RANDOM ACCESS POINT PERIOD OPTIMIZATION FOR VIEWPORT ADAPTIVE TILE BASED STREAMING OF 360° VIDEO

Y. Sanchez, R. Skupin, C. Hellge, T. Schierl

Fraunhofer HHI, Einsteinufer 37, 10587 Berlin

ABSTRACT

360° video streaming introduces stricter requirements to the established transmission chain than in traditional streaming services. Transmission of the complete 360° video in desirable quality can lead to multiple times UHD resolution which would waste a large fraction of network resources as the larger fraction of the video is not presented on the end device. Adaptivity to the user viewport promises substantial benefits and contrary to per-user or per-orientation encoding, tiled based streaming provides a scalable solution. We consider tiled streaming of 360° video using the cubic projection, where tiled content resides on the server at 2 different resolutions. User traces have been collected for a range of content and based on consumption patterns, tiles at the server have been optimized. More concretely, the paper optimizes the Random Access Point period that leads to the minimum transmitted bitrate, while ensuring that users watch most of the time high resolution content.

Index Terms— Virtual Reality, 360 video, tiles, HEVC, viewport adaptive streaming

1. INTRODUCTION

The recent market availability of plenty of Head Mounted Displays (HMD) such as the Oculus Rift [1], Samsung Gear VR [2] or Google Cardboard [3], is boosting the popularity of interactive streaming services, which we envision they will become ubiquitous in a near future. In fact, major platforms such as Youtube and Facebook are already streaming 360° video to various devices.

VR streaming, aka. as 360° video streaming, is understood as omnidirectional video content consumed with HMDs. For such applications, the head pose may change considerably within milliseconds and with it the content shown to the user (i.e. viewport). This makes solutions that consist of transmitting only a subset of the video inadequate and therefore, the complete 360° video content needs to be transmitted permanently. Current deployments of 360° video streaming service suffer from the problem that they transmit the content in a user viewport agnostic way, i.e. covering the full 360° video at same quality. This approach sacrifices a substantial amount of throughput for pixels that are not even presented to the user. A better solution can be provided with viewport adaptive coding and transmission schemes that allow achieving a desirable visual quality within the user viewport by downloading a lower quality for the pixels that do not belong to the viewport so that there is some content in case it needs to be shown.

Adaptive streaming is currently the dominant means for distribution of video on demand. Specifically, the MPEG DASH [4] standard has seen major deployment in recent years, e.g. Youtube and Netflix. While adapting the video bitrate to available throughput is most important for traditional video content, 360° video streaming requires further approaches, i.e. viewport adaptive solutions. The most straight-forward solution, consist in performing per-orientation encodings, as for instance implemented on a large scale at Facebook [5]. With this solution, numerous versions are encoded and made available for different viewport-orientations with a high visual quality within that orientations. However, this approach comes along with significant overhead for the generation, encoding and storage of as much as multiple dozen versions of the same content.

Alternatively, tile based streaming [6] can be used. The main idea is to divide the picture horizontally and vertically into smaller regions that are encoded independently. Then the regions that contain the content belonging to the viewport can be transmitted at higher quality to the user. With tiled streaming, the number of streams encoded only depends on the granularity, with which the content has been tiled, and not on the potential viewport-orientations. Recently, tiled streaming has become a very popular topic within the research community and standardization bodies. In fact, MPEG has standardized a solution for Dynamic Adaptive Streaming over HTTP (DASH) that allows for tiled streaming ([7]). Furthermore, many papers have been published in the last years that focus on optimization of the tiles [8] or on video delivery of panorama videos using tiles [9] or usage of tiles for 360° video[10]. However, one issue that to our knowledge has not yet been properly tackled is how set the RAP period of the tiles for each video so that the transmitted bitrate is optimized.

The remainder of the paper is organized as follows. Section 2 provides an overview of the problem description. Section 3 describes the content and obtained user traces for the evaluations. In Section 4 the performed optimization is explained and Section 5 presents the results. Finally, Section 6 provides the conclusion of the paper.

2. PROBLEM DESCRIPTION

A possible tile based streaming system based on DASH and HEVC was presented in [11]. The system uses the cubic projection to represent the 360° video, i.e. the 360 scene is mapped to the six faces of a cube around a central camera point via rectilinear projection and the faces are arranged in a common plane of the video picture. The video is then sampled to various resolutions and tiled at a desired granularity (see Figure 1). Encoding of the tiles then employs a feature of HEVC referred to as motion constrained tiles, which avoids coding and prediction dependencies between individual tiles. The constrains allow usage of the so called Compressed Domain Tile Aggregation [11] in which the selection of video tiles is merged into a single HEVC bitstream on client side enabling thus usage of a single decoder, e.g. the hardware decoder of a mobile phone.

On server side, all tile bitstreams are offered and the client can select which tiles to download at a high resolution and which tiles at low resolution, as illustrated in Figure 1.

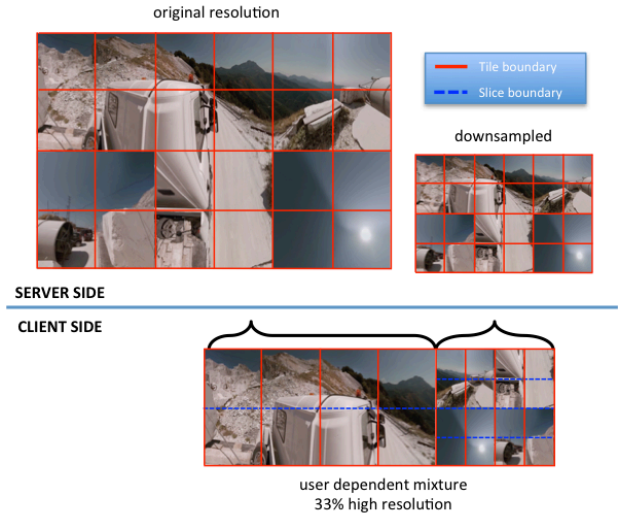

Figure 1: Tiling of cubic video at different resolutions

In this work, video tiles of the complete 360° video content are downloaded constantly, albeit at varying resolution, so that at least a low-quality content variant can be presented to the user in cases of rapid orientation changes. The tile resolution selection can be adjusted to changes of the user head orientation but only at so-called random access points (RAPs), since RAPs break prediction dependencies to preceding pictures by containing only intra coded blocks. The tiling granularity and the used RAP distance thereby determine the responsivity with which the tile selection can be changed. As the user changes his orientation, some tiles wander out of the viewport and are replaced by new tiles. These new tiles may be presented to the user in a low-resolution variant and are only switched to the high-resolution variant at the next available RAP. The time duration for which it is considered to be acceptable to show low resolution content is referred to as D_{LQ} in the following.

Similar to the work in [8] the target of this paper is to optimize the way the content is generated to minimize the bitrate transmitted to the users. However, in [8], the authors discuss on the optimal tile sizes in pixels and derive a model based on pixel overhead η and bitrate per pixel φ. In this paper, we optimize the RAP period for a given tile granularity. The reason for that is that we focus on VR streaming where the full content is transmitted instead of a subset of all tiles. Therefore, we consider that the biggest impact to the bitrate of the downloaded video is bound to the RAP period instead of the sizes of the tiles (tile granularity). The optimization problem is shown in Eq. (1):

$$\min R = \min_{\Delta t}(T_{HR}(\Delta t) * BR_{HR}(\Delta t) + T_{LR}(\Delta t) * BR_{LR}(\Delta t))$$
$$\text{subject to: } P(D_{LQ}) \leq 90\% \quad (1)$$

where $T_{HR}(\Delta t)$ and $T_{LR}(\Delta t)$ are the number of tiles of high and low resolution respectively that are required for a given RAP period (Δt) so that at least 90% of the time low resolution tiles are not shown to the user for longer than D_{LQ}. $BR_{HR}(\Delta t)$ and $BR_{LR}(\Delta t)$ correspond to the average bitrate of the high and low resolution tiles respectively for a given RAP period (Δt).

3. CONTENT AND USER TRACES

The results of this paper were obtained using a test set of 360° video sequences, provided by Deep Inc and GoPro. The test sequence set is selected to cover a wide range of spatio-temporal activity characteristics (see Figure 2), as indicated in [12].

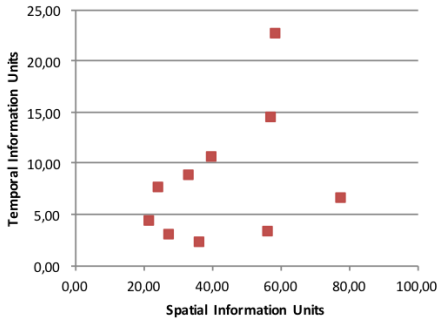

Figure 2: Spatio-temporal activity of test sequences

We have used 10 sequences in cube projection format that have a resolution of 3072x2048 pixels, i.e. 1024x1024 per cube face with framerate of 25 fps or 30 fps. For each of the test sequences, user traces were recorded from 17 test subjects using an OculusRift CV1 HMD. Figure 3 shows two plots of a user trace, namely the yaw, pitch and roll angle as well as the measured angular velocity omega over frames.

4. OPTIMIZATION

In order to solve Eq. (1), we first derive, based on Eq. (2), the number of high and low resolution tiles for the worst case,

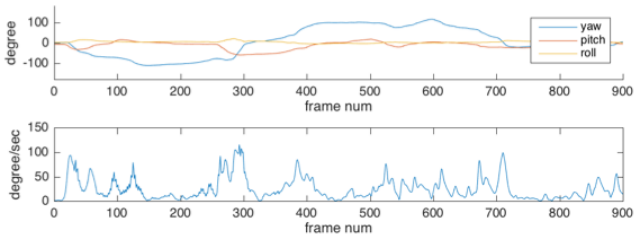

Figure 3: User trace example

i.e. for the yaw (φ), pitch (δ) and roll (ϕ) of the viewport orientation that leads to the biggest $T_{HR}(\Delta t)$.With the computed $T_{HR}(\Delta t)$, the content is shown at lowest resolution at most for the duration of D_{LQ} with a probability of $P(D_{LQ}) \leq 90\%$. This is done by defining a prefetch area, i.e. the tiles outside the current viewport that would be needed if the viewer moved with on a horizontal and vertical velocity v_h and v_v. v_h and v_v are defined as the 90 percentile of the horizontal and vertical velocities based on the recorded user traces to ensure that $P(D_{LQ}) \leq 90\%$.

In order to derive $T_{HR}(\Delta t)$, we define first $V=\{V_1,V_2,\ldots,V_M\}$ as the set of vertices of the tiles which the cube has been divided into. For the cube faces divided into NxN tiles, there are $M=6N^2+2$ vertices. In order to find the worst case, we look for (φ, δ, ϕ) that maximize the metric I, which has a maximum when the number of vertices into the viewport is maximized (f(.) in Eq.3 is 0 if vertices are not in the viewport):

$$I_{max}(\Delta t, D_{LQ}) = \max_{\varphi,\delta,\phi} \sum_{i=1}^{M} f\left(\overrightarrow{Ver_i},\overrightarrow{VP}\,(\varphi,\delta,\phi), \Delta t, D_{LQ}\right) \quad (2)$$

where

$$f\left(\overrightarrow{Ver_i},\overrightarrow{VP}\,(\varphi,\delta,\phi), \Delta t, D_{LQ}\right) = \begin{cases} \overrightarrow{Ver_i} * \overrightarrow{VP}\,(\varphi,\delta,\phi) & if\ \alpha < \frac{\pi}{4} + v_h * \Delta T_{high} \ \ and \\ & \beta < \text{atan}\,(\cos\,(\alpha)) + v_v * \Delta T_{high}) \\ & with\ \Delta T_{high} = \max(0, \Delta t - D_{LQ}) \\ 0 & else \end{cases} \quad (3)$$

where $\overrightarrow{Ver_i}$ is the unitary vector pointing to the vertex V_i and $\overrightarrow{VP}\,(\varphi,\delta,\phi)$ is the unitary vector of the center of the viewport corresponding to the extrinsic Euler angles (φ, δ, ϕ), which representation in Cartesian coordinates is $\overrightarrow{VP_{xyz}}\,(\varphi,\delta,\phi) = [\cos(\varphi) * cos(\delta),\ \sin(\varphi) * cos(\delta),\ \sin\,(\delta)]$. v_h and v_v are the horizontal and vertical angular velocities aforementioned and α and β are the horizontal and vertical angles formed by the unitary vectors $\overrightarrow{Ver_i}=[x_i, y_i, z_i]$ and $\overrightarrow{VP}\,(\varphi,\delta,\phi)$. They can be calculated by computing $\overrightarrow{Ver'_i}=[x_i', y_i', z_i']$ in the rotated coordinate space of the rotation corresponding to the viewport $\overrightarrow{VP}\,(\varphi,\delta,\phi)$ and deriving the horizontal and vertical angle in the new space as follows.

$$\alpha = \text{acos}\left(\frac{x_i'}{\sqrt{{x_i'}^2+{y_i'}^2}}\right) \text{and}\ \beta = \text{atan}\left(\frac{z_i'}{\sqrt{{x_i'}^2+{y_i'}^2}}\right) \quad (4)$$

with,

$$x_i' = \cos(\varphi)cos(\delta)x_i + \sin(\varphi)cos(\delta)y_i - sin(\delta)z_i$$

$$y_M' = (\cos(\varphi)sin(\delta)\sin(\phi) - \sin(\varphi)\cos(\phi))x_i + (\sin(\varphi)sin(\delta)\sin(\phi) + \cos(\varphi)\cos(\phi))y_i + cos(\delta)\sin(\phi)\,z_i$$

$$z_M' = (\cos(\varphi)sin(\delta)\cos(\phi) + \sin(\varphi)\sin(\phi))x_i + (\sin(\varphi)sin(\delta)\cos(\phi) - \cos(\varphi)\sin(\phi))y_i + cos(\delta)\cos(\phi)\,z_i$$

Then, once we have the φ, δ and ϕ for I_{max} for each possible RAP period (Δt) and a given D_{LQ}, we compute the number of high resolution tiles ($T_{HR}(\Delta t)$) that lay into the viewport and prefetch area. Since $T_{HR}(\Delta t) + T_{LR}(\Delta t) = 6*N^2$, we derive $T_{LR}(\Delta t)$ as well. Then, with $T_{HR}(\Delta t)$ and $T_{LR}(\Delta t)$ we can compute the optimal solution solving Eq. (1).

5. RESULTS

In this section, we show the results for the sequences described in section 3. Two tiling granularity versions, i.e. number of tiles which each face of the cube has been tiled into, have been tested: 2x2 and 4x4. This accounts to 24 and 96 tiles respectively. In addition, the videos have been encoded at the original resolution and at half of the original resolution with several RAP periods {8,16,24,32 frames}. A single downsampling factor of 1:2 is considered for brevity.

With these encodings, the values of $BR_{HR}(\Delta t)$ and $BR_{LR}(\Delta t)$ have been computed for 2 values of QP: 27 and 32. Then, together with $T_{HR}(\Delta t)$ and $T_{LR}(\Delta t)$, derived as described in section 4, the optimal solution for Eq. (1) has been computed. We have investigated different values for the D_{LQ} constraint: 50, 100, 150, 200, 250 and 300 ms. In Figure 4, we show $T_{HR}(\Delta t)$ for one of the sequences for the 2 tiling granularities.

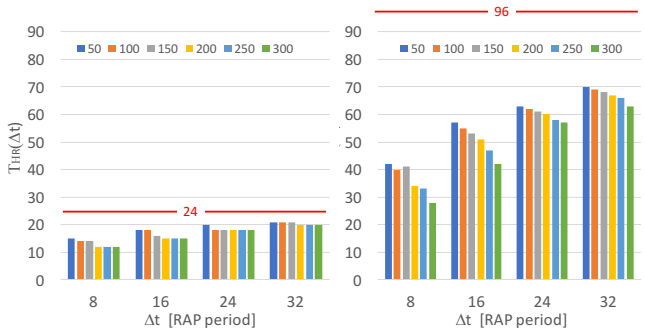

Figure 4: Minimum T_{HR} fulfilling D_{LQ} constraint (50-300 ms): 2x2 granularity (left) and 4x4 (right).

It can be seen that for the low granularity solution (24 tiles), the number of high resolution tiles accounts from 50% of the tiles (Δt =8 frames) up to 87,5% of the tiles (Δt =32

frames and D_{LQ} = 50-150 ms). For the finer granularity solution, $T_{HR}(\Delta t)$ lays on the range of 35,4%-72,9%. Overall, the higher the value of Δt, the higher is T_{HR} and higher values of D_{LQ} or higher number of tiles the lower is $T_{HR}(\Delta t)$.

Table 1 shows the optimal value of Δt (Δt_{opt}) for the different experiments and sequences. It can be seen that the values differ a lot from sequence to sequence

Table 1: Δt_{opt} for 2x2 and 4x4 tiling and QP 27 and 32

QP	G	S_1	S_2	S_3	S_4	S_5
27	2x2	8	32	24*	32	32
	4x4	8	32	16*	32	24*
32	2x2	8*	32	24*	32	32
	4x4	8	32	24*	32	32*
QP	G	S_6	S_7	S_8	S_9	S_{10}
27	2x2	32	32	24*	8	32
	4x4	32	32	24*	8	32
32	2x2	32	32	32*	8*	32
	4x4	32	32	32*	8	32

(*)=The optimal value of Δt varies for different D_{LQ}. The value that has a lower overhead compared to the optimal value for each D_{LQ} is shown.

There are some sequences for which Δt_{opt} is different for different values of D_{LQ} (see * in the table). For those, Δt_{opt} for different values of D_{LQ} is shown in Table 2.

Table 2: Δt_{opt} varying with D_{LQ}

	D_{LQ} (ms)					
S_n_QP_G	50	100	150	200	250	300
S_1_32_2x2	8	8	16	8	8	8
S_3_27_2x2	24	24	24	24	24	16
S_3_27_4x4	16	24	16	16	8	8
S_3_32_2x2	24	32	24	24	24	16
S_3_32_4x4	24	24	24	24	24	8
S_5_27_4x4	32	32	24	24	24	24
S_5_32_4x4	32	32	32	24	24	24
S_8_27_2x2	32	32	32	24	24	24
S_8_27_4x4	24	32	32	24	24	24
S_8_32_2x2	32	32	32	24	24	24
S_8_32_4x4	32	32	32	24	24	24
S_9_32_2x2	8	8	16	8	8	8

We have computed the average overhead (penalty) over all sequences when choosing the values shown in Table 1 instead of in Table 2 and the overhead lays between 0.3% and 2.7% depending on the QP values and tiling granularity. However, it can reach 7% for specific sequences and configurations.

Figure 5 shows the average bitrate saving of the proposed optimization in comparison to a static case where the RAP period is the same for all sequences. It shows that the worst configuration corresponds to a RAP period of 8, for which the solution within this paper improves the transmitted bitrate in about 20%-30%. The best configuration corresponds to a RAP period of 32. Still the proposed solution provides bitrate savings of 3%-5% on average.

Figure 6 shows similar results as in Figure 5 but in terms of maximum bitrate saving instead of average values. It can be seen that bitrate savings of up to 20%-26% are achieved in comparison to RAP period of 32. The difference compared to Figure 5 is due to the fact that RAP period 32 is the best configuration for many sequences (see Table 1) and thus the average savings are not that high. But when looking at specific sequences it can be seen that very big gains can be achieved by performing the optimization presented in this paper.

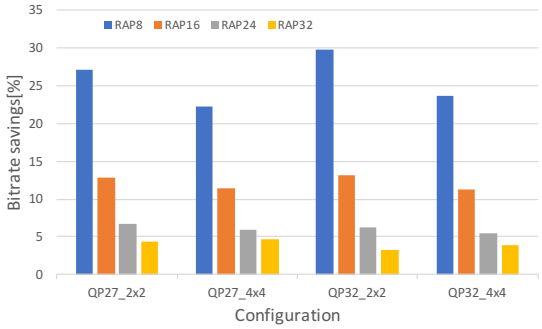

Figure 5: Avg. Bitrate savings over static configuration

Finally, we compared the bitrates for the different tiling granularities. Tiling the content with 4x4 tiles instead of 2x2 leads to an increase of the bitrate of 0.5% and 7.2% for QP27 and QP32 respectively. However, in terms of resolution, the former leads to a reduction of 13.6% of the video size, which could be beneficial to reduce the decoding complexity.

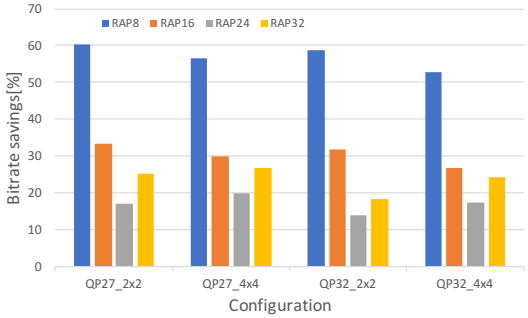

Figure 6: Max. Bitrate savings over static configuration

6. CONCLUSION

This paper describes an optimization problem of tiled streaming for 360 video that targets finding out the Random Access Point (RAP) period for each sequence that leads to the minimum transmitted bitrate, while ensuring that users watch most of the time high resolution content. With the proposed optimization, average bitrate saving from around 3%-5% up to 30% can be obtained in comparison with static RAP period configurations and for specific sequences from around 12% up to 60% depending on the RAP period configuration.

7. REFERENCES

[1] http://www.oculus.com/rift/. Accessed 18 December 2015.

[2] http://www.samsung.com/global/microsite/gearvr/. Accessed 18 December 2015.

[3] https://www.google.com/get/cardboard/. Accessed 18 December 2015.

[4] Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part1: Media presentation description and segment formats, ISO/IEC 23009-1:2014.

[5] E. Kuzyakov, D. Pio, "Next-generation video encoding techniques for 360 video and VR", Retrieved from: https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/, 2016.

[6] C. Grünheit, A. Smolic, and T. Wiegand: "Efficient Representation and Interactive Streaming of High-Resolution Panoramic Views", *IEEE International Conference on Image Processing (ICIP'02)*, Rochester, NY, USA, September 2002.

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

[8] A. Mavlankar, B. Girod, "Spatial-Random-Access-Enabled Video Coding for Interactive Virtual Pan/Tilt/Zoom Functionality", IEEE Transactions on Circuits and Systems for Video Technology (CSVT), Vol. 21, No. 5, May 2011.

[9] J. Le Feuvre, C. Concolato, "Tiled-based Adaptive Streaming using MPEG-DASH", ACM Multimedia Systems 2016, May 2016, Austria.

[10] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Viewport-Adaptive Navigable 360-Degree Video Delivery", in Proc. IEEE Int'l Conf. Communications, Paris, France, May 2017.

[11] R. Skupin, Y. Sánchez, D. Podborski, C. Hellge, T. Schierl, "HEVC Tile Based Streaming to Head Mounted Displays", 14th Annual IEEE Consumer Communications & Networking Conference, 8-11 January 2017, Las Vegas, USA.

[12] ITU-T P.910. "Subjective video quality assessment methods for multimedia applications", April 2008.