Energy-Aware HEVC Transrating based on Frame Partitioning Inheritance

Thiago Bubolz, Ruhan Conceição, Bruno Zatt, Luciano Agostini, Guilherme Correa
Video Technology Research Group (ViTech)
Federal University of Pelotas (UFPel)
Pelotas, Brazil
{tlabubolz, radconceicao, zatt, agostini, gcorrea}@inf.ufpel.edu.br

Mateus Grellert
Graduate Program in Electrical and Computing Engineering
Catholic University of Pelotas
Pelotas, Brazil
mateus.grellert@ucpel.edu.br

***Abstract*— Video transcoding for bitrate adaptation, commonly referred as transrating, has become an indispensable operation to enable the transmission of videos at different quality levels in streaming services. Transrating is usually performed by a decoding and an encoding operation in sequence, but to obtain various bitrates and quality levels, the process must be repeated several times, making it very costly in terms of energy consumption, especially for streaming service providers. This work proposes a solution to reduce the energy consumption for video transrating with the High Efficiency Video Coding (HEVC) standard. The solution allows the transcoder to inherit the frame partitioning structures from the first bitstream to the second bitstream, avoiding the test of different configurations when reencoding. When compared to the original transrating solution, the proposed strategy reduces energy consumption by 50.4%, with a negligible loss of 0.835% in encoding efficiency.**

Keywords— HEVC, video coding, transrating, transcoding, energy efficiency.

I. Introduction

High-Efficiency Video Coding (HEVC) [1] represents the state-of-the-art regarding video coding standards. The main goals of HEVC before its standardization were doubling the H.264/AVC [2] compression capabilities while maintaining image quality and encoding complexity [3]. However, the complexity constraint was not achieved and HEVC requires 500% more computations than its predecessor to encode a given video sequence [4]. The high-complexity HEVC encoder hampers video coding in real time and leads to large energy consumption. In [5], the authors show that the HEVC encoder requires an energy consumption 17% higher than its predecessor. This is especially critical in multimedia-capable embedded devices, such as smartphones and tablets, which usually depend on limited battery resources. Thus, efficient solutions that reduce HEVC encoder complexity and energy consumption with negligible losses in compression efficiency are highly required.

According to [6], the use of streaming and downloading video content services will surpass 80% of the internet traffic by the year 2021. With the aim of allowing compatibility among devices, services and applications that transmit/receive digital videos on the internet, there is a need to convert encoded videos to different standards (heterogeneous transcoding) or only change their characteristics (homogeneous transcoding). Due to the increasing use of video streaming services such as YouTube and Netflix, transcoding for bitrate adaptation, also called as transrating, has become an essential task, since it is necessary to maintain several versions of the same video with different bit rates on the server side. As the transrating operation requires long processing times, it is usually performed offline and the several bitstream versions are stored in the servers for future requests. Additionally, rarely-accessed videos can also be reencoded on-the-fly, i.e., as they are requested by the viewers.

The typical implementation of a transcoder, usually referred to as tandem transcoder, first decodes a bitstream originally encoded under certain parameters and constraints, generating a video output. This video is then used as the input to an encoder in sequence, configured differently. Thus, as the HEVC transrating process is composed of an HEVC decoding and an HEVC encoding in sequence, an energy consumption even larger than the observed at the HEVC encoder is required.

Even though streaming servers do not operate on battery resources, video transrating needs to be performed several times for each video, which demands both computational and energy resources. With this in view, some authors have been proposing strategies to reduce energy consumption at the HEVC encoder. In [7], the authors present a hardware-software collaborative energy reduction scheme for the HEVC intra-frame encoder. However, it does not achieve significant levels of energy reduction and the work does not indicate results for compression efficiency loss, which is essential for services that value the user quality of experience. In [8], a recursive search reduction algorithm is proposed for HEVC partitioning structure decisions, which achieves an energy reduction of 60% with a compression efficiency loss of 3.4%. In [9], a Pareto-based energy control strategy is proposed for HEVC and achieves a 25.2% reduction in energy consumption, with a loss of 5% in compression efficiency. Besides presenting non-negligible losses in compression efficiency, none of these works are focused on solutions for video transcoding or transrating.

This work proposes an energy-aware transrating method for the HEVC standard. Differently from the related works, it focuses on the transcoder architecture to reduce energy consumption at the encoder side. To accomplish that, the HEVC transrating process was modified to inherit frame partitioning information from the original bitstream and simplify the reencoding. A threshold inherited from the decoding step was defined to limit the search for best partitioning during the transrating, which allowed the algorithm to maintain the image quality to acceptable levels even when reducing significantly the transrating operations.

This paper is organized as follows. Section II presents a background overview. Section III presents a statistic evaluation on frame partitioning, and Section IV presents the

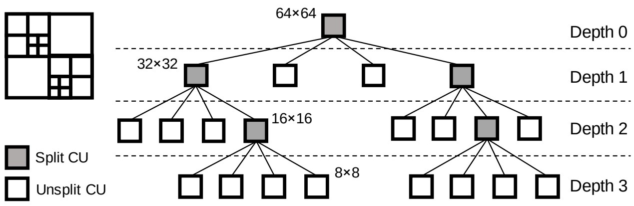

Fig.1: Example of a CTU partitioned into several CUs following the HEVC quadtree structure.

proposed transrating scheme. Section V presents the obtained results. Finally, section VI concludes the paper.

II. HEVC PARTITIONING STRUCTURES

HEVC introduced a much more flexible frame partitioning scheme in comparison to its predecessors, with the goal of enabling better encoding efficiency for various types of content and resolution. Initially, each frame is partitioned into equally-sized square Coding Tree Units (CTUs), which size is typically 64×64. Then, each CTU can be partitioned into multiple Coding Units (CUs), from just one representing the whole 64×64 region, up to many 32×32, 16×16 and 8×8 CUs. The partitioning is performed in a quadtree-like recursive process, in which each CU is split into four other CUs, until the minimum CU size (8×8) is reached.

Figure 1 shows an example of a CTU that was partitioned into multiple CUs. The best partitioning for a CTU is chosen after trying all possibilities, computing the Rate-Distortion (RD) cost for each case, and comparing them all. The partitioning that returns the minimum RD cost is finally chosen. As HEVC allows up to four partitioning levels, the computational complexity involved in this decision process is extremely high.

III. STATISTICAL EVALUATION OF CU PARTITIONING

This section presents a statistical analysis on the correlation between CU sizes in a high bit rate (HBR) video and its transcoded versions with lower bit rates (LBR). The analysis provides the basis for the method proposed in this work.

The HEVC test Model (HM) reference software, version 16.5 [10], was used to collect information for this analysis. All the settings defined in the Common Test Conditions (CTC) document [11] were followed in the experiments, and the Random Access, Main HEVC encoder configuration was used. The videos used for this statistical evaluation also belong to the CTC specifications and differ from one another in terms of motion and texture characteristics, as well as in spatial resolution: *BlowingBubbles*, *RaceHorses*, *SlideEditing*, *KristenAndSara*, *BasketBallDrive*, and *Traffic*. All the sequences were first encoded with the HM software and QP 22, to guarantee an HBR bitstream with good image quality. Then, the transrating bit rates for each video were calculated as 80%, 60%, 40% and 20% of the bit rate obtained in the HBR encoding. Each video was then transcoded four times (once for each target LBR) and the CU size after the transrating process was saved to be compared with the partitioning in the HBR bitstream during the correlation analysis.

TABLE I: AVERAGE CORRELATION RESULTS FOR LBR=80%.

Original CU size	CU size after transrating to LBR=80%			
	64×64 (%)	**32×32 (%)**	**16×16 (%)**	**8×8 (%)**
64×64	**74.1**	22.72	2.88	0.29
32×32	23.92	**66.58**	8.62	0.87
16×16	11.24	34.18	**50.4**	4.16
8×8	3.66	16.38	25.49	**54.46**

TABLE II: AVERAGE CORRELATION RESULTS FOR LBR=60%.

Original CU size	CU size after transrating to LBR=60%			
	64×64 (%)	**32×32 (%)**	**16×16 (%)**	**8×8 (%)**
64×64	**77.48**	19.94	2.45	0.12
32×32	29.99	**61.76**	7.68	0.55
16×16	14.56	37.92	**44.52**	2.99
8×8	5.25	19.05	26.2	**49.49**

TABLE III: AVERAGE CORRELATION RESULTS FOR LBR=40%.

Original CU size	CU size after transrating to LBR=40%			
	64×64 (%)	**32×32 (%)**	**16×16 (%)**	**8×8 (%)**
64×64	**82.61**	15.58	1.73	0.067
32×32	40.7	**52.98**	5.99	0.34
16×16	24.23	39.69	**34.4**	1.66
8×8	9.26	24.12	22.97	**43.63**

TABLE IV: AVERAGE CORRELATION RESULTS FOR LBR=20%.

Original CU size	CU size after transrating to LBR=20%			
	64×64 (%)	**32×32 (%)**	**16×16 (%)**	**8×8 (%)**
64×64	**87**	12.18	1.06	0.05
32×32	53.41	**42.48**	3.99	0.1
16×16	38.49	39.51	**21.44**	0.54
8×8	17.63	27.32	17.11	**37.92**

Tables I-IV show average correlation results in percentages for the four transrating processes performed in this analysis. The rows in Tables I-IV represent each CU size chosen by the encoder during the original encoding process (i.e., the HBR bitstream, encoded with QP 22). The columns represent the CU sizes chosen during the transrating to the LBR cases (i.e., the 80%, 60%, 40% and 20% bit rates). For example, considering LBR=40% (Table III), 52.98% of the CUs encoded as 32×32 in the HBR bitstream were encoded in the same size (32×32) when transrating to LBR=40%, whereas 40.7% of them were encoded as larger CUs (64×64) and 6.33% were encoded as smaller CUs (5.99% as 16×16 CUs and only 0.34% as 8×8 CUs).

Notice that this behavior is recurrent for all CU sizes. In most cases, the same CU size used in the original encoding or a larger CU size is employed during the transrating for lower bit rates, but very rarely a smaller CU size is used. Specifically for 64×64 CUs, on average in 80.22% of the cases the same CU size is employed and in 19.78% of the cases a smaller CU is used. For 32×32 and 16×16 CUs, only in 7.04% and 2.34% of the cases smaller CUs are chosen, respectively.

Thus, the statistical analysis reveals that there is usually a small chance of a CU being encoded as smaller partitions when transrating from HBR bitstream to LBR bitstreams, especially when very small LBR values are used. This is expected because using smaller CUs requires including more side information to the bitstream, such as block headers,

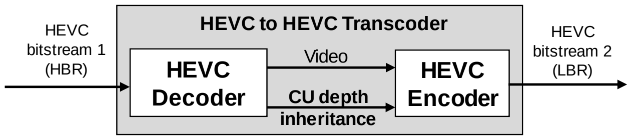

Fig. 2: CU splitting process using side information from decoder.

motion vectors, etc. This way, when transrating for reduced bit rates, larger (and less numerous) CUs are prioritized.

IV. Proposed Energy-Aware Transrating Scheme

The analysis presented in the previous section led to conclusions that guided the energy-aware transrating scheme presented in this section, aiming at reducing energy consumption during the HEVC reencoding process.

The proposed scheme directly employs the partitioning of CUs observed during the decoding process that composes the transrating to assist the CU decision process during the reencoding, as shown in Fig. 2. Notice in the figure that the CU depth is inherited from the decoding by the encoder. This is possible because when the transcoder starts to decode a frame from the HBR bitstream, the quadtree depth of each CU is stored to be used as side information during the reencoding of the same frame. This way, when the decoded frame is delivered to the encoder, a mapping of the CU depths for this corresponding frame is also delivered and the transcoder uses it to speed up the reencoding.

The adapted CU splitting algorithm executed at the encoder is presented in Fig. 3. The flowchart shows that the recursive search for the best CU size (*CU_encode* function) is halted if the current depth (depth variable) reaches an *HBR_CUDepth* threshold, which is the CU depth observed while decoding the HBR bitstream. Oppositely, if the current depth is below the *HBR_CUDepth* threshold, the CU splitting process is continued. For example, if a CU was encoded at depth 1 in the HBR bitstream, only depths 0 and 1 will be tested during the reencoding and then the process will be halted.

As the proposed method applies for transrating from HBR bitstreams to LBR bitstreams, the amount of erroneous halting decisions will be small, since lower bit rates will prioritize the use of larger CUs, as shown in Section III.

V. Experimental Results and Discussions

The experiments were conducted in the same conditions of the analysis presented in Section III, except for the video sequences that are different from those. All videos used in the experiments are listed in Table V, and they include classes A2, B, C, D, E from the CTC document [10]. To evaluate the proposed method in terms of encoding efficiency and energy consumption, a tandem transrating with no changes in the encoding algorithm was performed for all video sequences for comparison purposes. Similarly to the experiments in Section III, the HBR bitstreams were encoded with QP 22, and the LBR cases as 80%, 60%, 40%, and 20% of the HBR bitstream. The original tandem transcoder was used to perform the four transratings for each video sequence. Then, a modified version of the transcoder was implemented with the strategy presented in Section IV, and the same transratings were performed on it. Thus, the results presented in the following paragraphs are comparisons between the proposed transcoder and the original tandem transcoder.

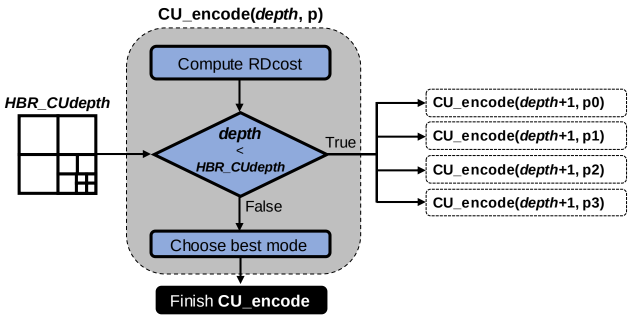

Fig. 3: CU splitting process using side information from decoder.

TABLE V: EXPERIMENTAL RESULTS MEASURED IN BD-RATE (%), BD-PSNR (DB) AND ENERGY SAVING (%).

Sequence	BD-rate (%)	BD-PSNR (dB)	ES (%)
Rollercoaster	1.111	-0.016	73.9
Kimono	0.854	-0.028	57.8
ParkScene	0.442	-0.015	45.6
Cactus	1.264	-0.016	44.2
BQTerrace	0.961	-0.011	37.8
BasketballDrill	0.578	-0.025	41.4
BQMall	0.848	-0.031	39.1
PartyScene	0.406	-0.017	29.6
BasketballPass	0.659	-0.033	41.4
BQSquare	0.041	-0.002	42.1
FourPeople	0.882	-0.031	67.4
Johnny	0.692	-0.016	69.5
ChinaSpeed	0.741	-0.040	38.3
SlideShow	2.212	-0.187	77.1
Average	**0.835**	**-0.033**	**50.4**

A. Encoding Efficiency

The Bjøntegaard Delta (BD)-rate and BD-PSNR [12] metrics were used to evaluate the encoding efficiency of the method for all sequences. Table V shows the average results obtained when transrating all video sequences from the HBR bitstream to the four LBR bitstreams.

The BD-PSNR results indicate that the image quality degradation was very small, with an average reduction of 0.033 dB, and the BD-rate results show that the method led to an average bit rate increase of 0.835%. Notice that BD-rate and BD-PSNR represent the same compression efficiency measure, but under different perspectives. This means that either an average loss of 0.033 dB in image quality is noticed or an average bit rate increase of 0.835% is noticed when the strategy is employed.

B. Energy Saving

The transrating scheme was evaluated in terms of energy consumption. These results were obtained with the Running Average Power Limit (RAPL) [13] tool, which is found in Intel architectures, such as Ivy Bridge and Sandy Bridge. RAPL uses the Model Specific Registers (MSR) to monitor the energy consumption of a processor. To obtain the results, RAPL was run at the same time as the transrating and stopped immediately after its operation. No other application besides the operational system and its basic functions were running along with the transcoder.

Energy savings (ES) results are presented in Table V. The numbers indicate the percentage ES in comparison to the energy consumed by the original tandem transrating. The obtained values show that the strategy is capable of reducing the transrating energy consumption significantly, with an average reduction of 50.4% in comparison to the original tandem transcoder.

The video that achieved the greatest reduction in energy consumption was *SlideShow*, which reached an ES value of 77.1%. However, this was also the sequence that presented the largest BD-rate increase among all tested cases (2.212%). This happens because *SlideShow* is the video sequence that presents the most homogeneous texture. For this reason, it is usually encoded with the largest CUs (64×64) in the HBR case. Thus, when transrating to LBR cases, the proposed strategy limits most of the CUs to 64×64, decreasing significantly the amount of encoding operations and energy consumption. However, when an inappropriate decision is taken (i.e., the CU should be smaller than 64×64 but was limited to that size due to the proposed strategy), a large area is affected, causing a significant decrease in compression efficiency.

Oppositely, the worst results in terms of energy savings are for the *PartyScene* video, which still managed to achieve an ES of 29.6% at the cost of a BD-rate increase of 0.406%. This is explained because *PartyScene* is highly heterogeneous sequence with detailed motion, generally encoded with many 8×8 and 16×16 blocks. This way, the limitations applied to the reencoding process are not so significant, since larger blocks still need to be tested.

VI. Conclusion

This work presented a strategy to reduce the transrating energy consumption in a homogeneous HEVC transcoder based on the inheritance of CU partitioning information from the decoding to the reencoding process. The strategy allowed significant results to be achieved, with an average transrating energy reduction of 50.4% in comparison to the original transcoder. Despite these results, the encoding efficiency did not show significant loss, with a BD-rate increase of only 0.835%, on average. The proposed strategy is especially useful for video streaming services that employ online transrating, thus requiring multiple transcodings for bit rate adaptation upon user request. It is also useful for offline transrating in energy or computationally-constrained systems.

For future work, other approaches can be explored, such as information from neighboring CUs, motion vectors, or even machine learning algorithms, which may help to achieve more significant results and higher coding efficiency.

References

[1] JCT-VC Editors, “Recommendation ITU-T H.265 – High Efficiency Video Coding” (ITU-T Rec.H.265), April 2013.

[2] JVT Editors, “Draft ITU-T Recommendation and final draft international standart of joint video specification” (ITU-T Rec.H.264|ISO/IEC 14496-10 AVC), 2003.

[3] G. J. Sullivan et al, “Overview of the High Efficiency Video Coding (HEVC) Standart.” IEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1649-1668, 2012.

[4] G. Corrêa, et al. “Performance and Computational Complexity Assessment of High-Efficiency Video Encoders.” IEEE Transactions on Circuits and Systems for Video Technology, vol.22, pp.1899-1909, 2012.

[5] E. Monteiro et al. “Rate-Distortion and Energy Performance of HEVC and H.264/AVC Encoders: a Comparative Analysis.” IEEE International Symposium on Circuits and Systems (ISCAS).

[6] CISCO, “Cisco Visual Networking Index: Forecast and Methodology, 2016–2021,” 2017.

[7] M. Khan et al. “Hardware-software collaborative complexity reduction scheme for the emerging HEVC intra encoder.” Automation and Test in Europe Proceedings of the Conference on Design, 2013.

[8] A. Mercat et al. “Prediction of Quad-Tree Partitioning for Budgeted Energy HEVC Encoding” IEEE International Workshop on Signal Processing Systems (SiPS), 2017.

[9] W. Penny et al. “Pareto-based energy control for the HEVC encoder” IEEE International Conference on Image Processing (ICIP), 2016.

[10] ISO/IEC-JCT1/SC29/WG11, “High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description,” in Geneva, Switzerland, ed. 17 October 2014.

[11] K. Sharman et al, “Commom Test Conditions for HM,” ISO/IEC JCTVC-Z1100. Geneva Jun. 19, 2017.

[12] G. Bjøntegaard, “Calculation of average PSNR differences between RDcurves,” Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, Apr. 2001

[13] J. Pan. “RAPL (Running Average Power Limit) Driver.” Available: http://lwn.net/Articles/545745/. Last Access: Sep. 2017.