Direct and Semi-Direct Frame Partitioning Mapping for Computation and Energy-Aware HEVC Transrating

Thiago Bubolz, Ruhan Conceição, Mateus Grellert*, Bruno Zatt, Luciano Agostini, Guilherme Correa

Video Technology Research Group (ViTech), Federal University of Pelotas (UFPel), Brazil
*Embedded Computing Lab, Federal University of Santa Catarina (UFSC), Brazil
{tlabubolz, radconceicao, zatt, agostini, gcorrea}@inf.ufpel.edu.br, mateus.grellert@ufsc.br

***Abstract*— The most common operation to maintain various versions of the same video sequence under different bitrates in live and streaming video applications is video transrating. The typical transrating operation is comprised of a decoding and a com-plete encoding step in sequence, which results in long processing times and high energy consumption. This work proposes two early-termination schemes based on frame partitioning inheritance between decoding and encoding to accelerate the complex quadtree structure decisions performed during HEVC transrating. The proposed Direct Mapping scheme achieved time saving results of 48.5%, with a compression efficiency loss of 0.818%, on average. The Semi-Direct Mapping reaches 40.2% in time savings at the cost of an average compression efficiency loss of 0.724% in comparison to the original transcoder. Energy consumption was reduced in 50.4%, on average, considering the Direct Mapping approach.**

***Index Terms*— HEVC, video coding, transrating, transcoding, complexity re-duction, energy efficiency.**

I. Introduction

According to a recent report published by CISCO [1], video streaming and downloading services, whose clients include smart TVs, cell phones and PCs, will be responsible for more than 80% of the internet traffic by 2021. In order to cope with the heterogenous sources that transmit/receive digital videos on the internet, each with its own capabilities and device characteristics, streaming servers must be able to deliver bitstreams encoded with different standards (usually called heterogeneous transcoding). In addition, it is also common to re-encode sequences using the same codec under different constraints, which is known as homogeneous transcoding. Homogeneous transcoding can be employed to modify the video resolution, to insert watermarks in the video, and to change the encoded-video bitrate (transrating). Both operations are usually implemented with a partial decoding followed by a full re-encoding step using the same (transrating) or a different (transcoding encoder).

Due to the increasing use of video streaming services such as YouTube and Netflix, transrating has become an essential task, since it is necessary to maintain several versions of the same video with different bitrates on the server side. As the encoding step requires a long processing time, transrating is usually performed offline, and the several bitstream versions are stored in servers for future requests. On one hand, often-accessed contents benefit from this strategy, since they are promptly available for users. On the other hand, rarely-accessed videos are also stored in the server in multiple versions, wasting valuable storage resources. Thus, transrating for videos seldom accessed could be performed on-the-fly upon user request.

Real-time transrating is a challenge, especially when complex encoders like those that follow the High Efficiency Video Coding (HEVC) [2] are used. HEVC was planned to double the H.264/AVC [3] compression rate for the same image quality while keeping the encoding complexity at acceptable levels [4]. However, reports show that HEVC requires up to 500% more computational effort than its predecessor [5]. This increased complexity also leads to high (trans)coding energy consumption. In [6], the authors show that the reference HEVC encoder consumes 17% more energy when compared to the reference H.264/AVC encoder. Since video transrating can be done several times on the server, reducing the encoding energy consumption of transrating is highly required. With this in mind, [7] presents a hardware-software collaborative energy reduction. However, it either does not achieve significant levels of energy reduction or indicates decreases in image quality, which is critical in services that value the user quality of experience.

This work proposes a fast and energy-efficient transrating method for the HEVC standard. To accomplish this, the HEVC transrating process was modified to inherit the frame partitioning information from a reference bitstream and use it to speed up the subsequent re-encodings. Partitioning decisions extracted at the decoding step are used as input in heuristics that skip some partitioning evaluations during the re-encoding, reducing processing time and energy consumption.

This paper is organized as follows. Section II presents a background overview and related work. Section III presents a statistic evaluation on frame partitioning, and Section IV presents the proposed transrating scheme. Section V presents the obtained results. Finally, section VI concludes the paper.

II. Background and Related Work

A. HEVC Frame Partitioning Structures

To achieve high compression efficiency results, the HEVC standard introduced some new tools, among them a much more flexible partitioning scheme when compared to its predecessor. These new partitions allow greater variability for content and video resolution. Therefore, it is able to more efficiently encode videos with higher resolutions and to adapt more efficiently to different types of texture and motion. Each video is divided into static images called frames, and each of these frames are also subdivided into a partitioning structure of equal size of 64x64 pixels. This partitioning structure is called the Coding Tree Unit (CTU), which is

Digital Object Identifier 10.29292/jics.v15i1.56

responsible for subdividing the video frame into smaller parts according to the region characteristics.

Each CTU is then subdivided into smaller partitions called Coding Units (CU), which can assume variable sizes of 64x64, 32x32, 16x16 and 8x8 pixels. CUs are split recursively in a quadratic tree structure and the best partitioning is decided after all possibilities are tested and compared in terms of rate-distortion (RD) cost, which requires performing the full encoding process for the candidate CU, checking the required bitrate and the distortion of the encoded CU. Figure 1 shows this quadratic tree scheme that is employed in HEVC, with all its possible depths and subdivisions. As HEVC allows up to four partitioning levels, the computational complexity involved in this decision process is extremely high. In fact, previous works that analyze the computational cost of HEVC have shown that most of its complexity is associated to the complex decision of finding the best partitioning structures [4].

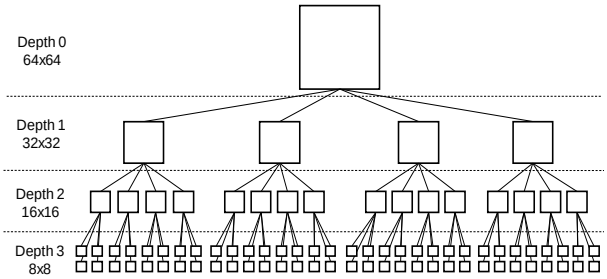

Fig. 1: Example of a CTU partitioned into several CUs following the HEVC quadtree structure.

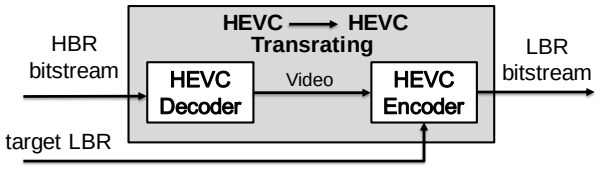

Fig. 2: CU splitting process using side information from decoder.

B. Video Transcoding

Video transcoding systems can be classified in two categories: heterogeneous and homogeneous transcoding. This work focuses on a specific type of homogeneous transcoding called transrating, which is performed to change the video bitrate while still maintaining the same video encoding standard. Transrating is especially useful for streaming service providers that need to store more than one version of the same video content for different types of users, allowing on-the-fly adaptivity to the client network bandwidth.

The typical implementation of a homogeneous transcoder (usually referred as serial or tandem transcoder) starts by decoding a bitstream encoded according to a standard, generating a video output. Then, this video is used as input to an encoder that follows the same standard but employs different encoding parameters to change the bitstream, such as the target bitrate or the quantization parameter (QP). Thus, as the HEVC transcoder consists of an HEVC decoder and an HEVC encoder in sequence, it requires even longer processing times and greater energy consumption than a single HEVC encoder. Figure 2 illustrates an HEVC transcoder for bitrate adaptation, which begins by decoding the high bitrate (HBR) bitstream and then re-encodes the generated video sequence as a lower bitrate (LBR) bitstream, following a target bitrate parameter.

C. Related Works

Due to the high complexity of HEVC and consequently of transcoding systems that employ this encoder, several works that aim at reducing the high computational cost of frame partitioning decisions have been published in the literature in the last years.

In [8], the authors present a method for fast spatial rescaling, which uses the number of CU partitions in the high-resolution video to limit the partitioning decision while transcoding for lower resolutions for bitrate reduction. The method is based on similarities between CUs in the higher resolution version and in the lower resolution.

In [8], the authors also propose a technique for reducing the computational cost of spatial resolution adaptation. They propose the use of a set of random forests to quickly perform CU splitting decisions based on the co-located blocks in the video bitstream with the largest resolution.

The authors in [9] propose a CU early termination solution to speed up the HEVC transrating process based on three methods that reuse information from the original bitstream, such as motion vectors, average CU depths and the rate-distortion (RD) costs of co-located CUs.

This paper also proposes a strategy that inherits CU partitioning information from the decoding step to speed up the reencoding process. However, differently from [8, 9], it focuses on transcoding for lower bitrates, which presents specific characteristics. A statistical evaluation presented in the next section emphasizes these characteristics, which are then explored in the proposed transrating scheme.

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 16.4 [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: *ToddlerFountain*, *Rollercoaster*, *BasketballDrive*, *KristenAndSara*, *SlideEditing*, *RaceHorses* and *BlowingBubbles*.

Table I shows all the specifications for the analysis described in this section, as well as for the testing and results setup. 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 bitrates for each video were calculated as 80%, 60%, 40% and 20% of the bitrate 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.

Tables II-V show average correlation results in percentages for the four transrating processes performed in this analysis. The rows in each table represent each CU size chosen by the encoder during the original encoding process (i.e., the HBR bitstream, encoded with QP=22). The columns in each table represent the CU sizes chosen during the transrating to the LBR cases (i.e., the 80%, 60%, 40% and 20% bitrates in Tables II to V, respectively). For example, in Table IV (LBR=40% of the original bitrate), considering CUs encoded as 32×32 in the HBR bitstream, 52.28% of them were encoded in the same size (32×32) when transrating to LBR, whereas 44.6% of them were encoded as larger CUs (64×64) and 3.07% were encoded as smaller CUs (2.74% as 16×16 CUs and only 0.33% as 8×8 CUs).

Figure 3 shows average results for all the LBR transratings considered in the analysis. Notice that the same 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 bitrates, but rarely a smaller CU size. Specifically for 64×64 CUs, as there are no larger CUs possible, in 93.09% of the cases the same CU size is employed on average, whereas only in 6.59% of the cases a smaller CU is used. For 32×32 and 16×16 CUs, only in 5.76% and in 0.86% of the cases smaller CUs are chosen, respectively. As there are no CUs smaller than 8×8, the same CU size or a larger CU size is necessarily employed when transrating for LBR.

Thus, the analysis presented in this section reveals that there is usually a small chance of a CU being encoded as smaller partitions when transrating from HBR bitstream to LBR bitstreams. This is expected because using smaller CUs requires including more side information to the bitstream, such as block headers, motion vectors, etc. This way, when transrating for reduced bitrates, it is expected that larger (and less numerous) CUs are used. It is important to note that there are still 5.76% of 64×64 CUs transcoded as 32×32 on average, which is still a considerable percentage that can cause significant compression efficiency loss if not treated properly. This observation led to the proposal of two different fast transrating schemes, which are described in the next section.

IV. Low-Complexity Transrating Schemes

The analysis presented in the previous section led to conclusions that guided the two schemes for a low-complexity transrating scheme presented in this section, aiming at reducing computational cost and energy consumption. Both schemes are directly linked to the statistical evaluation presented in the previous section. In both cases, the main idea consists of stopping earlier the search for the optimal quadratic tree that represents the best CTU partitioning.

The analysis revealed that when transcoding an HBR bitstream to LBR, a CU is hardly partitioned in a smaller size than in the HBR version, so there is usually encoded at the same or lower quadtree depth. Both proposed schemes explore this characteristic, as presented in the following paragraphs.

TABLE I: ANALYSIS AND TESTING SET CONFIGURATION

Codec	HEVC Model 16.4 (HM 16.4)
Configuration	Random Access
Base QP*	22
Bitrate Ratios	80%, 60%, 40% and 20%
Training Sequences	*ToddlerFountain, Rollercoaster, BasketballDrive, KristenAndSara, SlideEditing, RaceHorses, BlowingBubbles*
Testing Sequences	*ToddlerFontain, Rollercoaster, Kimono, ParkScene, Cactus, BQTerrace, BasketballDrive, BasketballDrill, BQMall, PartyScene, BasketballPass, BQSquare, FourPeople, Johnny, ChinaSpeed, SlideShow*

*used to obtain the High Bitrate (HBR)

TABLE II: 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	**86.92**	10.16	1.83	1.05
32×32	27.51	**64.59**	6.62	1.26
16×16	9.41	19.95	**65.47**	5.15
8×8	5.34	10.72	21.76	**62.16**

TABLE III: 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	**91.83**	7.00	1.07	0.10
32×32	39.92	**55.54**	4.02	0.52
16×16	17.34	23.07	**56.99**	2.60
8×8	14.76	16.98	22.48	**45.77**

TABLE IV: 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	**95.48**	4.07	0.42	0.03
32×32	44.66	**52.28**	2.74	0.33
16×16	20.01	23.50	**54.49**	2.00
8×8	13.50	17.70	22.59	**46.21**

TABLE V: 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	**98.16**	1.71	0.12	0.01
32×32	55.17	**43.27**	1.42	0.14
16×16	29.67	22.22	**47.39**	0.72
8×8	22.37	21.18	19.40	**37.04**

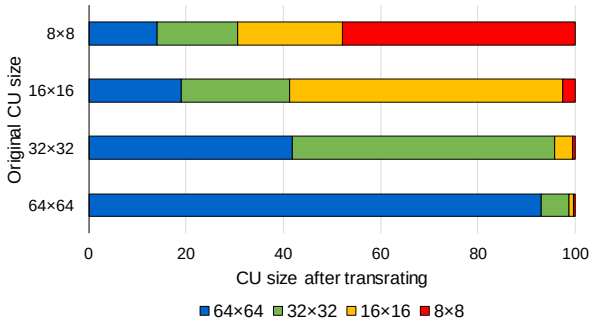

Fig. 3: Average CU partitioning correlation after transrating for different LBR.

A. *Direct Mapping*

The first proposed scheme, named as Direct Mapping, directly employs the partitioning of CUs observed during the decoding process within the transrating to assist CU splitting decisions during the reencoding. Notice in the Fig. 2 that since there is a decoding process before the reencoding process, it is possible to retrieve information from the decoder, such as the CU depth.

Figure 4 shows the Direct Mapping scheme implemented in the transcoder. For each frame that the HEVC decoder decodes from the HBR bitstream, the quadtree depth of every 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 HEVC encoder, a mapping of the CU depths for this corresponding frame is also delivered (*HBR CU Depth Map*, in Fig. 4) and the transcoder uses it to speed up the reencoding.

During the HEVC encoding process, the recursive search for the best quadtree partitioning will continue only if the current CU depth is smaller than the depth obtained from the *HBR CU Depth Map*. Oppositely, if the current CU depth is larger than or equal to the depth obtained from the *HBR CU Depth Map*, the CU splitting process is halted. For example, if a CU was decoded with size 32×32 (i.e., depth 1), the recursive CU splitting will be forced to stop at depth 1.

In summary, the Direct Mapping scheme aims at applying in the second encoding at most the same quadtree depth used in the first encoding, limiting significantly the number of partitioning possibilities to be tested. As previously discussed, if an HBR-to-LBR transrating is considered, the number of inaccurate decisions will be very small in most cases, causing small losses in compression efficiency.

B. *Semi-Direct Mapping*

The Semi-Direct Mapping scheme is proposed in this article based on the observation from Fig. 3 that specifically for 64×64 CUs there is a larger amount of outstanding cases in which the transcoded CUs assume a smaller size. This is expected because the encoder does not allow CUs larger than 64×64, so that the transcoded units will necessarily be encoded with the same size or as smaller block. Fig. 3 showed that in 93.09% of the cases the same CU size is employed on average, whereas in 6.59% of the cases a smaller CU is used. This means that the scheme presented in the previous section will lead to inaccurate decisions and therefore to compression efficiency losses in 6.59% of the 64×64 CUs.

To overcome this, the Semi-Direct Mapping allows that CUs encoded as 64×64 in the HBR bitstream are allowed to advance one more level when searching the best quadtree composition in the LBR encoding. In other words, CUs encoded as 64×64 in the HBR will be tested as 64×64 and as 32×32 in the LBR encoding. This modification allows that inaccurate decisions decrease from 6.59% to only 0.6% in 64×64 CUs.

The same flowchart presented in Fig. 4 applies for the Semi-Direct Mapping, except for the test that halts the recursive CU splitting, which is different for 64×64 CUs. For 32×32 and 16×16 CUs, the same test performed by the Direct Mapping scheme is used: if the current CU depth is larger than or equal to the depth obtained from the *HBR CU Depth Map*, the CU splitting process is halted; otherwise, it continues. For 64×64 CUs, if the current CU depth is larger than or equal to the depth obtained from the *HBR CU Depth Map* plus one, the CU splitting process is halted; otherwise, it continues.

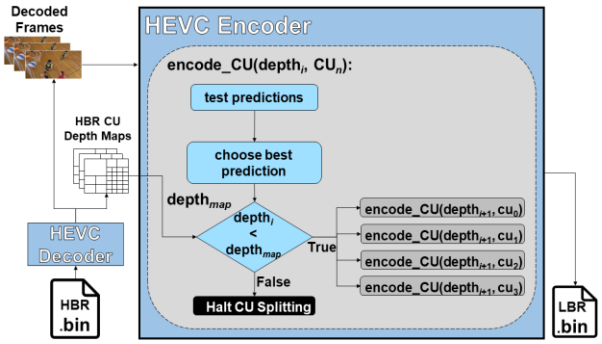

Fig. 4: Proposed CU splitting process for the Direct Mapping scheme using side information from the decoder.

V. Testing and Experimental Results

The experiments presented in this section were conducted in the same conditions of the analysis presented in section III, except for the video sequences that are different from those to avoid biased results. All videos used in the experiments are listed in the last row of Table I, and they include classes A2, B, C, D, E from the CTC document [11]. To evaluate the proposed method in terms of encoding efficiency, time savings 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 described 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, two modified versions of the transcoder were implemented with the schemes proposed in section IV.

The proposed transrating schemes were also executed for the 15 video sequences, taking the same four LBR cases as target bitrates. Each video sequence and LBR combination was executed under the two schemes (i.e., the Direct Mapping and the Semi-Direct Mapping), totalizing 120 transcodings. Thus, the results presented in this section are comparisons between the two proposed schemes and the original tandem transcoder, which totalizes 180 transcodings.

A. *Encoding Efficiency*

The Bjøntegaard Delta (BD)-rate [12] metrics were used to evaluate the encoding efficiency of the schemes. BD-rate corresponds to the bitrate difference between two compared encoding solutions given the same image quality, so that a positive BD-rate value indicates bitrate increase (i.e., loss of compression efficiency). The BD-rate is usually calculated based on the bitrate and the peak-to-noise ratio (PSNR) obtained by encoding a video sequence according to four different Quantization Parameters (QP). As streaming systems

TABLE VI: OBTAINED RESULTS IN TERMS OF BD-RATE, TIME SAVINGS (TS) AND BD-RATE/TS (BD/TS).

Sequences	Direct Mapping			Semi-Direct Mapping		
	BD-rate (%)	TS (%)	BD/TS (×100)	BD-rate (%)	TS (%)	BD/TS (×100)
ToddlerFountain	0.534	45.0	1.19	0.435	39.2	1.11
Rollercoaster	1.111	69.9	1.59	1.103	56.3	1.95
Kimono	0.854	57.4	1.49	0.577	49.0	1.17
ParkScene	0.442	44.7	0.99	0.678	43.7	1.55
Cactus	1.264	39.7	3.18	1.095	35.9	3.04
BQTerrace	0.961	39.7	2.42	0.649	32.4	2.00
BasketballDrill	0.578	40.2	1.44	0.503	30.9	1.62
BQMall	0.848	42.1	2.01	0.781	33.2	2.34
PartyScene	0.406	33.3	1.22	0.511	31.6	1.61
BasketballPass	0.659	30.0	2.20	0.635	25.4	2.49
BQSquare	0.041	42.8	0.09	0.032	31.4	0.10
FourPeople	0.882	63.3	1.39	0.726	49.3	1.47
Johnny	0.692	71.3	0.97	0.763	57.9	1.31
ChinaSpeed	0.741	37.9	1.96	0.789	33.2	2.36
SlideShow	2.212	70.3	3.15	1.580	53.2	2.96
Average	**0.818**	**48.5**	**1.69**	**0.724**	**40.2**	**1.81**

are focused on guaranteeing a set of target bitrates, instead of four different QPs, four different LBR bitstreams presented in the previous sections are used in the BD-rate computation.

Table VI shows the average results obtained when transrating all video sequences from the HBR bitstream to the four LBR bitstreams according to the Direct Mapping and the Semi-Direct Mapping schemes. BD-rate values results show that the Direct Mapping scheme led to an average bit rate increase of only 0.818%. On the other hand, the Semi-Direct Mapping method achieves better results in terms of compression efficiency, with BD-rate of 0.724%. This was expected since in the second method allows a more flexible decision for CUs originally encoded as 64×64 in the HBR bitstream.

The worst-case results in compression efficiency are for the *SlideShow* video sequence, which presented a BD-rate of 2.212% in Direct Mapping and 1.580% in Semi-Direct Mapping. This happens because *SlideShow* is the most homogeneous video sequence, exclusively composed of screen content. Thus, with a much larger occurrence of 64×64 CUs, the quadtree search is limited already at the first depth in most cases considering the Direct Mapping scheme. Notice that *SlideShow* is also the case with larger BD-rate decrease when the Semi-Direct Mapping is used instead of the Direct Mapping. Since most CUs in this video are 64×64, compression efficiency increases significantly when the Semi-Direct Mapping scheme is employed, allowing one more CU size to be tested.

The sequence that presents the best compression efficiency results in both schemes is *BQSquare*. For this sequence, BD-rate results are very close to zero, which means that compression efficiency losses are negligible. It is important to emphasize, however, that this a spatially heterogeneous video sequence with small resolution (320×240 pixels). This leads the original encoding process to decide more frequently for smaller CUs (16×16 and 8×8) in the HBR bitstream, and consequently halts the LBR encoding process at deeper quadtree levels.

Table VII shows results for average bitrate difference inpercentage. Notice that negative values indicate that the video encoded with the direct mapping technique achieved a smaller bitrate than the original bitstream. Even when values are positive (which indicates that the proposed solution increased the bitstream size), the values are negligible, increasing a maximum of 0.0003% for ParkScene. This shows that even when the algorithm skipped some CU sizes the rate control tool kept the bitrate as close as possible from the target.

In addition, Table VII shows the SSIM (Structural similarity) between sequences encoded with the original and the proposed encoders (using Direct Mapping). This metric is used to compare local patterns of pixel intensities that have been normalized for luminance and contrast [13]. When SSIM is closer to one, it indicates that the image quality is less degraded. It is possible to observe that the SSIM values were greater than 0.96, which means that the proposed solution produces a negligible image degradation over the baseline encoder. Therefore, it is possible to conclude that the proposed algorithm does not impose significant losses in terms of both image quality and bitrate.

Figure 5 shows a set of images from the *BQTerrace* sequence that illustrate the compression efficiency of the Direct Mapping scheme. The figure compares the partitioning obtained by the original tandem transcoder and by the Direct Mapping scheme when transcoding the 60% LBR bitstream and the 40% LBR bitstream. In both cases, it is noticeable that the obtained partitioning is very similar between the

TABLE VII: AVERAGE RESULTS IN TERMS OF Δ BITSTREAM (ΔB) AND SSIM (STRUCTURAL SIMILARITY INDEX) FOR DIRECT MAPPING.

Direct Mapping		
Sequence	***ΔB (%)**	**SSIM**
ToddlerFountain	$+7e^{-07}$	0.973
Rollercoaster	$-1e^{-04}$	0.994
Kimono	$-6e^{-05}$	0.974
ParkScene	$+3e^{-04}$	0.975
Cactus	$+3e^{-06}$	0.980
BQTerrace	$-8e^{-05}$	0.977
BQMall	$-1e^{-05}$	0.978
PartyScene	$-1e^{-04}$	0.969
BasketballDrill	$+7e^{-06}$	0.967
BQSquare	$-9e^{-04}$	0.975
BasketballPass	$+4e^{-06}$	0.966
FourPeople	$-1e^{-04}$	0.987
Johnny	$-6e^{-04}$	0.991
ChinaSpeed	$+1e^{-06}$	0.976
SlideShow	$-2e^{-03}$	0.992
Average	$-2e^{-04}$	0.978

*ΔB means the percentage increase of the bitstream, in positive cases, and the decrease of the bitstream, in negative cases, comparing the original encoding with the modified encoding

original and the modified transcoder, with few differences (mostly in 64×64 CUs). These similar decisions are responsible for the high compression efficiency achieved by both schemes.

B. Time Savings

Table VI also shows time savings (TS) results achieved with the two proposed schemes. The values show that the two strategies are capable of reducing transrating time significantly, with an average reduction of 48.5% for the Direct Mapping and 40.2% for the Semi-Direct Mapping. As expected, the Semi-Direct Mapping scheme reaches smaller

$$TS = \frac{T_O - T_M}{T_O} \quad (1)$$

time savings in comparison to the original tandem transcoder than the Direct Mapping, since it allows further tests in the recursive quadtree splitting.

TS results are calculated according to (1), where T_O represents the processing time required by the original transcoder and T_M represents the processing time required by the transcoder modified according to one of the schemes proposed in section IV.

The video that achieved the greatest reduction in transrating time was *Johnny*, with a TS of 71.3% for Direct Mapping and 57.9% for Semi-Direct Mapping. This is because the sequence has a High Definition resolution (1024×720) and is composed of several homogeneous regions along the frame, which tend to assume larger partitions when transcoded for lower bitrates. The worst-case result in time savings is for the *BasketballPass* video, a temporally and spatially heterogeneous sequence, which still managed to achieve a TS of 30% and 25.4% for both methods at the cost of a BD-rate increase of 0.659% and 0.635%.

For fair comparison, Table VI also presents the ratio between BD-rate and TS (BD/TS), which shows the amount of encoding efficiency loss for each percentage in time savings. The numbers are multiplied by 100 as shown in (2) for clarity. On average, an increase of 0.0169% in BD-rate is required for each one percent of TS achieved by Direct Mapping and an average increase of 0.0181% in BD-rate is required for each one percent of TS achieved by Semi-Direct Mapping. These numbers are especially useful for comparisons with related works, as shown later.

$$BD/TS = \frac{BD\text{-}rate}{TS} \times 100 \quad (2)$$

C. Energy Saving

The transrating scheme was also evaluated in terms of energy consumption. These results were obtained with the Running Average Power Limit (RAPL) [14] tool, which is found in some 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

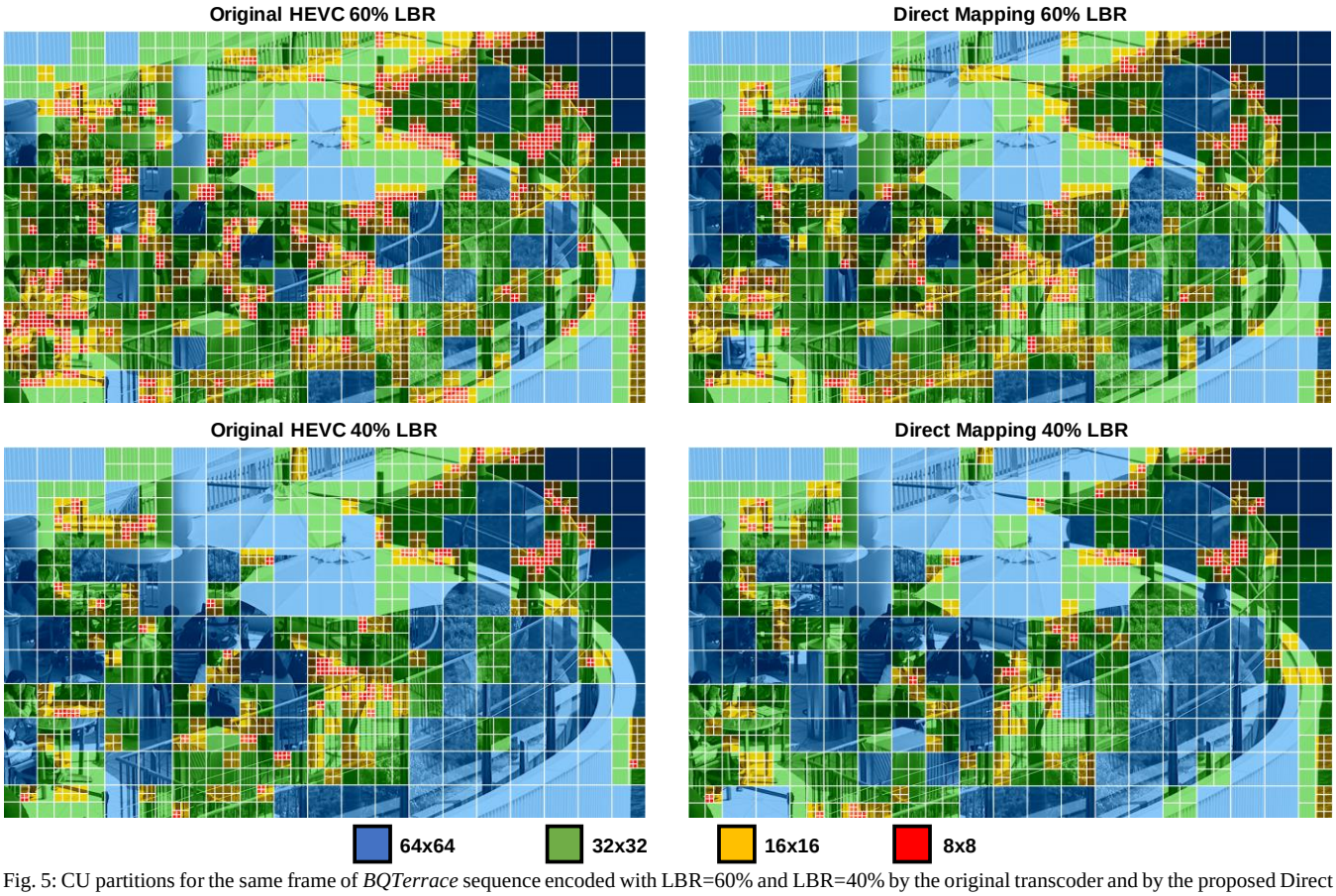

Fig. 5: CU partitions for the same frame of *BQTerrace* sequence encoded with LBR=60% and LBR=40% by the original transcoder and by the proposed Direct Mapping scheme.

TABLE VIII: AVERAGE RESULTS IN TERMS OF ENERGY SAVING (ES) AND BD-RATE/ES (BD/ES) FOR DIRECT MAPPING.

Direct Mapping		
Resolution	**ES (%)**	**BD/ES (×100)**
4K	73.9	1.50
Full HD	46.3	1.96
HD	63.0	1.77
480p	36.7	1.64
Average	**50.4**	**1.66**

application besides the operational system and its basic functions were running along with the transcoder. RAPL tool has been used in other previous works as a mean of estimating energy consumption in HEVC encoders [15, 16].

Firstly, the HEVC encoder is executed and the consumed energy is collected from the RAPL report. Then, a second process is called running the sleep command for the same duration of the encoding task. This is necessary to remove the energy overhead spent on OS tasks.

Energy savings (ES) results are presented in Table VIII for the Direct Mapping scheme only. The numbers indicate the ES in percentage in comparison to the energy consumed by the original tandem transrating. On average, an ES of 50.4% was achieved by the Direct Mapping scheme. Table VIII also shows the ratio between BD-rate and energy saving (BD/ES). Similarly to BD/TS, this ratio evaluates the amount of encoding efficiency loss for each percentage in energy savings achieved.

The energy savings analysis was performed only for the Direct Mapping scheme, since it showed the best tradeoff between time savings and compression efficiency in the previous section. As expected, the energy savings results are closely correlated with time saving ones.

D. Estimate Memory Bandwidth Reduction

Skipping CU computations benefits not only encoding time, but memory bandwidth as well. As stated in section II, Motion Estimation (ME) is one of the encoding tools nested inside each CU node of the quadtree, and this is reportedly the most computation and memory-intensive HEVC task, consuming up to 45% of the overall encoder memory bandwidth [17]. This high bandwidth comes mostly from accessing the reference picture samples required in the search step, which are stored in the Decoded Picture Buffer (DPB). This buffer is sometimes implemented as an online cache, reducing the off-memory communication bottleneck, but the online bandwidth remains a concern regarding throughput and energy restrictions. Therefore, reducing the bandwidth is always beneficial.

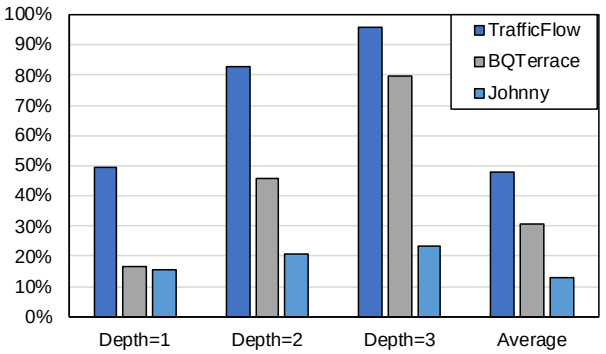

Fig. 6: Memory bandwidth reduction at each CU depth during the Motion Estimation step.

To compute the memory bandwidth of the original and the proposed transcoders, the HM reference was instrumented to report the number of luminance pixels fetched from memory during the ME search for each CU size. Figure 6 shows the estimated memory bandwidth savings achieved by Direct Mapping scheme with respect to the reference encoder at each CU depth, using one 4K and two HD sequences. Note that depth 0 is not represented in the chart, as there is no bandwidth reduction in this case (it is always evaluated in the proposed method).

The chart shows that memory bandwidth is significantly reduced, achieving an average reduction of 48%, 31%, and 13% with the *ToddlerFountain(4096x2160)*, *BQTerrace(1920x1080)* and *Johnny(1280x720)* sequences respectively. The lowest results observed with the *Johnny* sequence are expected, because the HM reference implements fast heuristics that skip some ME calls in regions with low movement [18], which are abundant in *Johnny*.

While most of the savings seem to come from skipping 8×8 CUs in Figure 6, it is important to note that the bandwidth is much higher in larger CUs. In fact, the actual memory bandwidth reduction (in absolute number of samples) obtained with the proposed Direct Mapping scheme is similar across all depths. To exemplify, the absolute reduction obtained with the *ToddlerFountain* sequence was 1.58, 1.95 and 1.77 GB/sec for CU depths 0, 1 and 2 respectively.

It is also important to notice that memory bandwidth reduction tends to be higher in videos with larger resolutions. This occurs because larger resolutions contain more blocks covering homogeneous areas of the scene. Therefore, larger CUs tend to be chosen by the proposed method, which skips more computations, ultimately leading to significant reductions in and memory access.

TABLE VIV: COMPARISONS WITH RELATED WORKS IN TERMS OF BD-RATE, TIME SAVING (TS) AND BD-RATE/TS (BD-TS)

Sequence	**Direct Mapping**			**Semi-Direct Mapping**			**Yang [10]**			**Praeter [9]**		
	BD-rate (%)	**TS (%)**	**BD/ TS**	**BD-rate (%)**	**TS (%)**	**BD/ TS**	**BD-rate (%)**	**TS (%)**	**BD/ TS**	**BD-rate (%)**	**TS (%)**	**BD/ TS**
BasketballDrive	0.86	47.9	1.789	0.75	40.8	1.838	2.01	47.3	4.249	6.40	59.6	10.738
BQTerrace	0.96	39.7	2.423	0.64	32.4	2.001	2.78	57.2	4.860	5.60	70.7	7.921
Cactus	1.26	39.7	3.185	1.09	35.9	3.043	2.47	52.4	4.713	6.50	59.4	10.943
Kimono	0.85	57.4	1.489	0.57	49.1	1.176	1.60	53.3	3.001	4.70	57.8	8.131
ParkScene	0.44	44.7	0.990	0.67	43.7	1.550	1.90	54.0	3.518	4.80	57.4	8.362
Average	**0.88**	**45.4**	**1.909**	**0.74**	**40.3**	**1.921**	**2.15**	**52.8**	**4.068**	**5.60**	**61.0**	**9.180**

E. Comparison with Related Works

The best related works found in the literature [8, 9] and reviewed in section II are compared in this section with the two schemes proposed in this work. Table VIV presents results for the related works and for both Direct Mapping and Semi-Direct Mapping. As each related work achieves different levels of TS, the BD/TS ratio (multiplied by 100) was used to assess the tradeoff between encoding efficiency and time savings for each compared solution. Also, to allow for fair comparisons, only the results for the video sequences used in related works are presented in Table VIV. The average results of both schemes proposed in this work were recalculated based only on these videos.

Table VIV shows that the Direct Mapping scheme achieves the best tradeoff between BD-rate and TS, with an average BD/TS ratio of only 1.909 for the considered video sequences. This means that, on average, an increase of 0.01909% in BD-rate is required for each one percent of TS achieved by Direct Mapping. The Semi-Direct Mapping scheme achieves a slightly worst tradeoff, with an average BD/TS of 1.921. Still, the scheme is more useful than the Direct Mapping when compression efficiency needs to be maintained as close as possible to the obtained by original transcoder.

Notice that no other work achieves BD/TS results as low as the two schemes proposed in this work. Also, although [8] achieves the greatest TS in comparison to all compared works (61%), an expressive BD-rate increase of 5.6% is noticed in that work, leading to a BD/TS ratio of 9.18. This means that the proposed scheme presents a better tradeoff in terms of encoding efficiency and TS than the related works when any of the three modes are used.

VI. CONCLUSION

This work presented two schemes to reduce the transrating time and energy consumption in a homogeneous HEVC transcoder based on the inheritance of CU partitioning information from the decoding to the re-encoding process. The schemes allowed significant results to be achieved, with an average transrating time reduction of 48.5% for the Direct Mapping scheme and 40.2% for the Semi-Direct Mapping scheme in comparison to the original transcoder. Energy consumption was reduced in 50.4%, on average, considering the Direct Mapping approach. Despite of that, encoding efficiency did not show significant losses, with a BD-rate increase of only 0.818% and 0.724%, respectively. Comparisons with related works that focus on transrating complexity reduction show that both proposed schemes achieve the best tradeoff between time savings and compression efficiency.

The proposed schemes are 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. Future work include the inheritance of other information from the decoding to speed up the re-encoding, such as motion vectors and prediction modes. Besides, machine learning-based approaches, which may help to achieve more significant results in terms of encoding efficiency, are included in future investigations.

ACKNOWLEDGEMENTS

This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance code 001, the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq), and the Fundação de Amparo à Pesquisa do Rio Grande do Sul - Brasil (FAPERGS).

REFERENCES

[1] VNI Cisco. Cisco visual networking index: Forecast and methodology2016–2021. (2017), 2017.

[2] High Efficiency Video Coding, “Recommendation itu-th. 265,” International Standard ISO/IEC, pp. 23008–2,2013.

[3] Thomas Wiegand, “Draft itu-t recommendation and final draft international standard of joint video specification (itu-t rec. h. 264— iso/iec 14496-10 avc),”JVT-G050, 2003.

[4] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, et al., “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, 2012.

[5] Guilherme Correa, Pedro Assuncao, Luciano Agostini, and Luis A da Silva Cruz, “Performance and computational complexity assessment of high-efficiency video encoders,”IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1899–1909,2012.

[6] 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), 2015

[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] Johan De Praeter, Antonio Jesus D´ıaz-Honrubia, Niels Van Kets, Glenn Van Wallendael, Jan De Cock, Peter Lambert, and Rik Van de Walle. Fast simultaneous video encoder for adaptive streaming. In Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on, pages 1–6. IEEE, 2015.

[9] Shih-Hsuan Yang and Chong-Cheng Zhong. Fast coding-unit mode decision for hevc transrating. In Computer and Information Technology(CIT), 2017 IEEE International Conference on, pages 93–100. IEEE, 2017.

[10] Ken McCann, C Rosewarne, B Bross, M Naccari, K Sharman, and G J Sullivan. High efficiency video coding (hevc) test model 16 (hm16) improved encoder description. Jonit Collaborative Team on Video Coding, JCTVC-S1002, Strasbourg, FR, 2014.

[11] Sharman and C Rosewarne. Common test conditions and software reference configurations for hevc. In Proceedings of the Meeting of Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC Z1100, 2017.

[12] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves.VCEG-M33, 2001.

[13] Wang, Zhou; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. (2004-04-01). "Image quality assessment: from error visibility to structural similarity". IEEE Transactions on Image Processing. doi:10.1109/TIP.2003.819861. ISSN 1057-7149

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

[15] E. Monteiro, *et al.*, "Rate-distortion and energy performance of HEVC and H.264/AVC encoders: A comparative analysis," *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, 2015, pp. 1278-1281.

[16] W. Penny, *et al.*, "Pareto-based energy control for the HEVC encoder," *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, 2016, pp. 814-818.

[17] Sampaio, Felipe, et al. "dSVM: energy-efficient distributed scratchpad video memory architecture for the next-generation high efficiency video coding." Design, Automation & Test in Europe (DATE), Germany, 2014.

[18] Kim, Il-Koo, et al., "CE2: Test results of asymmetric motion partition (AMP)." JCTVC meeting document F379, Torino, 2011.