HEVC-to-AV1 Transcoder Acceleration Based on Block Size Inheritance

Alex Borges, Bruno Zatt, Marcelo Porto, Guilherme Correa
Video Technology Research Group (ViTech)
Computer Science Graduate Program (PPGC) – Federal University of Pelotas (UFPel)
Pelotas, Brazil
{amborges, zatt, porto, gcorrea}@inf.ufpel.edu.br

***Abstract*—With the advent of the recently launched AOMedia Video 1 (AV1) bitstream specification, there is currently a need for converting legacy content encoded with the state-of-the-art High Efficiency Video Coding (HEVC) standard to the new format. However, transcoding is a complex task composed of a decoding and an encoding process in sequence, which requires long processing times and high energy consumption. This paper proposes the first HEVC-to-AV1 transcoding solution, which is based on the high correlation between block size decisions in HEVC and AV1. The solution allows the AV1 encoder to inherit Coding Unit (CU) depth information from the HEVC bitstream to constrain the AV1 re-encoding process. Experimental results show an average transcoding time reduction of 35.21% at the cost of a compression efficiency loss of 5.15%.**

Keywords—transcoding, HEVC, AV1, video coding, complexity reduction

I. Introduction

Since 2013, the High Efficiency Video Coding (HEVC) standard [1] has become the state-of-the-art solution for video compression, gradually replacing its predecessor, H.264/AVC [2], in most applications and multimedia-capable devices. However, HEVC is expensive for commercial use, especially for video streaming companies, due to royalty distribution policies [3]. Aiming at this issue, the Alliance for Open Media (AOMedia) was founded with the goal of developing royalty-free video codecs, such as the recently launched AOMedia Video 1 (AV1) [4][5].

As companies belonging to the AOMedia group announce their adoption to the new format, an increasing interest on AV1 starts to grow in industry and academy. During the forthcoming AV1 adoption, such companies will need to update their HEVC-encoded bitstreams to the new, royalty-free format. Besides, end users will also choose to re-encode their personal videos, aiming at reducing storage requirements, especially for Ultra High Definition (UHD) videos. This re-encoding process, called heterogeneous video transcoding, usually consists of a cascaded connection between a decoder and an encoder, changing the video bitstream format. In the case of an HEVC-to-AV1 transcoder, the HEVC bitstream is decoded, generating a video output that is then re-encoded by the AV1 encoder.

However, the practical use of AV1 still faces a serious obstacle: the high computational cost of its encoding process. Even though recent analyses present positive results in terms of compression efficiency for AV1 [6][7], all of them show that the AV1 codec requires a significantly higher encoding time than previous encoders. For example, the authors in [7] show that the run-time of the current AV1 reference encoder is up to 106 times higher than that of the HEVC reference encoder.

Due to this high computational cost, a simple tandem transcoding system composed of an HEVC decoder and an AV1 encoder sequentially is not recommended. For streaming service providers or even end users that need to transcode several bitstream representations, long processing times and high energy consumption will be required in the migration process between formats. This way, developing strategies for accelerating the HEVC-to-AV1 transcoding becomes mandatory.

Even though several previous works [8][9][10] aim at decreasing the transcoding complexity between different bitstream formats, to the best of authors' knowledge there are no solutions published yet for the HEVC-to-AV1 transcoding process. The paper [10] is the only rudimentary proposal, which aims at using object tracking from the HEVC motion compensation to infer the most probable region for inter prediction in AV1. However, the proposed idea was not implemented, and no experimental results have been presented in [10] to prove its efficiency.

This work presents an HEVC-to-AV1 transcoder, which aims at decreasing the high computational cost required by the original tandem solution. The strategy allows the transcoder to inherit block size information from the HEVC bitstream, which is then used to quickly decide partitions during the AV1 re-encoding process, avoiding the test of all possible partition sizes. As HEVC and AV1 follow different frame partitioning schemes, a statistical analysis was first performed to detect similarities and correlations between partition sizes, which served as the main basis for the proposed strategy.

II. HEVC and AV1 Frame Partitions

Although both HEVC and AV1 are based on the same hybrid block-based video coding flow, they differ from each other in several aspects regarding their coding tools. However, when aiming at inheriting information from one bitstream to accelerate the transcoding process, one would focus on the similarities between the two standards. Even though the HEVC standard allows block sizes and formats very different from those introduced in AV1, most of them can actually be mapped to one another. Thus, this section first presents a comparison between the partitioning schemes used in the two standards and then presents a correlation analysis between them for a set of video sequences.

A. Frame Partitioning Structures in HEVC and AV1

In HEVC, each frame is partitioned into equal-sized square blocks called Coding Tree Units (CTU), composed of 64×64 pixels. Each CTU can be encoded as one single 64×64 Coding Unit (CU) or as a combination of several 32×32, 16×16 or 8×8 CUs, decided in a recursive process. For prediction purposes, CUs can be also divided into Prediction Units (PUs) following different formats. AV1 introduced the Superblock (SB) structure, similar to the HEVC CTU. SBs are square blocks of 128×128 pixels that can be divided according to the partition modes 0-9 shown in Fig. 1. Among the 10 modes, the only case that allows recursively splitting the current block into four square blocks is mode 9 (*SPLIT*). In this case, the four new blocks can be once again divided according to one of the 10 modes. This process is repeated until the minimum 4×4 block size is reached.

Although the partition formats are not the same between HEVC and AV1, the recursive decision process is conceptually similar in both cases and the initial block size at each tree depth (i.e., before splitting) is the same. Thus, it is possible to correlate CU and block sizes that belong to the same tree depth, aiming at finding similarities in the partitioning decisions of the two encoders. Fig. 2 represents an HEVC CTU (left) and an AV1 SB (right), both divided multiple times into smaller CUs/blocks. For each tree depth level (DL), the figure shows the corresponding CU size in HEVC and the initial block size in AV1. Notice that in HEVC the DL varies between DL1 (64x64 CUs) and DL4 (8x8 CUs), whereas in AV1 it ranges between DL0 (128x128 blocks) and DL5 (4x4 blocks).

B. *CU/Block Size Correlation Analysis*

As there are important similarities in the partitioning process of HEVC and AV1, a set of experiments was performed aiming at identifying any possible correlations between CU and block size decisions performed by the two codecs, considering the same image region of the video sequence. The experiments were performed based on 10 frames of eight Netflix recommended video sequences from the XIPH database [11] (*Boat, BoxingPractice, DinnerScene, Narrator, RitualDance, ToddlerFountain, TunnelFlag, and WindAndNature*), which were first encoded in HEVC and then transcoded from HEVC to AV1. The reference software implementations HEVC Model (HM) 16.20 and AOM 1.00 were used for HEVC and AV1, respectively. In the HM encoder, the quantization parameter (QP) was set to 22, which corresponds to the first recommended QP in the Common Test Conditions (CTC) [12]. In AOM, the CQ value was set to 12 after an exhaustive search for the CQ that leads to the most similar image quality (in terms of PSNR) between HEVC and AV1.

During the HEVC decoding and the AV1 re-encoding process, the DL observed for each 4×4-pixel region of the same video was stored for comparison. Table I presents the obtained correlation results. Each row in the table represents the DL observed during the HEVC decoding process, whereas each column represents the DL noticed during the AV1 re-encoding. The results show a significant correlation between the observed DLs. For example, 58.09% of HEVC CUs encoded in DL2 were also re-encoded in DL2 in AV1. When summing up neighboring DL values (i.e., one above and one below the observed in HEVC), the correlation is much higher. Considering the same example, notice that 92.95% of HEVC CUs encoded in DL2 were re-encoded either in DL1 (15.54%), DL2 (58.09%) or DL3 (19.21%) in AV1. These results provide important insight for the complexity reduction proposal presented in the next section.

III. Fast HEVC-to-AV1 Transcoder Proposed

Based on the correlation analysis presented, a fast HEVC-to-AV1 transcoding solution is proposed. The solution is based on the idea of inheriting the decoded CU size information from the HEVC bitstream to infer decisions taken during the AV1 re-encoding process.

During the transcoding from HEVC to AV1, a block partitioning map for every CTU decoded from the HEVC bitstream, named here as DL*map*, is stored. As the smaller block size allowed in AV1 is 4×4, the DL*map* is processed to generate a sequence of numbers representing the DL for each 4×4 region of a frame. Then, when re-encoding the video with AV1, the imported DL*map* is used to constrain the encoding process according to the HEVC partitioning.

As previous shown in section II.*B*, the highest correlation between DL decisions in HEVC and AV1 occurs at the same or at neighboring depths of the coding tree. Thus, the proposed transcoding strategy consists of allowing the AV1 encoder to test only three depths instead of the original six depths allowed: the DL observed in HEVC, the DL above, and the DL below.

Fig. 3 shows the proposed transcoding flow, where the HEVC decoder generates a decoded video sequence and an associated DL*map*, which are both imported by the AV1 encoder. During the re-encoding process, for each SB the current DL value is initially set to 0 (DL*n*=0). Then, the current DL is compared with the value obtained from the DL*map*. As DL*map*–1, DL*map* and DL*map*+1 are allowed depths, two tests are necessary to check if the current DL (DL*n*) is within such limits. If so, all the 10 partition modes are tested and the normal AV1 prediction flow is performed for each mode. The best mode at the current depth is chosen and the current block is recursively split into four square

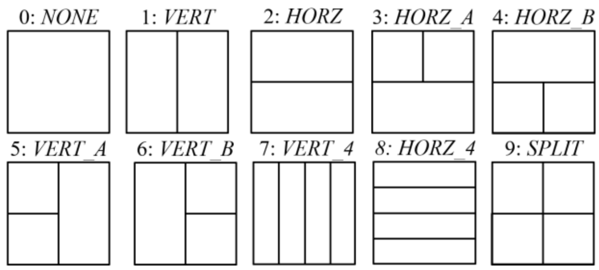

Fig. 1. Partition modes allowed in AV1

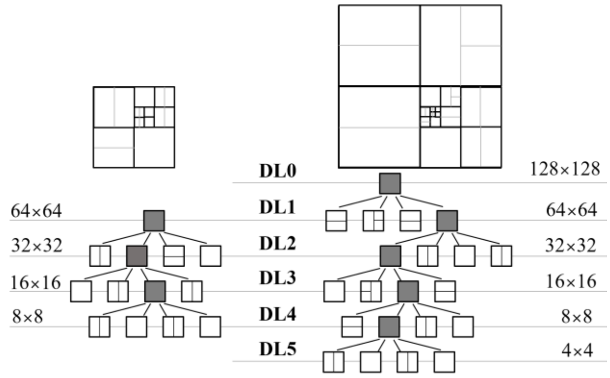

Fig. 2. Example of HEVC CTU (left) and AV1 SB (right) divided into CUs and blocks.

TABLE I. DL Correlation Between HEVC and AV1

DL in HEVC	DL in AV1 (%)					
	0	**1**	**2**	**3**	**4**	**5**
1	22.03	**45.61**	23.80	7.59	0.90	0.06
2	4.32	15.65	**58.09**	19.21	2.54	0.18
3	2.47	7.55	23.45	**56.09**	9.54	0.91
4	1.21	4.18	12.60	39.47	**37.77**	4.77

blocks after incrementing DL*n*. Otherwise, two scenarios can occur: either the current DL is above DL*map*–1 or below DL*map*+1. In the first case, the 10 partition modes are also tested in a simplified prediction scheme for complexity reduction (i.e. only one prediction is performed for each format), but only partition mode 9 (*SPLIT*) can be chosen. The block is then recursively split into four square blocks after incrementing DL*n*. In the second case, no mode is tested, and the recursive process is halted.

IV. Experimental Results

This section first presents the experimental setup to evaluate the transcoding proposal and then presents obtained results in terms of compression efficiency and computational cost.

A. *Experimental Setup*

The HEVC Model (HM) 16.20 reference software was used to encode all video sequences, generating the input bitstream for the HEVC-to-AV1 transcoder. During the encoding process, the *Random Access Main* configuration from the HEVC Common Test Conditions (CTC) [12] was employed and the quantization parameter (QP) was set to 22, since it is the value that yields the best image quality among the four QPs recommended in the CTC.

In the first transcoding step (HEVC decoding), HM 16.20 was used to decode all sequences, generating the input video sequences for the AV1 encoder and the corresponding DL maps. In the second step (re-encoding), a modified version of the reference AV1 encoder (AOM) – version 1.00 (Nov. 20th, 2018) [13] – with the proposed transcoding scheme implemented was used to re-encoded all sequences. The *Low Latency CQP* configuration was used and the constant quality (CQ) parameter values were set to 20, 32, 43 and 55, following the recommendation [14]. To provide a *Random Access Main*-like configuration in AV1 and guarantee a similar temporal prediction structure in both input and output bitstreams, the AOM encoder was configured to allow just the first frame to be encoded as *KeyFrame* in each video sequence.

Seven HD1080 and seven UHD4K video sequences (60 frames) from the *objective-2-fast* and *objective-2-slow* classes recommended in [14] were used in the experiments. For UHD4K videos, both encoders were configured in the 10-bit per sample encoding mode.

B. *Speedup and Compression Efficiency Results*

Table II presents Bjontegaard Delta (BD)-rate and time-saving (TS) results obtained when transcoding the video sequences with the proposed HEVC-to-AV1 transcoder, taking the original tandem transcoder as reference for comparison. The results show that an average transcoding time reduction of 35.21% was achieved at the cost of a compression efficiency loss of 5.15%. The worst-case results in terms of BD-rate is for the UHD4K *Netflix_Dancers* video sequence, whereas the best cases correspond to the UHD4K sequences *Netflix_ToddlerFountain*. About this best case, besides the small losses in compression efficiency, TS results are above 46%. The best TS results are obtained for the HD1080 *DucksTakeOff* sequence, which reached a transcoding speedup of 50% at the cost of a BD-rate of 2.4%.

Table II also shows that the strategy performs better for HD1080 than for UHD4K sequences. Even though very similar compression efficiency results are achieved for both

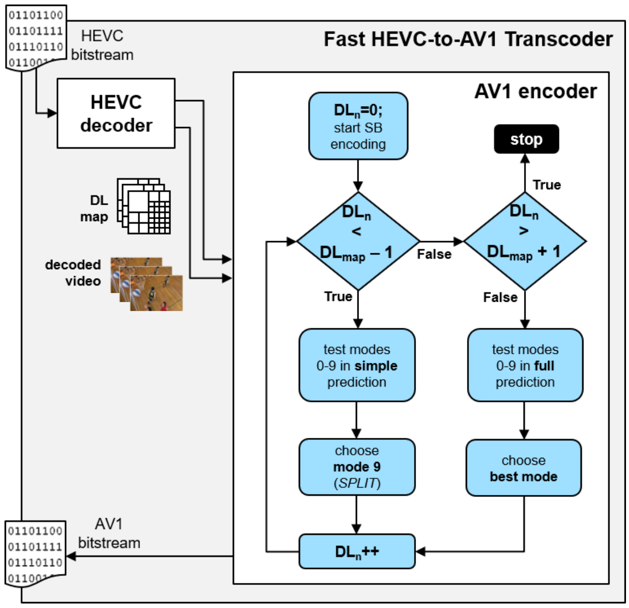

Fig. 3. Algorithm flow for the proposed solution.

TABLE II. Obtained Results for the Proposed Transcoder in Comparison to the Original Tandem HEVC-to-AV1 Transcoder.

	Video Sequence	BD-rate (%)	TS (%)
HD1080 (1920×1080)	*DucksTakeOff*	2.4469	50.31
	Life	8,4439	40,39
	Netflix_Boat	3.9970	38.77
	Netflix_FoodMarket	5.6108	34.02
	Netflix_SquareAndTimeLapse	6.2268	38.41
	Netflix_TunnelFlag	4.5439	38.03
	RushHour	2.6238	35.41
	SeaPlaneHDRAmazon	6.0557	33.92
UHD4K (4096×2160)	*Netflix_BarScene*	7.5995	36.75
	Netflix_BoxingPractice	6.3069	21.42
	Netflix_Dancers	10,4522	27,11
	Netflix_Narrator	4.5629	26.01
	Netflix_RitualDance	4.9895	26.90
	Netflix_ToddlerFountain	1.5973	46.71
	Netflix_WindAndNature	5.4149	28.16
	StreetHDRAmazon	1.6608	41.04
AVERAGE (HD1080)		**4.9936**	**38.66**
AVERAGE (UHD4K)		**5.3230**	**31.76**
AVERAGE (all videos)		**5.1583**	**35.21**

classes, TS results for HD1080 are significantly higher. This can be explained because the proposed method inherits block size decisions from HEVC, which was developed mainly for HD and HD1080 resolutions. However, AV1 was developed to support UHD video coding, including larger block sizes and more efficient tools to encode them. Thus, as UHD video sequences are not efficiently encoded in HEVC, the inherited partitioning decisions from it are not the best decisions for AV1 more frequently.

As there are no other papers published so far aiming at complexity reduction for the HEVC-to-AV1 transcoder, direct comparisons with related works are still impossible.

V. Conclusion

This paper presented a complexity reduction solution for the HEVC-to-AV1 transcoding process based on the high block size correlation between HEVC and AV1 bitstreams. The proposed transcoder allows the inheritance of partitioning information from the HEVC decoder to accelerate the AV1 re-encoding process, which tests only a limited number of block sizes and partitioning formats. When compared to the original, unmodified HEVC-to-AV1 transcoding flow, the proposed transcoder achieves an average time saving of 35.21%, with a compression efficiency loss of 5.15%. Although AV1 is still in its initial steps and its codec will be optimized in the near future to allow a real-time hardware encoding, the strategy proposed in this paper can be implemented in any future HEVC-to-AV1 transcoder, including those that may integrate optimized AV1 codecs. This is an important contribution because accelerating the AV1 encoding process, it is possible to develop a hardware structure with low energy cost. To the best of the authors' knowledge, this is the first HEVC-to-AV1 transcoding solution, which hinders direct comparison with related works

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and in part by the Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul, Brasil (FAPERGS).

REFERENCES

[1] G. J. Sullivan, et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, 2012.

[2] T. Wiegand, et al., "Overview of the H.264/AVC Video Coding Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, 2003.

[3] T. Vaughan, "HEVC Advance Reduces Proposed License Fees," December 2015, X265, online, Available on http://x265.org/hevc-advance-reduces-proposed-license-fees/, accessed on January 20, 2019.

[4] Y. Chen, et. al., "An overview of core coding tools in the AV1 video codec," in 2018 Picture Coding Symposium (PCS), San Francisco, 2018.

[5] P. de Rivaz, J. Haughton. "AV1 bitstream & decoding process specification," The Alliance for Open Media, 2018, online, Available on https://aomediacodec.github.io/av1-spec/, accessed on January 20, 2019.

[6] P. Akyazi, T. Ebrahimi, "Comparison of Compression Efficiency Between HEVC/H.265, VP9 And AV1 Based on Subjective Quality Assessments," in: Twentieth International Conference on Quality of Multimedia Experience (QOMex), 2018.

[7] A. S. Dias, et al. "An Overview of Recent Video Coding Developments in MPEG and AOMedia," International Broadcasting Convention (IBC), 2018. Available on https://www.ibc.org/content-management/an-overview-of-recent-video-coding-developments-in-mpeg-and-av1/3303.article, accessed on January 20, 2019.

[8] E. De La Torre, R. Rodriguez-Sanchez, J. L. Martínez, "Fast video transcoding from HEVC to VP9," in: IEEE Transactions on Consumer Electronics, v.61, n.3, p.336–343, 2015.

[9] Y. Chen, Y. Zhou, and J. Wen, "Efficient Software HEVC to AVS2 Transcoding," in: Information, vol. 7, no. 3, p. 53, Sep. 2016. doi: 10.3390/info7030053

[10] I. Liapin, "Fast H.264/H.265 to AV1 stream transcoding using a moving object tracker," in: 2018 International Symposium on Consumer Technologies (ISCT), St. Petersburg, 2018, pp. 9-13. DOI: 10.1109/ISCE.2018.8408927.

[11] XIPH, "Xiph.org Video Test Media," 2018, Available on https://media.xiph.org/video/derf/, accessed on January 20, 2019.

[12] F. Bossen, et al., "Common test conditions and software reference configurations," Geneva, Switzerland: 12th Meeting VCEG (JCTVC-L1100), v. 12, 2013

[13] AOM. "AV1 codec library," 2018. Available on https://aomedia.googlesource.com/aom/, accessed on January 20, 2019. Downloaded on November 20, 2018, hash code e33e1243bd0ea470a60ac43a0125321d019259f3.

[14] T. Daede, A. Norkin, I. Brailovskiy, "Video Codec Testing and Quality Measurement," available on https://tools.ietf.org/html/draft-ietf-netvc-testing-07, accessed on January 20, 2019.