Efficient Encoding of Interactive Personalized Views Extracted from Immersive Video Content

Johan De Praeter, Pieter Duchi, Glenn Van Wallendael, Jean-Francois Macq, and Peter Lambert

In: Proceedings of the 1st International Workshop on Multimedia Alternate Realities, 25–30, 2016.

http://doi.acm.org/10.1145/2983298.2983301

Efficient Encoding of Interactive Personalized Views Extracted from Immersive Video Content

Johan De Praeter
Ghent University - iMinds
Sint-Pietersnieuwstraat 41
9000 Ghent
johan.depraeter@ugent.be

Pieter Duchi
Ghent University - iMinds
Sint-Pietersnieuwstraat 41
9000 Ghent
pieter.duchi@ugent.be

Glenn Van Wallendael
Ghent University - iMinds
Sint-Pietersnieuwstraat 41
9000 Ghent
glenn.vanwallendael@ugent.be

Jean-Francois Macq
Nokia Bell Labs
Copernicuslaan 50
2018 Antwerp, Belgium
jean-francois.macq@nokia.com

Peter Lambert
Ghent University - iMinds
Sint-Pietersnieuwstraat 41
9000 Ghent
peter.lambert@ugent.be

ABSTRACT

Traditional television limits people to a single viewpoint. However, with new technologies such as virtual reality glasses, the way in which people experience video will change. Instead of being limited to a single viewpoint, people will demand a more immersive experience that gives them a sense of being present in a sports stadium, a concert hall, or at other events. To satisfy these users, video such as 360-degree or panoramic video needs to be transported to their homes. Since these videos have an extremely high resolution, sending the entire video requires a high bandwidth capacity and also results in a high decoding complexity at the viewer. The traditional approach to this problem is to split the original video into tiles and only send the required tiles to the viewer. However, this approach still has a large bit rate overhead compared to sending only the required view. Therefore, we propose to send only a personalized view to each user. Since this paper focuses on reducing the computational cost of such a system, we accelerate the encoding of each personalized view based on coding information obtained from a pre-analysis on the entire ultra-high-resolution video. By doing this using the High Efficiency Video Coding Test Model (HM), the complexity of each individual encode of a personalized view is reduced by more than 96.5% compared to a full encode of the view. This acceleration results in a bit rate overhead of at most 19.5%, which is smaller compared to the bit rate overhead of the tile-based method.

Keywords

Personalized video, video interaction, High Efficiency Video Coding (HEVC), fast encoding

AltMM'16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2983298.2983301

1. INTRODUCTION

Watching a live broadcast of an event such as a sports match or a music concert through traditional television is a static experience. Contrary to sitting in the stadium or concert hall, the viewer is restricted to a single viewpoint. As a result, viewers are merely observers who are disconnected from the actual event. However, with the advent of new technology such as virtual reality glasses, this static experience will evolve into a more immersive one.

In order to allow users to truly experience this immersion, 360-degree or panoramic video content is necessary. This will allow the user to look around in the scene, unrestricted by a single viewpoint, and will thus increase the level of immersion. However, delivery of such video content to the home is not trivial, since the resolution of this video is far beyond 3840×2160 pixels, which is the highest resolution consumer displays and distribution channels currently support. Resolutions beyond this ultra-high resolution lead to even higher bit rates, which cannot be transported over the limited bandwidth capacity of users at home.

A common approach to the above problem is to split the high-resolution video into tiles, which are separately encoded video streams. A subset of these tiles is then sent to the client depending on the desired view. However, this system still requires either an extra component that will join the separate video streams into one for decoding with a standard decoder, or the client needs multiple decoders to decode the multiple streams. Moreover, depending on the size of the separate tiles, some tiles will contain pixel data that is not displayed within the current view, resulting in a waste of bandwidth capacity.

A better solution is to send only the required view to the client. This means that a separate, personalized video is encoded for each client. Since encoding is computationally complex, such a system requires a high amount of resources. Therefore, as the first main contribution of this paper, we propose a fast personalized-view approach in order to reduce the complexity of each individual encode by using information obtained from a pre-analysis of the entire ultra-high-resolution video. Such an approach typically results in some bit rate overhead. As such, the second main contri-

Figure 1: Illustration of the tile-based method. The tile borders are indicated by the white lines. The red areas are not transmitted to the user. The gray areas indicate the pixels that are not displayed on the user side.

Figure 2: Illustration of the personalized-view method. The red area is not transmitted to the user. Only the required view is sent to the user side.

bution of this paper is to investigate how the overhead of the personalized-view approach compares to the tile-based approach.

In the rest of this paper, we first give an overview of the state-of-the-art in Section 2. In Section 3, we present a short introduction of the coding information in the High Efficiency Video Coding (HEVC) standard which is used in this paper to encode video streams. We describe our proposed fast personalized-view approach in Section 4. This method is evaluated and compared to the tile-based approach in Section 5. Finally the conclusion is given in Section 6.

2. STATE-OF-THE-ART

The first approach to provide users with more interactive video, the tile-based method, has already been thoroughly discussed in the literature [3, 5, 6, 9, 10, 12]. This technique was mostly applied to the H.264/AVC codec. In this approach, the high-resolution video is downsampled at the server to different resolutions (including a thumbnail) in order to provide zooming by having multiple resolution layers. These different layers are then subdivided into a grid of non-overlapping tiles and encoded. At the user side, the user selects his Region of Interest (RoI) based on a thumbnail, which is a low-resolution overview of the entire video, or by using other input methods such as virtual reality glasses. Next, the tiles contained within and intersecting with the RoI boundary for the requested resolution are streamed from the server. These tiles are rendered at the user side and cropped to the appropriate resolution of the display if necessary.

The tile-based method has several disadvantages. A first disadvantage is that tiled streaming results in a bit rate overhead due to sending additional pixels outside the RoI that are not displayed at the user side, as illustrated in Figure 1. This is because some tiles may partially overlap with the RoI, since the RoI is unlikely to be aligned with tile boundaries. To reduce these wasted bits, one can reduce the dimension of the tiles. However, since each tile is encoded independently, small tiles lead to a lower compression ratio, increasing the number of bits needed for the RoI [1]. Another disadvantage is that the user also needs a customized video player to decode, combine and synchronize the tiled streams, which makes this approach harder to deploy. A third disadvantage is that the tiles need to have an encoding structure that allows random access, since the tiles can only be decoded starting from an intra-coded frame. Moreover, the intra-period should be small in order to allow low-delay panning, which leads to an increase in bit rate due to the lower compression efficiency of intra-coded frames. Finally, a structural delay is introduced, since repositioning the RoI to another region is only possible after decoding all frames within an intra-period. This structural delay can have a negative impact on very low-delay interactive applications such as viewing the video through virtual reality glasses.

As a solution to the above disadvantages, a monolithic streaming method has been investigated by Quang Minh Khiem et al. [8]. In this method, only H.264/AVC macroblocks that belong to the RoI are sent to the viewer, together with the blocks in previous frames that are used as a reference for the sent macroblocks. This results in more bandwidth-efficient streaming compared to the tile-based method. However, a random access coding structure is still used, meaning that the structural delay and the bit rate overhead of intra-frames remain.

3. HIGH EFFICIENCY VIDEO CODING

HEVC is the newest video compression standard and is the successor of the H.264/AVC standard [11]. Its main improvement is its increased compression efficiency (up to 50% bit rate reduction for the same perceptual video quality as H.264/AVC). This is achieved by dividing the frame into Coding Tree Units (CTUs) of typically 64×64 pixel blocks. These CTUs can be recursively split into smaller Coding Units (CUs) according to a quadtree structure. The smallest CU size that is allowed is 8×8 pixels. Each CU becomes the decision making point for the prediction mode (intra or inter) and can be partitioned further into eight possible Prediction Units (PUs), which are the basic units for intra- and inter-prediction.

For inter-prediction, there are two types of motion vector prediction modes, namely Advanced Motion Vector Prediction (AMVP) and merge mode. Both techniques use motion vectors from the neighboring PUs to determine a good match for the current PU. AMVP uses these motion vectors as predictors to determine a motion vector difference with the actual motion vector. For merge, the motion vector is copied from its (spatial or temporal) neighbors. This merge concept can be used in combination with a skip mode. If a skip mode is used, it implies that merge mode is used, CUs only contain one PU, and no residual data is present in the bitstream. This is well suited to encode static regions where the prediction error tends to be very small.

Since the above tools contribute to the complexity of an HEVC encoder, accelerating these tools will reduce the computational complexity of the encoding process. Consequently, the proposed personalized-view method will focus on limiting the decisions of these tools.

4. PROPOSED METHOD

Due to many disadvantages of the tile-based method, a personalized-view technique has first been proposed by Van Kets et al. [13]. This technique differs from the tile-based method by encoding the selected RoI of each user on the

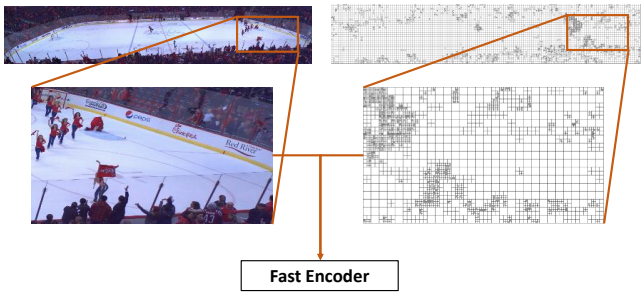

Figure 3: Accelerated personalized-view encoding. A part of the coding information (on the right) of the full ultra-high-resolution video is fed to the encoder together with the cropped view in order to accelerate the encoding of this view.

fly. Consequently, no extra pixel overhead is sent to the user, as illustrated in Figure 2. Another advantage of the personalized-view approach is that the user can use a standard decoder and has very flexible digital pan/tilt/zoom possibilities. A disadvantage is that the method is not scalable to a large number of users. In order to alleviate this problem, the encoding complexity of the individual encodes was lowered by reusing coding information of an encoded full ultra-high-resolution video in order to speed up the encoding process of the RoI of each user. However, only CUs were reused from the ultra-high-resolution video, resulting in limited acceleration. As a result, the remaining complexity of each individual encode is still significant.

In order to further accelerate the individual encodes in the personalized-view approach, we propose to extract prediction mode, PU information, motion vectors, and merge information from the full encode of the ultra-high-resolution video and use this information in addition to CU information to speed up the encoding of the individual personalized views. This is illustrated in Figure 3, where coding information is extracted from the same location in the ultra-high-resolution video as the personalized view. Both the personalized view and this coding information are then fed to a fast encoder which has to make less encoding decisions.

By feeding the encoder with more coding information such as PU information and motion vectors, more coding steps can be skipped. Consequently, copying more coding information of the panoramic video lowers the coding complexity of the personalized views and thus speeds up the encoding process. However, this will also lead to less optimal coding decisions since the cropped view lacks the surrounding pixels, which impacts intra- and inter-prediction at the borders of the view.

5. RESULTS

In this section, the content used for evaluation is described. Next, the performance of the accelerated personalized-view approach is evaluated. Finally, an extensive analysis is performed to investigate how the personalized-view method compares to the traditional tile-based approach

5.1 Used Content

In order to evaluate the methods to allow interactive video, panoramic content was chosen due to its current availability compared to high-resolution 360-degree video. The tested

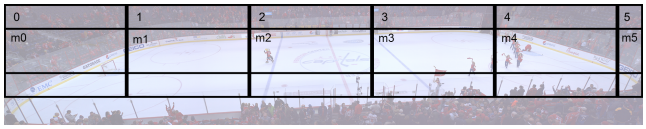

Figure 4: Selected 1088p RoIs for *hockey1_1*. The RoIs are marked with their corresponding notation on the figure. The middle views are indicated by their prefix *m*.

content consists of a hockey sports game, because this type of content has static areas such as the ice hockey field, moving areas such as the audience, and fast moving parts such as the hockey players. It is important to have a large range of spatial and temporal variability in the scenes, because this influences the complexity of the encoding. The hockey content consists of five sequences, split in two scenes. Only the first sequence of each scene was used for the eventual comparison, called *hockey1_1* and *hockey2_1*. *Hockey1_1* contains a scene where cheerleaders enter the field, whereas *hockey2_1* is a scene during the match itself. The sequences have a duration of 10s each, at a rate of 60 frames per second. They have a resolution of 10.000×1880 pixels and have been 4:2:0 chroma subsampled.

From the two sequences, static RoIs were chosen that contain different types of spatial and temporal activity. The RoIs are regions that many users will choose to watch, such as the ice hockey field itself. These selected RoIs are shown in Figure 4. The top and middle views are indicated by their corresponding view numbers as shown in the figure. The middle views, which mostly show the ice hockey field, are specified by their prefix *m*. The views with view number five (*5* and *m5*) were ignored because these do not have the correct RoI resolution. The RoIs each have a resolution of 1920×1088 pixels (1088p). The reason for the small deviation from 1080p is to have a multiple of the CTU size in HEVC. Note that only static views without zooming are considered in this paper.

In order to have a better indication on how much spatial and temporal information each view contains, the spatial perceptual information (SI) and the temporal perceptual information (TI) measure is calculated as described in the ITU-T Recommendation P.910 [4]. As seen in Figure 5, the RoIs have a large variety of TI/SI values, which corresponds with the assumption that views with different types of motion and spatial details are considered.

5.2 Personalized-View Approach

For the personalized-view method both the entire panoramic video and the RoIs were encoded. All encodings were done with version 16.5 of the HEVC Test Model (HM) [7]. Both the full panoramic video and all cropped views were encoded with four different quantization parameter (QP) values: 22, 27, 32 and 37.

All the views for both the non-accelerated reference encode and accelerated encodings were encoded with a *low-delay* configuration, meaning that the sequence is encoded with an I-frame, followed by all P-frames. This configuration was chosen since, contrary to the tiled-based approach which requires many random access points, each user has a personalized view and encoder instance. Therefore, the cropped region of the raw panoramic video (views) can be fed to one

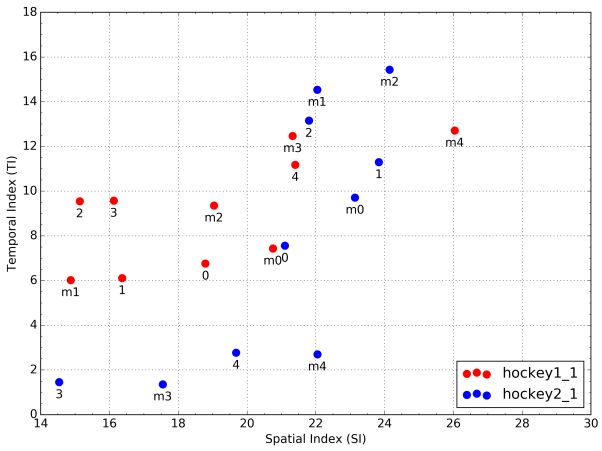

Figure 5: Spatial and temporal information for each view in sequence *hockey1_1* and *hockey2_1*. The notes beneath the markers specify the particular view. The selected RoIs cover a variety of spatial and temporal activity.

encoder instance for each user. It does not need I-frame refreshes because the user continues to use the same personalized stream. This configuration results in a lower delay, which is an important requirement for interacting with personalized views. For the tile-based method, a *random access* configuration would be needed because all the tiles can be retrieved by all users at any time and any position that corresponds with their selected RoI.

In order to evaluate the performance of the proposed personalized view method, bit rate overhead and encoder speed-up were evaluated. The bit rate overhead was evaluated using the Bjøntegaard Delta (BD) rate [2]. This metric shows the average bit rate increase for the same quality (measured as Peak Signal-to-Noise Ratio (PSNR)) of an accelerated encoder compared to a non-accelerated reference. In order to determine the complexity reduction, the time saving (TS) metric was calculated. This metric is determined by comparing the encoding time of the fast encoder (T_{fast}) to the encoding time of a non-accelerated encoder (T_{ref}) and is defined as (1).

$$\mathrm{TS}(\%) = \frac{\mathrm{T_{ref}} - \mathrm{T_{fast}}}{\mathrm{T_{ref}}} \quad (1)$$

Table 1 shows that by reusing only CU information the BD-rate is between 4.9% and 7.4% for the selected views with a time saving of around 79%. This time saving is comparable to the complexity reduction reported in the state-of-the-art [13]. However, using CU, mode, PU and motion vectors increases this complexity reduction to more than 96.5%. In this case, the BD-rate is between 8.3% and 19.5%. How these values compare to a tile-based approach is further investigated in the next section. Finally, copying merge (with skip) information results in irregular behavior for all views except for view *0*, since the blocks encoded with skip-mode attempt to copy the pixel-data of blocks that do not exist in the cropped views. This effect propagates further during the sequence and results in afterimages as illustrated in Figure 6. As a result, despite offering higher complexity reduction beyond 99%, merge information should not be copied.

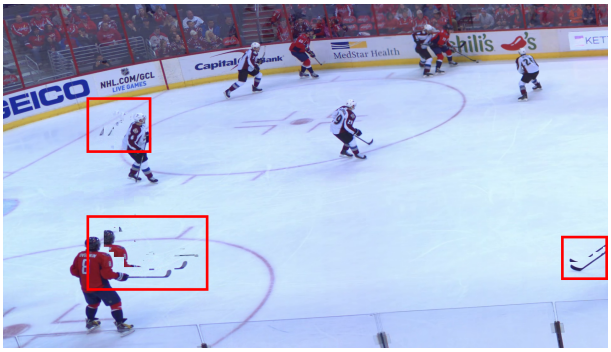

Figure 6: Illustration of afterimage-effect caused by copying unavailable blocks in the cropped views with merge/skip mode. Affected regions where afterimages are seen are marked with red rectangles.

5.3 Comparison

The second main contribution of this paper is to compare the tile-based method and the personalized-view method in terms of bit rate and PSNR for particular views. The bit rate should be low in order to make the interactive video system usable for clients with a limited bandwidth capacity. However, the PSNR should be high to have a good video quality for the RoI. In order to provide a fair comparison, several tile sizes of the tile-based approach were selected.

For the tile-based approach, the panoramic sequences were split into different tile sizes. The choice was made to pick 16:9 tile resolutions, because the RoIs are also close to 16:9 and this is a common aspect-ratio. The tested tile sizes were 1280×720, 1024×576, 640×360, 256×144 and 128×72 pixels. Next, these tiles were compressed using HM 16.5 as was done for the personalized-view method.

The tiles need to provide random access in order to allow changing of the RoI at any time as the tiles are pre-encoded on the server. Therefore, a *random access* configuration was chosen which consists of a structure of I-frames followed by B-frames that repeats every intra-period. This intra-period was selected as 32 frames, because this corresponds to a delay of 0.5s. Note that this is still a considerable amount of maximum delay when other tiles are selected, e.g. when another RoI is chosen. However, a smaller intra-period would result in higher bit rates due to worse compression of I-frames compared to B-frames. All tiles were encoded with the same four different QP values as in the previous section (22, 27, 32 and 37).

In the following subsections, the tile-based method and the personalized-view method are subsequently compared in terms of bit rate and PSNR.

5.3.1 Bit Rate Comparison

For the personalized-view method, the bit rates are retrieved from the encoding step with different coding information supplied to the encoder. For the tile-based method, the bit rates are calculated as the sum of the tiles of one particular tile size that (partially) overlap with the corresponding view. Figure 7 shows the bit rates of both methods and of the non-accelerated personalized-view encode for *hockey1_1* view *m4*. This view represents the plain white ice hockey field, the cheerleaders and the audience.

When both methods are compared, the bit rates of the

Table 1: BD-rates and Time Savings obtained by supplying different coding information. The columns incrementally represent the type of coding information that is reused from the panoramic video.

		BD-rate (%)					Time saving (%)				
Sequence	View	CU	+ mode	+ PU	+ motion	+ merge	CU	+ mode	+ PU	+ motion	+ merge
hockey1_1	0	7.4	8.4	15.6	19.5	22.1	78.3	79.5	92.9	96.5	99.2
	m1	7.2	8.9	16.5	17.7	1164.1	80.7	81.0	93.1	97.3	99.5
	m4	6.1	8.0	15.0	16.2	278.6	78.1	79.3	92.6	96.6	99.2
hockey2_1	m1	5.5	7.3	13.2	13.7	862.9	79.2	79.7	92.6	96.8	99.3
	m3	4.9	6.0	9.5	8.3	-98.9	81.2	81.8	93.4	97.5	99.6

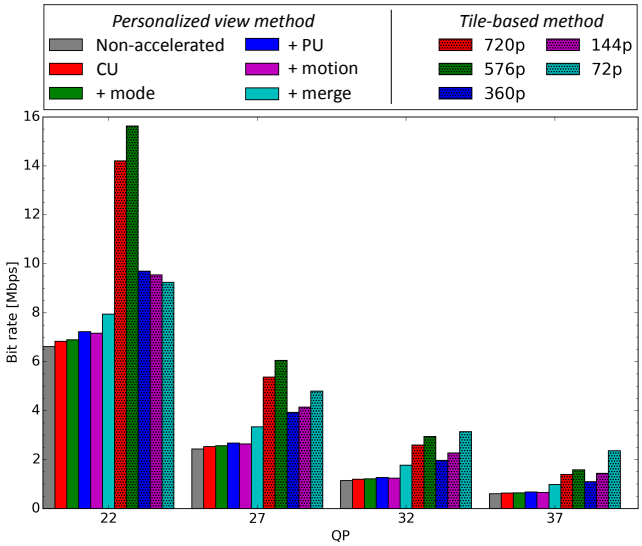

Figure 7: Comparison between the tile-based method with different tile sizes and the personalized-view method in terms of bit rate for *hockey1_1* view *m4* with the personalized-view method using a *low-delay* configuration.

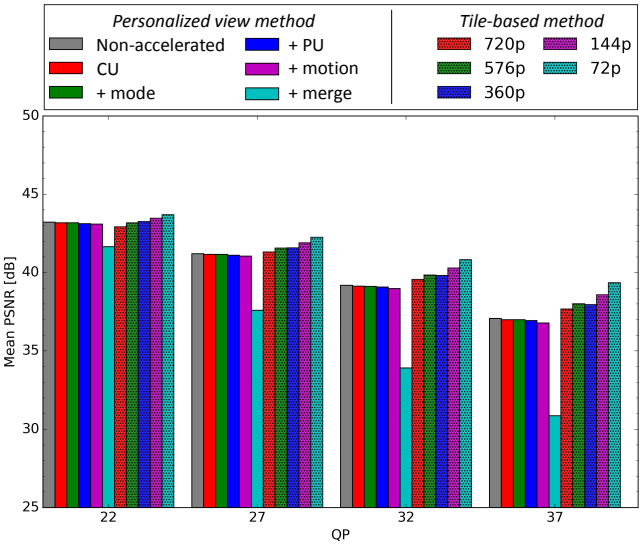

Figure 8: Comparison between the tile-based method and the personalized-view method in terms of PSNR for *hockey1_1* view *m4*.

personalized-view method are lower than the bit rates of the tile-based method. For example, in Figure 7, a bit rate of 6.83 Mbps is seen for QP 22 for which the cropped CU coding information of the panoramic video is reused, whereas 144p tiles have a bit rate of around 9 Mbps for QP 22. As seen for each QP, the bit rate of the personalized-view method remains below the bit rate of the tile-based method, even despite the overhead caused by accelerating the encoding of the personalized views. The reason for this difference is likely due to overhead caused by encoding a video with separate tiles. Because of the separate tiles, decisions such as motion estimation are constrained to the tile itself. As a result, if an object leaves the tile, it will be encoded less efficiently. Moreover, the tile boundaries will usually not match the boundaries of the selected view. As a result, extra pixels outside the view are also encoded, which further increases the bit rate overhead.

In the above comparisons only static views are considered, but it is expected that the personalized-view method will further outperform the tile-based method in terms of bit rate when panning and tilting are taken into consideration. In such a scenario, all tiles covered by a dynamic view during the same intra-period need to be retrieved. If the user completely changes the RoI within this intra-period, the bit rate will at least double during this period since both the tiles corresponding to the old and new RoI will have been transmitted.

5.3.2 PSNR Comparison

Another important aspect to compare with is quality, which is measured in PSNR. The mean PSNR for the tile-based method was calculated by first transforming all the PSNR values back to the Mean Squared Error (MSE). Then the average of all the MSE values, temporally and spatially corresponding to each tile size and to each QP that covers the view, was calculated and transformed back to PSNR. Since $\text{PSNR} \propto 10\log_{10}(\text{MSE})$, averaging MSEs instead of PSNR tends to penalize more if a single tile has a low PSNR. Consequently, minimizing such an average tends to enforce a more constant PSNR over the different tiles.

Figure 8 presents the PSNR of both methods for *hockey1_1* view *m4*. This figure shows that the tile-based method for all the tile sizes performs better in terms of PSNR than the personalized-view method. For QP 32, the PSNR is around 39 dB when the cropped CU, mode and PU coding information of the panoramic video is reused, whereas the 144p tiles have a PSNR of around 40 dB for QP 32. Similar behavior is seen for the other views. Note that for the personalized-view

Figure 9: Illustration of tiling artefacts at QP 37. The tile borders are indicated by the white ticks.

method the PSNR drops significantly when merge coding information is also reused from the panoramic video.

Although the PSNR of the tile-based method appears to be higher, inter-tile artefacts are visible at higher QP-values for the tile-based method as a blocking effect at the border of each tile (illustrated in Figure 9). These are not taken into account with the PSNR metric, despite lowering the subjective visual quality.

6. CONCLUSION

In this paper, a fast personalized-view method for delivery of interactive views from immersive video content was proposed. This method was compared to the tile-based method. It was shown that reusing coding information obtained from a panoramic video to fast encode each personalized view performs better in terms of bit rate compared to the tile-based method.

As future work, the proposed method will also be extended for dynamic views. We anticipate that in this case the difference in bit rate between the personalized-view method and the tile-based method will become larger when the user pans or tilts the view. Additionally, a way to accelerate merge and skip decisions with a less negative impact on the video quality will be investigated. Finally, more HEVC coding information can still be exploited to further speed-up the encodings of the personalized-view method.

7. ACKNOWLEDGMENTS

The work in this paper was performed as part of the iMinds PRO-FLOW (Predictive deliveRy Orchestration For ultra-LOw-latency Web applications) project.

The research activities described in this paper were funded by the Data Science Lab (Ghent University - iMinds), Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union. The computational resources (STEVIN Supercomputer Infrastructure) and services used in this work were kindly provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation, and the Flemish Government department EWI.

The video content used in this paper was kindly provided by Kiswe.

8. REFERENCES

[1] P. R. Alface, J. F. Macq, and N. Verzijp. Interactive omnidirectional video delivery: A bandwidth-effective approach. *Bell Labs Technical Journal*, 16(4):135–147, March 2012.

[2] G. Bjøntegaard. Calculation of average PSNR differences between RD-curves. Technical Report VCEG-M33, ITU-T Video Coding Experts Group (VCEG), Apr. 2001.

[3] L. D'Acunto, J. van den Berg, E. Thomas, and O. Niamut. Using MPEG DASH SRD for Zoomable and Navigable Video. In *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, pages 34:1–34:4. ACM, 2016.

[4] ITU-T. Subjective video quality assessment methods for multimedia applications. Technical Report Rec. P.910, Apr. 2008.

[5] J. Le Feuvre and C. Concolato. Tiled-based Adaptive Streaming Using MPEG-DASH. In *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, pages 41:1–41:3. ACM, 2016.

[6] A. Mavlankar and B. Girod. Spatial-Random-Access-Enabled Video Coding for Interactive Virtual Pan/Tilt/Zoom Functionality. *IEEE Trans. Circuits Syst. Video Technol.*, 21(5):577–588, May 2011.

[7] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan. High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description. Technical Report JCTVC-S1002, ITU-T Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2014.

[8] N. Quang Minh Khiem, G. Ravindra, A. Carlier, and W. T. Ooi. Supporting Zoomable Video Streams with Dynamic Region-of-interest Cropping. In *Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems*, MMSys '10, pages 259–270, 2010.

[9] N. Quang Minh Khiem, G. Ravindra, and W. T. Ooi. Adaptive encoding of zoomable video streams based on user access pattern. *Signal Processing: Image Communication*, 27(4):360–377, 2012.

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

[11] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1649–1668, Dec 2012.

[12] Y. Umezaki and S. Goto. Image segmentation approach for realizing zoomable streaming HEVC video. In *Int. Conf. Inf. Commun. Signal Process. (ICICS)*, pages 1–4, Dec 2013.

[13] N. Van Kets, J. De Praeter, G. Van Wallendael, J. De Cock, and R. Van de Walle. Fast encoding for personalized views extracted from beyond high definition content. In *Proc. IEEE Int. Conf. Broadband Multimedia Syst. and Broadcast. (BMSB)*, pages 1–7, June 2015.