BLOCK STRUCTURE REUSE FOR MULTI-RATE HIGH EFFICIENCY VIDEO CODING

Damien Schroeder, Patrick Rehm and Eckehard Steinbach

Technische Universität München
Chair of Media Technology
Munich, Germany

ABSTRACT

Adaptive HTTP streaming requires a video to be encoded at multiple independently decodable bitrates. Encoding of multiple bitrates is a complex and time consuming process, especially with the new video coding standard HEVC. In this paper, we analyze the relation of HEVC block structures across encodings of a video at different bitrates. We propose a multi-rate encoding method for HEVC to decrease the overall encoding complexity while keeping the rate-distortion performance as high as possible. Experimental results show an average encoding time decrease of 27% compared to the reference HEVC encoder without degrading the rate-distortion performance.

***Index Terms*—** HEVC, DASH, block structure, multi-rate encoding

1. INTRODUCTION

Nowadays, video streaming is mostly implementing the adaptive HTTP streaming paradigm, recently standardized as Dynamic Adaptive Streaming over HTTP (DASH) [1]. The video content is made available at the server at different bitrates called representations, and the streaming clients can request the representation best suiting their requirements. Scalable codecs (e.g. [2]) have been proposed to encode a video and offering multiple bitrates. The rate-distortion (RD) performance of scalable codecs is, however, lower than with single layer coding. Transcoding can also be used to generate multiple bitrates from an encoded video. Again, the RD performance is lower than for a single layer coding due to requantization of the encoded video [3]. This paper focuses on the simultaneous encoding of multiple independently decodable representations of a single video, as represented in Figure 1.

The new video coding standard High Efficiency Video Coding (HEVC) [4] offers improved RD performance compared to previous standards at the cost of increased encoding complexity. Since DASH requires a video to be encoded in multiple representations, the encoding with HEVC can become a performance bottleneck, especially in the case of live

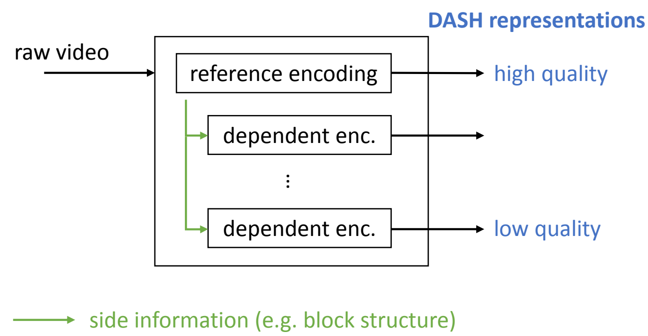

Fig. 1. Proposed multi-rate HEVC coding system.

DASH streaming, and is thus generally performed on powerful servers. Accordingly, research on adaptive HTTP streaming has almost exclusively considered the downlink channel of communication, especially in mobile networks (e.g., [5, 6]). In order to enable a mobile uplink DASH communication with resource-constrained terminals, the complexity of the encoding of multiple representations has to be significantly reduced.

Encoding the same video multiple times, even at different qualities, implies a certain degree of redundancy. This redundancy can be used to reduce the complexity of encoding multiple representations. In 2002, in the context of offering a video sequence at different fixed bitrates, [3] proposed to perform the motion estimation only for one reference representation but didn't assess the encoding time gains resulting from this method. Later, [7] reuses the same idea and performs motion compensation only once for a reference stream in a VP8 encoder. The resulting RD performance is, however, severely degraded compared to the reference encoder.

The contribution of this paper is twofold: first, we analyze the specificities of the block structure of HEVC in the case of multiple encodings at different SNR qualities, that is, when the video signal is more or less quantized. Second, based on the findings, we propose a method to reuse the block structure from a high quality reference representation to speed up the dependent encoding of lower quality representations. We implement and assess the proposed method based on the ref-

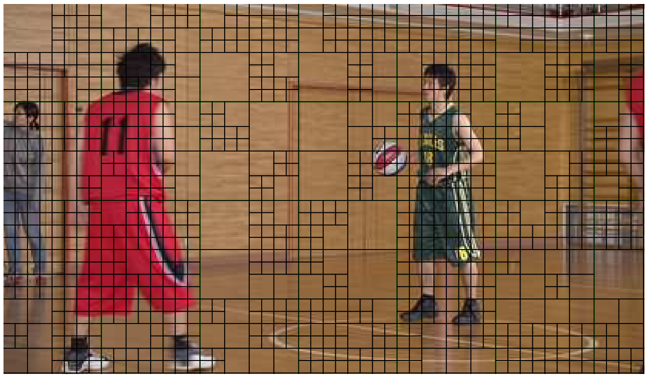

Fig. 2. First frame of *BasketballPass* encoded at QP 22 with resulting block structure.

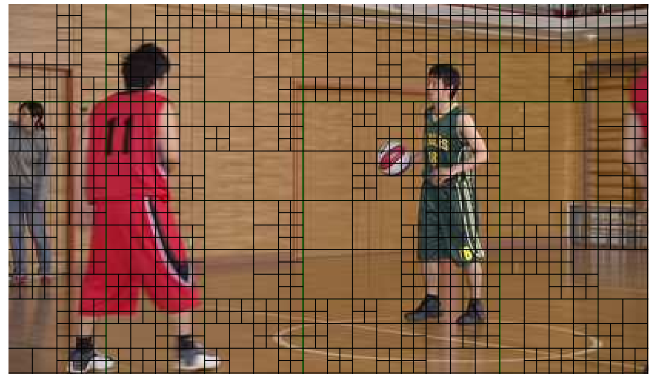

Fig. 3. First frame of *BasketballPass* encoded at QP 26 with resulting block structure.

erence HEVC software and show that we can significantly reduce the encoding time for multiple HEVC representations without notably degrading the RD performance. Reducing the complexity either allows the encoding of more representations for a given computing power or allows devices with limited computing resources, such as mobile devices, to provide DASH content, opening the way to mobile uplink DASH communication.

The rest of the paper is organized as follows. Section 2 describes the HEVC block structure behavior across multiple representations at different qualities. The proposed multi-rate HEVC encoder is presented in Section 3. Section 4 shows the encoding results compared with the original HEVC encoder and Section 5 concludes the paper.

2. BLOCK STRUCTURE AT MULTIPLE RATES

2.1. HEVC block structure

The block structure based on a quadtree representation is considered to be one of the most significant changes in HEVC compared to prior video coding standards [8]. A video slice is first divided in multiple coding tree units (CTU) which comprise a quadtree structure whose leafs are called coding units (CU). The decision of the prediction mode (e.g., inter/intra) is made at the CU level. The size of a CU is given by its depth in the quadtree with depth 0 corresponding to the biggest block size and a greater depth corresponding to a smaller block size. Determining the block structure is a major part of the rate-distortion-optimization (RDO), which is the most time consuming part of HEVC [9]. The RDO could be shortened if information about the block structure were available beforehand.

2.2. Similarities across encodings

In the case of multi-rate HEVC video encoding, we observe that there are strong similarities in the block structure across multiple encodings of a single video sequence at different SNR qualities. As an example, Figure 2 shows the first frame of the *BasketballPass* sequence encoded with the HEVC reference software, HM version 16.2 [10] at quantization parameter (QP) 22 with the resulting quadtree block structure. Figure 3 shows the same frame encoded at QP 26 with the resulting block structure. Similarities in the block sizes can be observed between the two figures.

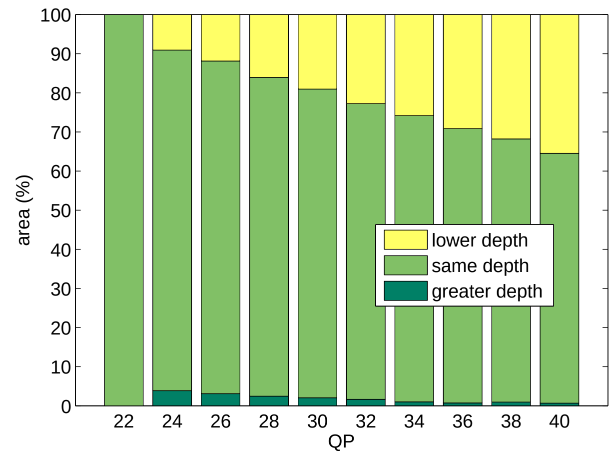

Fig. 4. Percentage of the area of the *BasketballPass* sequence with block depth greater, identical or lower than the reference encoding at QP 22.

In order to quantify the similarity between two encodings with different QP parameters, we calculate the percentage of the area of the frames where the block size is the same, that is, the block has the same depth. If the block depth is not identical, it can either have a greater depth (i.e. a smaller block size) or a lower depth (i.e. a larger block size).

Figure 4 shows the percentage of the area of the frames where the block depth is greater, identical, or lower than the reference depth given by the encoding at QP 22. For that, one second (50 frames) of the *BasketballPass* sequence is en-

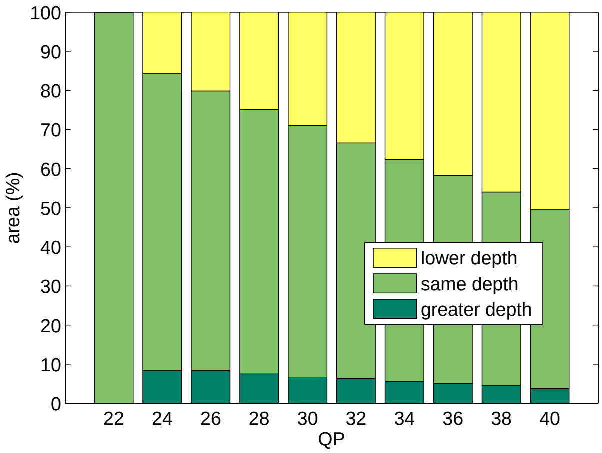

Fig. 5. Average percentage of the area of 8 sequences with block depth greater, identical or lower than the reference encoding at QP 22.

coded with QPs ranging from 22 to 40. The encoding at QP 22 shows 100% similarity with the reference encoding QP 22, as expected. The more the QP differs from the reference QP, the more the block structure differs from the one at the reference QP.

Interestingly, we observe that with growing QP, the video tends to have more blocks with a lower depth, that is, with a larger block size. On the other hand, less blocks will have a greater depth. This behavior can be intuitively explained by the fact that at a greater QP, the strong quantization leads to less details in the image and thus prediction can be more easily performed at a larger block size.

This behavior is also observed in general for other encoded videos. In Figure 5, the percentage of the area with greater, identical or lower depth is presented as a mean over one second of 8 video sequences (*BasketballPass, BlowingBubbles, BQmall, Kimono, ParkScene, PartyScene, PeopleOnStreet, RaceHorses*) defined in [11].

3. PROPOSED MULTI-RATE ENCODER

Given the fact that there are similarities in the block structure across multiple encodings of a single video at different qualities, we propose to reuse the information of the block structure of a reference encoding to shorten the RDO process of other dependent encodings, and thus reduce the encoding time.

The RDO process in HM 16.2 is implemented recursively starting from the largest CU size, that is, depth 0. Depending on the encoder settings, various predictions (inter/intra) are examined and best candidates in the RD sense are chosen. Then, the block is divided into 4 subblocks (depth + 1) and the RDO is recursively applied to these subblocks, and so on. The block structure combined with the predictions which

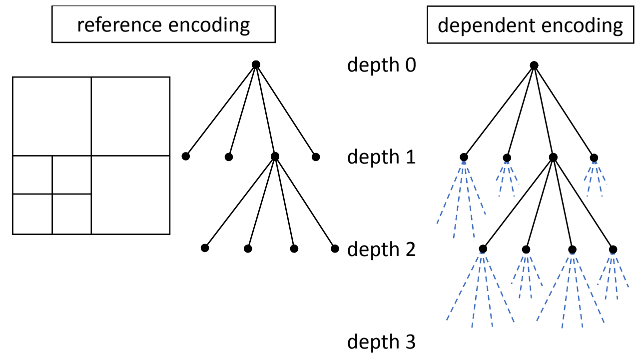

Fig. 6. Example CTU block structure and quadtree for the reference encoding and quadtree checked during the RDO process for the dependent encoding.

gives the smallest RD cost is chosen in order to maximize the RD performance. This recursive block subdivision is equivalent to a depth-first traversing of the quadtree [9].

Knowing that, on one hand, the blocks of an encoding will mostly have a lower or equal depth compared to a high quality reference encoding, and that the amount of blocks having a greater depth than in the reference encoding is relatively small (cf. Figure 5), and on the other hand, the RDO process of the HEVC encoder is implemented recursively, we propose to stop the RDO process of the dependent encodings at the depth given by the reference encoding for each block in the video sequence. This is illustrated in Figure 6, where an example block structure of the reference encoding and the corresponding quadtree is shown on the left and on the right the quadtree that is checked during RDO of a dependent encoding. The depths depicted in dashed lines are not checked during the dependent RDO process, which leads to significant encoding time savings. In fact, the number of blocks to check is multiplied by 4 from one depth to the next greater depth.

In the case of blocks that should have a greater depth as in the reference encoding, a suboptimal block size in the RD sense will be chosen. The overall RD loss should be small, however, as this concerns only a relatively small percentage of all blocks (cf. Figure 5). On the other hand, the relatively large number of blocks which have a lower depth in a dependent encoding will still have the optimal block size in the RD sense as the RDO process will pass through these low depths during the dependent encoding.

The fact that the highest quality encoding tends to have the most small blocks combined with the recursive RDO process implementation is an argument to chose the highest quality encoding as the reference encoding, if we want to achieve best RD performance. Conceptually, a low quality reference could be used in addition to an RDO process where first the smallest blocks would be processed and then going up the quadtree to larger blocks. Due to lack of space, we do not consider this option in this paper.

4. RESULTS

4.1. Settings

To evaluate the proposed method, we compare our implementation with the unmodified HM 16.2 encoder [10]. We consider both the RD performance as well as the encoding time as comparison metrics. The RD-performance difference is measured using the Bjontegaard delta rate (BD-rate) [12] and the Bjontegaard delta PSNR (BD-PSNR) [13]. The encoding time difference (ΔT) is measured as the difference of the total encoding time for 5 representations including the reference at different qualities (fixed QP 22, 26, 30, 34 and 38).

All encodings are performed on an Ubuntu server with a Xeon X5460 CPU at 3.16 GHz and 32 GB RAM. 8 videos defined in [11] with resolutions from 416×240 to 2560×1600 pixels and frame rates between 24 fps and 60 fps are encoded. All sequences are encoded with the "random access, main" profile defined in [11], a CTU size fixed to 64×64 pixels and a maximum CU depth of 4.

4.2. Encoding results

Table 1 shows the encoding performance of the proposed encoding method compared with the original HM 16.2 reference. On average, the encoding time for 5 representations is reduced by 27% compared to the original HM encoder. At the same time, the BD-rate increase is only 0.56% and the BD-PSNR decrease is only 0.02 dB on average, which can be considered acceptable. We observed similar results when applying rate-control instead of a fixed QP encoding (overall encoding time decreased by 27.34%, BD-rate increased by 0.49% and BD-PSNR decreased by 0.02 dB).

Table 1. Comparison of encoding results

Sequence	BD-rate	BD-PSNR	ΔT
PeopleStreet (2560×1600)	0.85%	-0.04 dB	-19.02%
Kimono (1920×1080)	0.75%	-0.02 dB	-43.10%
ParkScene (1920×1080)	0.43%	-0.01 dB	-35.79%
BQMall (832×480)	0.57%	-0.02 dB	-28.60%
PartyScene (832×480)	0.26%	-0.01 dB	-22.10%
BasketballPass (416×240)	0.62%	-0.03 dB	-35.12%
BlowingBubbles (416×240)	0.43%	-0.02 dB	-20.03%
RaceHorses (416×240)	0.55%	-0.03 dB	-12.67%
Average	**0.56%**	**-0.02 dB**	**-27.05%**

Figure 7 shows an example of PSNR-bitrate curves for the *BasketballPass* video encoded both with the original HM 16.2 encoder and the proposed encoder which reuses the CTU structure across representations. As indicated in Table 1, the difference between the curves is negligible. Figure 8 shows the encoding time for each of the 5 representations. The encoding time for the reference representation (here at QP 22) is approximately the same as for the original HM encoder and can be slightly longer due to the additional output of the block structure information for the other representations. The encoding time for the lower quality dependent representations (QP 26, 30, 34 and 38) is then reduced compared to the original encoder by reusing the block structure information. The encoding time for these four representations is reduced by 46% in the case of the represented *BasketballPass* sequence.

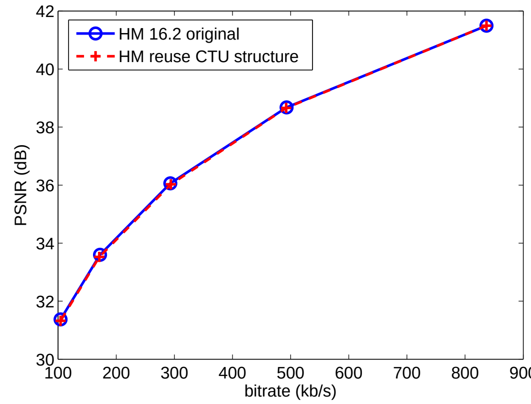

Fig. 7. RD curves of the *BasketballPass* sequence for the original HM 16.2 encoder and the proposed encoder.

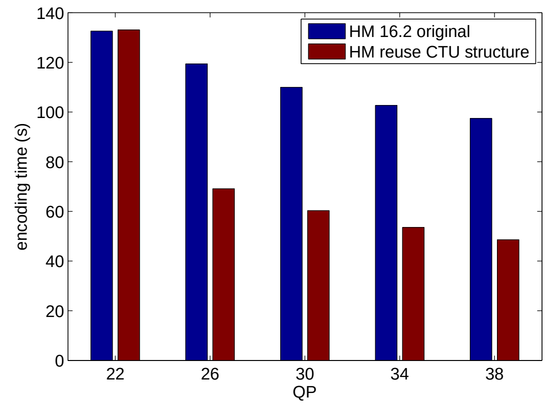

Fig. 8. Encoding time of the *BasketballPass* sequence for the original HM 16.2 encoder and the proposed encoder.

5. CONCLUSION

In this paper, we analyzed the similarities between the block structures of a single video encoded at different qualities with HEVC. Based on the observations, we proposed to reuse the block structure information from a high quality reference encoding in order to shorten the RDO process and reduce the dependent encoding time for lower quality representations. Our experimental results show that the total encoding time of 5 representations can be reduced by approximately 30% on average without any significant loss in RD performance compared to the reference HEVC encoder.

6. REFERENCES

[1] ISO/IEC 23009-1, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media Presentation description and segment formats," Tech. Rep., Apr. 2012.

[2] Philipp Helle, Haricharan Lakshman, Mischa Siekmann, Jan Stegemann, Tobias Hinz, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "A Scalable Video Coding Extension of HEVC," in *Data Compression Conference (DCC)*, Snowbird, UT, USA, Mar. 2013.

[3] André Zaccarin and Boon-Lock Yeo, "Multi-rate encoding of a video sequence in the DCT domain," in *IEEE International Symposium on Circuits and Systems, ISCAS 2002*, May 2002.

[4] Gary J Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[5] Ozgur Oyman and Sarabjot Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, Apr. 2012.

[6] Ali El Essaili, Damien Schroeder, Dirk Staehle, Mohammed Shehada, Wolfgang Kellerer, and Eckehard Steinbach, "Quality-of-experience driven adaptive HTTP media delivery," in *IEEE International Conference on Communications (ICC)*, Budapest, Hungary, June 2013.

[7] Dag Haavi Finstad, Håkon Kvale Stensland, Håvard Espeland, and Pål Halvorsen, "Improved multi-rate video encoding," in *IEEE International Symposium on Multimedia (ISM)*, Dana Point, CA, USA, Dec. 2011.

[8] Il-Koo Kim, Junghye Min, Tammy Lee, Woo-Jin Han, and JeongHoon Park, "Block partitioning structure in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1697–1706, Dec. 2012.

[9] Frank Bossen, Benjamin Bross, Karsten Suhring, and David Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[10] HEVC reference software HM 16.2, "https://hevc.hhi.fraunhofer.de/svn/svn_hevcsoftware /tags/hm-16.2/," 2015.

[11] Frank Bossen, "Common test conditions and software reference configurations," Tech. Rep. I1100, JCT-VC, 2012.

[12] Gisle Bjontegaard, "Improvements of the BD-PSNR model," *ITU-T SG16 Q*, vol. 6, pp. 35, 2008.

[13] Gisle Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA*, Apr. 2001.