Complexity Scalable HEVC-to-AV1 Transcoding Based on Coding Tree Depth Inheritance

Alex Borges, Bruno Zatt, Marcelo Porto, Luciano Agostini, Guilherme Correa
Video Technology Research Group (ViTech)
Computer Science Graduate Program (PPGC) – Federal University of Pelotas (UFPel)
Pelotas, Brazil
{amborges, zatt, porto, agostini, 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 time and high energy consumption. This paper proposes a complexity scalable HEVC-to-AV1 transcoding solution with three operation modes, which allow adjusting the tradeoff between encoding efficiency and computational cost. The solution is based on the high correlation between block size decisions in HEVC and AV1, allowing the AV1 encoder to inherit Coding Tree depth information from the HEVC bitstream to constrain the AV1 re-encoding process. Experimental results for the three operation modes show a transcoding time reduction between 35.4% and 69.5% at the cost of a compression efficiency loss that varies between 4.9% and 16.8%.**

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

I. INTRODUCTION

High Efficiency Video Coding (HEVC) standard [1] was launched in 2013 and is gradually replacing its predecessor H.264/AVC [2] in most applications and multimedia-capable devices. However, HEVC has a royalty distribution policy [3] that incurs in high costs for commercial use, especially for video streaming companies. Thus, the Alliance for Open Media (AOMedia) was founded in 2015 with the goal of developing royalty-free video codecs. In 2018, the AOMedia Video 1 (AV1) [4,5] was launched as the first open-source, royalty-free video codec of AOMedia, with the goal of competing with HEVC and its predecessors.

As the company members of AOMedia slowly announce their adoption to this new video coding format, there is an increasing interest in AV1 from both industry and academy. During the forthcoming AV1 adoption, such companies will need to update their HEVC-encoded bitstreams to the new format in order to decrease royalty costs and to provide content compatible with a broad variety of users and platforms. End users may also choose to re-encode their personal videos with video transcoding, aiming at reducing storage requirements, especially for Ultra High Definition (UHD) content.

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 (although the measurements lead to different numbers due to distinct experimental setups). For example, the authors in [6] show that the runtime of the current AV1 reference encoder is up to 9 times higher than that of the x265, an HEVC open-source encoder, whereas [7] shows that the AV1 reference encoder is up to 106 times more complex than the HM, 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 between formats. This way, developing strategies for accelerating the HEVC-to-AV1 transcoding becomes mandatory.

Even though several previous works [8-10] aim at decreasing the complexity of transcoding from HEVC to other different bitstream formats, to the best of authors' knowledge there are no solutions published yet for the HEVC-to-AV1 transcoding process. The work [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 a complexity scalable 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. The number of partitioning possibilities allowed for testing in the AV1 re-encoding is used to adjust the complexity scalability level according to the system's or user's requirements. 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

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

other in several aspects regarding 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 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.

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 as 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 (64×64 CUs) and DL4 (8×8 CUs), whereas in AV1 it ranges between DL0 (128×128 blocks) and DL5 (4×4 blocks). Table I summarizes all the allowed CU/block sizes in both standards and their corresponding DL in the coding tree.

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 video sequences from the XIPH database [11] (*Boat, BoxingPractice, DinnerScene, Narrator, RitualDance, ToddlerFountain, TunnelFlag, and WindAndNature*), which were encoded by HEVC and by 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 lowest recommended QP in the Common Test Conditions (CTC) [12]. In AOM, the Constant Quality (CQ) parameter was set to 20 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 and the AV1 encoding process, the DL observed for each 4×4-pixel region of the same video was

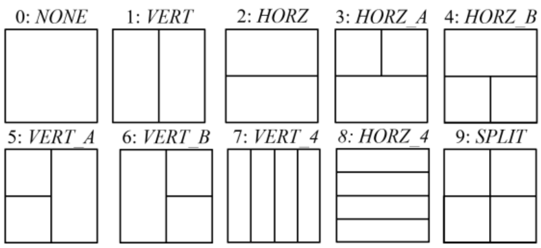

Fig. 1. Partition modes allowed in AV1

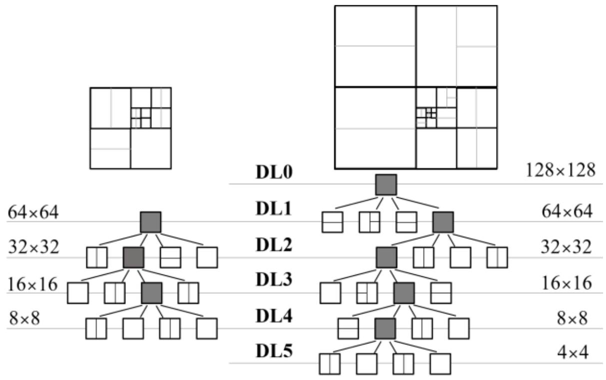

Fig. 2. Example of HEVC CTU (left) and AV1 SB (right) divided into CUs and blocks.

TABLE I. HEVC CU AND AV1 BLOCK SIZES ALLOWED IN EACH DL

DL	HEVC CU	AV1 Block Sizes
0	-	128×128, 128×64, 64×128
1	64×64	64×64, 64×32, 32×64, 64×16, 16×64
2	32×32	32×32, 32×16, 16×32, 32×8, 8×32
3	16×16	16×16, 16×8, 8×16, 16×4, 4×16
4	8×8	8×8, 8×4, 4×8
5	-	4×4

TABLE II. 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

stored for comparison. Table II 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 encoded either in DL1 (15.54%), DL2 (58.09%) or DL3 (19.21%) in AV1. These results provide important insight into the complexity reduction proposal presented in the next section.

III. SCALABLE HEVC-TO-AV1 TRANSCODER

Based on the correlation analysis presented in the previous section, a complexity scalable HEVC-to-AV1 transcoding solution is proposed. The solution is based on the idea of inheriting the decoded coding tree depth information from the

HEVC bitstream to infer decisions taken during the AV1 re-encoding process, avoiding the test of all possible depths.

During the transcoding from HEVC to AV1, a block partitioning map for every CTU decoded from the HEVC bitstream, named here as DL_{map}, it is stored. As the smallest 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 target level of transcoding complexity, which is an external parameter (Target Complexity – TC) controlled by the system or by user preference.

As previously 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 a few depths above or under the observed in DL_{map}, according to the Target Complexity. The distance allowed from DL_{map} is defined by values X (number of levels above DL_{map} will be evaluated) and Y (number of levels below DL_{map} will be evaluated) derived from TC, as summarized in Table III. For example, if TC=2 then X=1 and Y=1. Thus, AV1 is allowed to choose any DL between $DL_{map}-1$ until $DL_{map}+1$. For TC equal 1, the value 4 in the X variable means that if the DL_{map} is 4 (CU equals 8×8) the first conditional ($DL_n > DL_{map} - X$) will always be false, which does not activate the AV1 acceleration algorithm at depths above DL_{map}. It is important to observe that AOM has a safety process to find a partition structure better than estimated by statistical analyzes. Eventually this safety process will be activated and evaluated non-allowed DLs. This behavior is observed more frequently for TC=3 since it is the most restrictive one.

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, as well as the external parameter TC. 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}. From $DL_{map}–X$ to $DL_{map}+Y$ any depths may be chosen, and 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 or the current block is recursively split into four square blocks after incrementing DL_n. Otherwise, two scenarios can occur: either the current DL is above $DL_{map}–X$ or below $DL_{map}+Y$. In the first case, the 10 partition modes are also tested in a simplified prediction scheme for complexity reduction, but only partition mode 9 (*SPLIT*) can be chosen. The simplified intra/inter prediction is performed limiting to the only first prediction mode selected by AOM in its internal code. 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, forcing the AV1 encoder to choose the best mode found.

IV. Experimental Results

This section first presents the experimental setup to evaluate the scalable transcoding proposal and then presents

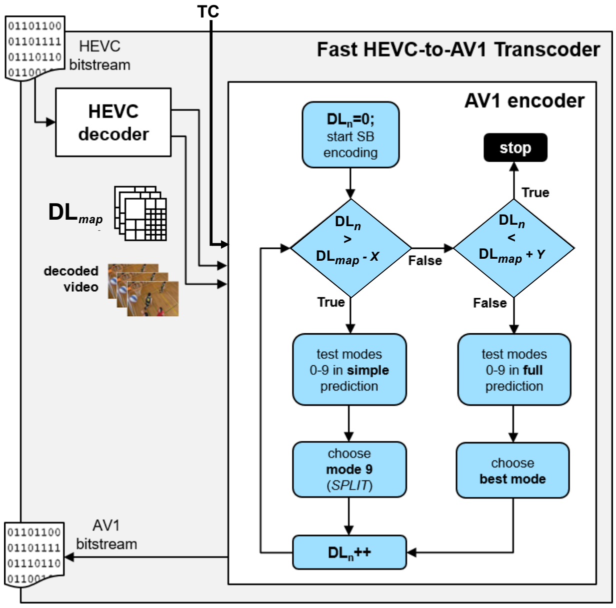

Fig. 3. Algorithm flow for the proposed solution.

TABLE III. Target Complexity Levels Used

TC value	X	Y	Allowed Depth Levels
1	4	0	From DL 0 until DL_{map}
2	1	1	From one DL above until one DL under DL_{map}
3	0	0	Equal to DL_{map}

the obtained results in terms of compression efficiency and computational cost for each target complexity level.

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. The Random Access Main configuration was chosen because it is most similar to the frame structure used by AV1.

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_{map}. 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-encode 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 compare different encoders or versions of the same encoder. To provide a Random Access-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

TABLE IV. COMPRESSION EFFICIENCY AND TIME SAVINGS RESULTS FOR THE THREE PROPOSED TARGET COMPLEXITY LEVELS IN COMPARISON TO THE ORIGINAL HEVC-TO-AV1 TRANSCODER.

	Video Sequence	Target Complexity = 1			Target Complexity = 2			Target Complexity = 3		
		BD-rate (%)	TS (%)	Ratio	BD-rate (%)	TS (%)	Ratio	BD-rate (%)	TS (%)	Ratio
HD1080 (1920×1080)	*DucksTakeOff*	0.3208	8.30	0.039	2.4469	50.31	0.049	7.8140	77.35	0.101
	Netflix_Boat	5.5245	21.25	0.260	3.9970	38.77	0.103	10.9816	72.16	0.152
	Netflix_FoodMarket	4.1710	19.97	0.209	5.6108	34.02	0.165	18.7696	69.30	0.271
	Netflix_SquareAndTimeLapse	5.6905	16.75	0.340	6.2268	38.41	0.162	15.1809	70.39	0.216
	Netflix_TunnelFlag	2.7987	20.59	0.136	4.5439	38.03	0.119	17.5075	68.41	0.256
	RushHour	2.4813	12.95	0.192	2.6238	35.41	0.074	15.1066	70.12	0.215
	SeaPlaneHDRAmazon	5.5123	28.44	0.194	6.0557	33.92	0.179	19.2815	70.46	0.274
UHD4K (4096×2160)	*Netflix_BarScene*	-1.3993	17.02	-0.082	7.5995	36.75	0.207	34.1441	63.64	0.537
	Netflix_BoxingPractice	5.0477	10.19	0.495	6.3069	21.42	0.294	20.9517	63.00	0.333
	Netflix_Narrator	2.2509	6.14	0.367	4.5629	26.01	0.175	16.1923	65.15	0.249
	Netflix_RitualDance	2.2608	10.85	0.208	4.9895	26.90	0.185	23.7196	65.56	0.362
	Netflix_ToddlerFountain	1.9522	9.89	0.197	1.5973	46.71	0.034	8.4869	75.79	0.112
	Netflix_WindAndNature	6.2693	11.70	0.536	5.4149	28.16	0.192	20.2780	66.63	0.304
	StreetHDRAmazon	1.4675	15.37	0.095	1.6608	41.04	0.040	6.6731	74.62	0.089
AVERAGE (HD1080)		**3.7856**	**18.32**	**0.196**	**4.5007**	**38.41**	**0.122**	**14.9488**	**71.17**	**0.212**
AVERAGE (UHD4K)		**2.5499**	**11.59**	**0.260**	**5.3706**	**32.46**	**0.161**	**18.6351**	**67.77**	**0.284**
AVERAGE (all videos)		**3.1677**	**14.96**	**0.228**	**4.9357**	**35.43**	**0.141**	**16.7920**	**69.47**	**0.248**

per sample encoding mode. Thus, for each one of the 14 video sequences coded in HEVC, four transcoding processes were performed (one per CQ) for each target complexity level plus the original tandem transcoding, totalizing 224 video transcoding experiments.

B. Speedup and Compression Efficiency Results

Fig. 4 shows the partitions structure obtained for the first interframe obtained by HEVC (Fig. 4.a), using the original tandem transcoding (Fig 4.b), and using the proposed transcoder considering the three TCs present in this paper (Fig. 4.c-e). These partition structures consider the same 128×128 region of *Netflix_SquareAndTimeLapse* video sequence processed with CQ=20 and artificially colored according to the color legend annex to the Fig. 4. There is the same DL partitioning in many sub-regions in those figures. TC=3 (Fig. 4.e) has the similar DL partitioning observed in the HEVC (Fig. 4.a).

Table IV presents Bjøntegaard Delta (BD)-rate and time-saving (TS) results obtained when transcoding the video sequences with the proposed solution under the three target complexity levels, taking the original tandem transcoder as the reference for comparison. For TC=1, the experimental results show that an average transcoding time saving of 14.96% was achieved at the cost of a compression efficiency loss of 3.16%. For TC=2, it was possible to reduce transcoding time by 35.43% with a BD-rate increase of 4.93%. Finally, for TC=3 the transcoding time was reduced by 69.47% at the cost of a BD-rate increase of 16.8%.

Considering the BD-Rate, the best case obtained for TC=1 is for the UHD4K *Netflix_BarScene* video sequence, which achieved a BD-rate decrease of 1.3% and an encoding time reduction of 17%. On the other hand, this video is the worst case for TC=2 and TC=3, with BD-rate increases of 7.6% and 34.1%, respectively. The best case of BD-Rate for TC=2 and TC=3 is the UHD4K *Netflix_ToddlerFountain* video sequence, which presents a BD-rate increase of 1.5% and 8.4%, respectively, both significantly below the average observed for each TC.

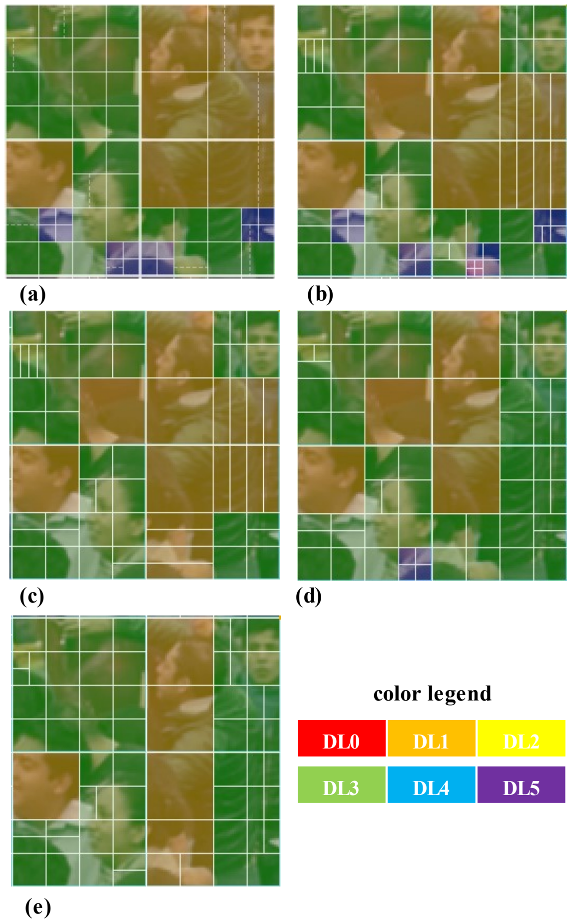

Fig. 4. DL for partitions chosen by (a) HEVC, (b) original AV1, (c) AV1 TC=1, (d) AV1 TC=2 and (e) AV1 TC=3.

Table IV also shows that the strategy always performs better for HD1080 than for UHD4K sequences. Even though very similar compression efficiency results are achieved for both classes, the ratio between BD-rate and TS is always better for HD1080 video sequences. For instance, the worst ratio obtained is for the video UHD4K *Netflix_BarScene* on TC=3 with 0.537% of BD-Rate increase for one percent of time saving. 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.

In order to better understand the behavior of the proposed transcoder, we have drawn the rate-distortion curves including (i) the original unmodified AV1 transcoder; (ii) the modified AV1 transcoders considering TC={1,2,3}; (iii) the HEVC encoder. The sequences with the average worst and average best BD-Rate/TS ratios of all TCs were selected according to Table IV: UHD4K *Netflix_BarScene* (Fig. 5) and UHD4K *StreetHDRAmazon* (Fig. 6) with 0.344% and 0.075% BD-Rate/TS ratios, respectively. One can notice that even for the worst case (Fig. 5) - where the modified transcoder with TC=3 leads to a BD-Rate increase of 34% in comparison with the tandem transcoder - the transcoder presents a compression efficiency much superior to HEVC along all quality points.

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 not possible.

V. Conclusion

This paper presented a complexity scalable 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 according to three different target complexity levels. When compared to the original, unmodified HEVC-to-AV1 transcoding flow, the transcoder achieves average time savings from 14.96% to 69.47%, with average compression efficiency losses between 3.17% and 16.79%. To the best of the authors' knowledge, this is the first HEVC-to-AV1 transcoding solution, which hinders direct comparison with related works.

Acknowledgments

This study 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 (CNPq), and the Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul, Brasil (FAPERGS).

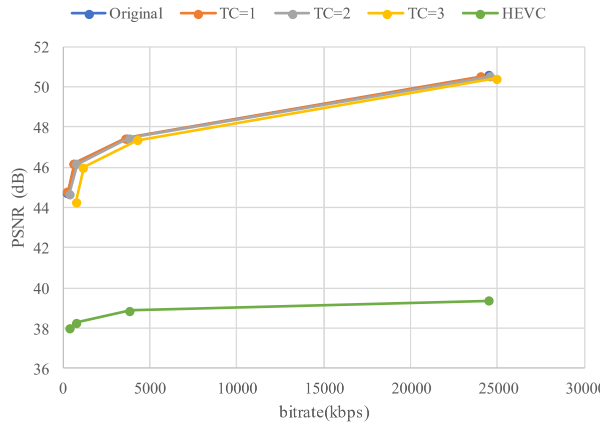

Fig. 5. Rate-distortion curve for UHD4K *Netflix_BarScene*.

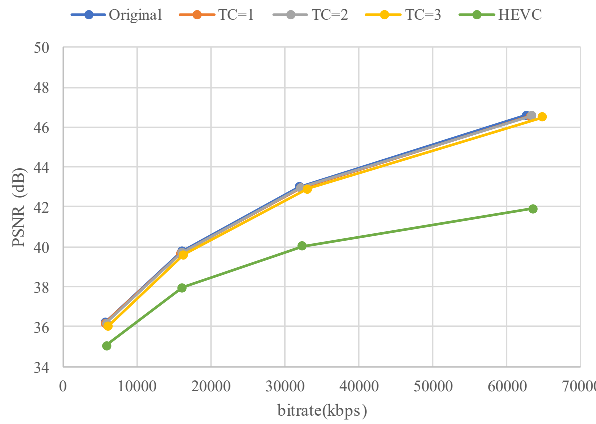

Fig. 6. Rate-distortion curve for UHD4K *SeaplaneHDRAmazon*.

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] M. A. Layek, et al. "Performance Analysis of AV1 for Video Coding in Very Low BitRates". In: Proceedings of the Korean Information Science Society, 2017. Available on http://www.dbpia.co.kr/Journal/ArticleDetail/NODE07207153, accessed on January 20, 2019.

[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". 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", 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.