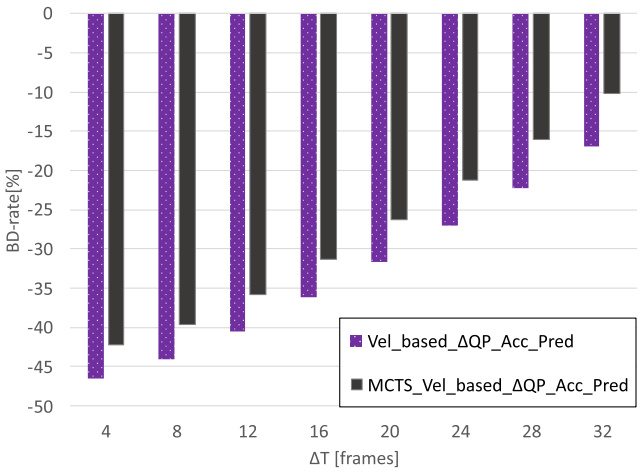

Fig. 10. BD-rate gains for the velocity-based QP distribution compared with the viewport-independent approach for the acceleration-based prediction with MCTS.

downloaded simultaneously and decoded using parallel decoders. It has been already discussed in the past (see, e.g. [38]) that requiring parallel decoding of tiles might be detrimental, since most of the deployed devices only have a single hardware decoder, and thus software decoding might be required draining the battery of the devices.

In order to solve such an issue, an alternative is to encode each of the tiles as a Motion Constraint Tile Set (MCTS), which implies encoding the content following a set of constraints. Such constraints lead to a slightly lower coding efficiency but allows aggregating tiles into a single bitstream that can be decoded with a single decoder, thus enabling implementation of tile-based streaming in deployed devices. For more information, the reader is referred to [38].

Fig. 10. Shows a comparison of the performance for the velocity-based QP distribution algorithm with the acceleration-based prediction with tiles encoded as MCTS and encoded normally, i.e. with no constraints. It can be seen that when MCTS is used, a drop in the gain of around 4% up to 6% occurs. Still, significant gains from 10% up to 42% are achieved in comparison to the viewport-independent approach and using MCTS has the benefit that it allows using a single hardware decoder.

VIII. CONCLUSION

In this paper, tile-based streaming for 360° video is presented and analysed. More concretely, the system setup consists of offering tiles at various resolutions so that each user can retrieve the tiles that best match its viewport, i.e. high-resolution tiles for that which correspond to the viewport and low-resolution tiles for the tiles that do not correspond to the viewport. The paper analyses the visual quality at the viewport based on the end-to-end delay. Since the delay has an impact on the time needed by clients to adapt to user movements so that movements are reflected on the retrieved content, the paper studies the impact of the delay and introduces various means to reduce the impact on the observed fidelity. The proposed solution shows that gains up to 46% compared

to the viewport-independent approach can be achieved for very low end-to-end delays. Still significant gains can be obtained up to an end-to-end delay 1 second.

REFERENCES

[1] *Oculus Rift*. Accessed: Jun. 22, 2018. [Online]. Available: http://www.oculus.com/rift/

[2] *Google Cardboard*. Accessed: Jun. 22, 2018. [Online]. Available: https://www.google.com/get/cardboard/

[3] *Google Daydream*. Accessed: Jun. 22, 2018. [Online]. Available: https://www.google.com/get/daydream/

[4] *HTC Vive*. Accessed: Jun. 22, 2018. [Online]. Available: https://www.vive.com/ca/

[5] *Playstation VR*. Accessed: Jun. 22, 2018. [Online]. Available: https://www.playstation.com/en-ca/explore/playstation-vr/

[6] *Gear VR*. Accessed: Jun. 22, 2018. [Online]. Available: http://www.samsung.com/global/microsite/gearvr/

[7] *White Paper: Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*, Cisco Visual Netw. Index, San Jose, CA, USA, Jun. 6, 2017.

[8] *GoPro*. Accessed: Jun. 22, 2018. [Online]. Available: https://vr.gopro.com

[9] *Google Odyssey*. Accessed: Jun. 22, 2018. [Online]. Available: https://gopro.com/odyssey

[10] *Samsung Project Beyond*. Accessed: Jun. 22, 2018. [Online]. Available: http://thinktankteam.info/beyond

[11] *Facebook Surround 360*. Accessed: Jun. 22, 2018. [Online]. Available: https://facebook360.fb.com/learn/#s3

[12] *Facebook*. Accessed: Jun. 22, 2018. [Online]. Available: https://www.facebook.com

[13] *Youtube*. Accessed: Jun. 22, 2018. [Online]. Available: https://www.youtube.com

[14] *3GPP Virtual Reality Profiles for Streaming Applications*, document 3GPP TS.26.118, 2018.

[15] *Additional Supplemental Enhancement Information*, Standard ISO/IEC 23008-2:2017/Amd 3:2018, 2018.

[16] *Information Technology—Coded Representation of Immersive Media—Part 2: Omnidirectional Media Format*, Standard ISO/IEC 23090-2, 2019.

[17] (2018). *VR-IF Guidelines Version 1.0*. [Online]. Available: https://www.vr-if.org/wp-content/uploads/vrif2018.018.09-clean.pdf

[18] S. McCarthy, "Quantitative Evaluation of Human Visual Perception for Multiple Screens and Multiple CODECs," in *Proc. SMPTE Annu. Tech. Conf. Exhib.*, 2012, pp. 1–10.

[19] *Equirectangular Projection*. Accessed: Jun. 22, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Equirectangular_projection

[20] K.-T. Ng, S.-C. Chan, and H.-Y. Shum, "Data compression and transmission aspects of panoramic videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 82–95, Jan. 2005.

[21] *Pyramid Projection*. Accessed: Jun. 22, 2018. [Online]. Available: https://cod.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/

[22] G. Van der Auwera, M. Coban, Hendry, M. Karczewicz, *AHG8: Truncated Square Pyramid Projection (TSP) For 360 Video*, document JVET-D0071, 4th JVET Meeting, Chengdu, China, 2016.

[23] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, San Jose, CA, USA, Dec. 2016, pp. 583–586.

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

[25] C. Grünheit, A. Smolic, and T. Wiegand, "Efficient representation and interactive streaming of high-resolution panoramic views," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Rochester, NY, USA, Sep. 2002, p. 3.

[26] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proc. ACM Multimedia Conf. (MM)*, 2016, pp. 601–605.

[27] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using MPEG-DASH," in *Proc. ACM 7th Int. Conf. Multimedia Syst.*, 2016, Art. no. 41.

[28] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 107–110.

[29] Y. Sanchez *et al.*, "Efficient HTTP-based streaming using scalable video coding," *Signal Process., Image Commun.*, vol. 27, no. 4, pp. 329–342, Nov. 2011.

[30] S. M. LaValle, A. Yershova, M. Katsev, and M. Antonov, "Head tracking for the Oculus Rift," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 187–194.

[31] R. T. Azuma, "Predictive tracking for augmented reality," Ph.D. dissertation, Dept. Comput. Sci., Univ. North Carolina Chapel Hill, Chapel Hill, NC, USA, Feb. 1995.

[32] T. C. Nguyen and J.-H. Yun, "Predictive tile selection for 360-degree VR video streaming in bandwidth-limited networks," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1858–1861, Sep. 2018.

[33] F. Qian, B. Han, L. Ji, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. 5th Workshop Things Cellular, Oper., Appl. Challenges (ATC)*, 2016, pp. 1–6.

[34] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming," in *Proc. ACM Multimedia Conf. (MM)*, 2017, pp. 315–323.

[35] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov, and A. Kondoz, "Predicting head trajectories in 360° virtual reality videos," in *Proc. Int. Conf. 3D Immersion (IC3D)*, 2017, pp. 1–6.

[36] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1161–1170.

[37] R. Skupin, Y. Sánchez, D. Podborski, C. Hellge, and T. Schierl, "HEVC tile based streaming to head mounted displays," in *Proc. 14th Annu. IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 2017, pp. 8–11.

[38] *Information Technology—Coded Representation of Immersive Media—Part 2: Omnidirectional Media Format*, Standard ISO/IEC 23090-2, 2019.

[39] Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "Shifted IDR representations for low delay live DASH streaming using HEVC tiles," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 87–92.

[40] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *Proc. VCEG-M33 Meeting*, 2001 pp. 1–4.

Yago Sánchez de la Fuente received the M.Sc.Ing. degree in telecommunications engineering from the Tecnun-Universidad de Navarra, Spain, in 2009. From 2008 to 2009, he carried out his master thesis on P2P application layer multicast using SVC at the Fraunhofer Heinrich-Hertz-Institute (HHI).

In 2013, he was visiting the End2End Mobile Video Research Group, Alcatel Lucent-Bell Labs, Murray Hill, NJ, USA, where he was doing research on mobile video delivery optimization for low delay HTTP streaming. He is currently a Researcher with the Image Communication Group of Prof. Thomas Wiegand, Technische Universität Berlin, and is a Guest Researcher at Fraunhofer HHI. In addition, he is the co-author of the *IETF RTP Payload Format for H.265/HEVC Video*. His research interests include adaptive streaming services for IPTV and OTT services. He is working on adaptive streaming services for Mobile TV over LTE networks and Virtual Reality 360° video.

Gurdeep Singh Bhullar received the B.Tech. degree in electronics and communication engineering from Guru Gobind Singh Indraprastha University, New Delhi, India, in 2016. He is currently pursuing the M.Sc. degree in broadcast engineering with Birmingham City University, Birmingham, U.K.

Since 2017, he has been with the Multimedia Communications Research Group, Video Coding and Analytics Department of Dr. Thomas Schierl and Dr. Detlev Marpe, Fraunhofer Heinrich-Hertz-Institute, Berlin, Germany. His research interests lie in the area of video data coding and analytics. He is currently focusing on adaptive streaming services for 360° video delivery.

Robert Skupin received the Dipl.Ing. (FH) degree in electrical engineering from BRSU, Sankt Augustin, Germany, in 2009, and the M.Sc. degree in computer engineering from the Technical University of Berlin, Germany, in 2014.

Since 2009, he has been with the Multimedia Communications Group, Video Coding and Analytics Department of Dr. Thomas Schierl and Dr. Detlev Marpe, Fraunhofer Heinrich Hertz Institute, Berlin, Germany. His research interests lie in the area of video data coding and transport. Currently, his work is focused on techniques toward high quality 360° video services with HEVC and beyond. In addition, he is the Editor of the third amendment of [16] ISO/IEC 23008-2:2017. He is an active participant in standardization activities in JCT-VC and MPEG and has as such successfully contributed to the HEVC and MPEG-DASH. He has also technically contributed to various scientific and industry-funded research projects such as SVConS and COAST in the EU FP7 program, and is steering work in the German BMBF project CODEPAN.

Cornelius Hellge received the Dipl.Ing. degree in media technology from the Ilmenau University of Technology, in 2006, and the Dr. Ing. degree with distinction *(summa cum laude)* from the Berlin University of Technology, in 2013.

In 2014, he was a Visiting Researcher at the Network Coding and Reliable Communications Group of Prof. Medard, Massachusetts Institute of Technology. Since 2015, he has been heading the Multimedia Communications Group, Fraunhofer Heinrich Hertz Institute. He is responsible for various scientific as well as industry-funded projects. Together with his team, he is regularly contributing to 3GPP (RAN1, RAN2, SA4), MPEG, JCT-VC, and JVET.

Dr. Hellge received the Best Paper Award from the IEEE ICCE'14 Conference and has authored over 60 journal and conference papers, predominantly in the area of video communication and next generation networks; moreover, he holds more than 30 internationally issued patents and patent applications in these fields.

Thomas Schierl (M'03–SM'17) received the Dipl.Ing. degree in computer engineering from the Berlin University of Technology (TUB), Germany, in 2003, and the Dr. Ing. degree in electrical engineering and computer science (passed with distinction) from TUB, in 2010.

He is heading the Department of Video Coding and Analytics, Fraunhofer HHI, Berlin, Germany. He is the co-author of various IETF RFCs, as well as the Co-Editor of the various ISO MPEG Standards. He and his team, as part of JCT-VC, also contributed to the standardization process of MPEG - HEVC / ITU-T Rec. H.265, mainly in the area of high level parallelism and high level syntax. His research interests include system integration of video codecs, as well as the delivery of real-time media over mobile IP networks, such as mobile media content delivery over HTTP.

Dr. Schierl received together with the ISO/IEC JCT1/SC29/WG11 Moving Picture Experts Group (MPEG), the Technology and Engineering Emmy Award by the National Academy of Television Arts and Sciences for the Development of the MPEG-2 Transport Stream, in 2014.