Low-Complexity TZS Algorithm for Embedded Video Encoders

Cândido Moraes, Paulo Gonçalves, Marcelo Porto, Guilherme Correa
Video Technology Research Group (ViTech)
Federal University of Pelotas (UFPel)
Pelotas, Brazil
{csmoraes, phrgoncalves, porto, gcorrea}@inf.ufpel.edu.br

***Abstract*—Video consumption is significantly growing in the last years, especially in embedded systems. Researchers have been developing efficient compression algorithms and standards, such as the High Efficiency Video Coding (HEVC), but such efficiency has caused an augment on the complexity to encode videos. This is a serious problem especially in embedded systems, which present restrictions on processing and power consumption. Because of that, it is essential to develop effective solutions not only in terms of compression efficiency, but also in the sense of computational resource usage. This paper presents a solution by combining two strategies: an early termination for the Test Zone Search (TZS) algorithm, called e-TZS, and the Octagonal Raster Search Pattern (OARP), both of which aim at reducing the complexity of Motion Estimation, one of the most complex stages of modern video encoders. When combined and implemented in the HEVC encoder, the strategies allowed an average complexity reduction of 75.16% in TZS, with a negligible BD-Rate increase of only 0.1242%, in comparison to the original algorithm.**

Keywords—Embedded Devices, HEVC, Complexity Reduction, Motion Estimation, Test Zone Search.

I. INTRODUCTION

According to a Cisco report, by 2016 video traffic in mobile devices was responsible for 60% of the global mobile data consumption and the prediction is that by 2021 this number will grow to 78% [1]. The popularization of embedded systems and the emergence of the Internet of Things (IoT) demands each time more video broadcasting applications, some of them requiring real-time operation.

When working with embedded systems, some aspects need to be considered, like energy and data consumption. For example, as most embedded systems are fed by batteries, the autonomy of such devices depend on both efficient hardware and software working together. A less complex software will demand less processing requirements, which leads to improved battery autonomy. Data consumption is another challenge that must be considered when an embedded system is developed. Social networks with live streaming and security cameras connected to the internet are examples of applications that demand a massive amount of broadcasting data. However, mobile networks still present several limitations and often do not deliver data as fast as they should. Besides, most mobile internet service providers usually limit and block the user connection in case it surpasses a pre-specified data download/upload limit. Other non-functional requirements, such as restricted physical space and reduced price are intrinsic features in embedded systems which may also complicate the project. Therefore, it is essential that efficient tools to compress videos with small impact on image quality are developed.

Developed by the Joint Collaborative Team on Video Coding (JCT-VC), the High Efficiency Video Coding (HEVC) is the current state-of-the-art video coding standard [2]. HEVC is 50% more efficient in terms of compression efficiency than its predecessor, the H.264/AVC standard [3]. However, the coding process is up to five times more complex than H.264/AVC [4]. Although mobile embedded devices such as smartphones and tablets usually perform decoding operations more often than encoding, the popularization of live streaming and video call services in such platforms currently requires the investigation of complexity reduction strategies for the HEVC encoding.

HEVC introduced a much more flexible partitioning scheme, which allows both very large (64×64) and very small (4×4) block sizes. During the encoding process, each frame is first divided into an equal-sized sequence of Coding Tree Units (CTUs), which are typically 64×64 blocks. A CTU can be further recursively split into smaller blocks, called Coding Units (CUs) in a quadtree structure. These CUs are then further divided into Prediction Units (PUs) and Transformation Units (TUs) for prediction and transform purposes, respectively.

The HEVC encoding process is based on the same hybrid scheme of its predecessors, composed mainly of intra/inter-picture predictions and 2D transform [2]. The intra-picture prediction explores spatial similarity between neighboring blocks within the same image, while the inter-picture prediction explores the temporal similarity between neighboring images of a video sequence. The difference between the original block and its prediction is called residual signal. This residue is exposed to a linear spatial transformation, quantized and entropy coded before transmission or storage as a compressed bitstream. The generated bitstream is then used at the decoder to recompose the original video.

One of the most complex stages of the HEVC encoder is the Motion Estimation (ME), which is part of the inter-picture prediction. The ME purpose is to find vectors representing an estimated motion of a block in an image belonging to a sequence. The Full Search (FS) is one of many algorithms available to solve the ME, but it is the most complex solution because it compares the current block with all the candidates blocks within a search area, achieving the best result possible. However, in practice the FS solution cannot be used because of its high complexity. Test Zone Search (TZS) is the current state-of-the-art ME algorithm, chosen to be implemented in the HEVC reference encoder during the standardization process [5]. However, despite being considered a fast ME approach, TZS is still one of the most demanding parts of the HEVC encoding process.

This paper combines two different strategies to reduce the TZS complexity with few losses in compression efficiency, aiming at a low-complexity solution that is aligned with embedded system requirements. The algorithms used in the combined solution are e-TZS and OARP, both of which are discussed in the next sections of this paper [6-7]. When implemented in the reference HEVC encoder, the combined solution is able to reduce TZS complexity by 75.16%, which results in an overall encoding complexity reduction of 16.18% and a negligible Bjøntegaard Delta (BD)-rate [8] increase of only 0.1242%.

II. The TZS Algorithm and Related Works

Test Zone Search (TZS) is a fast ME algorithm that presents a coding efficiency close to the achieved in FS, requiring less computational resources [5]. TZS is composed of four steps: (1) Motion Vector Prediction, (2) First Search, (3) Raster Search, and (4) Refinement, as shown in Fig. 1.

In the first step, Motion Vector Prediction, also simply referred as Prediction, TZS looks at the Motion Vectors (MVs) of neighbor PUs and selects the one that yields the lowest rate-distortion (RD) cost. The analyzed PUs in the neighborhood are the median, co-located, left, top and top-right PUs. The PU chosen in the Prediction step is the central point of a diamond or a square search pattern employed in the second step, called First Search. This pattern is expanded in powers of 2 until a pre-defined search range is reached. The best point is again the one with lowest RD cost. If after three expansions no point becomes better than the central point, the execution of this step is interrupted.

The third and most complex step is the Raster Search. In this step, a sub-sampled full search is performed in a limited search area. If the distance between the center of the search area and the best block found at the previous step is smaller than *iRaster* (a constant equal to 5 by default), the Raster Search step is skipped. The sub-sampling of *iRaster* is performed both horizontally and vertically, resulting in a total of n_{points} candidates blocks to be compared, defined in (1). For a search range (SR) equals to 64, i.e., with a span of [-64,+64] positions, a total of 676 candidates blocks are compared. Due to the number of comparisons performed, this is the most computationally expensive step of the TZS algorithm.

$$n_{points} = \left\lceil \frac{1 + (SR \times 2)}{iRaster} \right\rceil^2 \qquad (1)$$

The fourth and final step of TZS is the Refinement. A new search identical to that performed in the first step occurs around the best position found in the previous step. If after two expansion levels no block with smaller RD cost is found, the Refinement step is stopped.

To reduce the complexity of TZS and speed up the overall encoding time, the authors in [9-10] propose techniques by decreasing the number of search points through dynamic search range and fast mode decision algorithms. These approaches can provide incremental gains in time reduction but incur in significant losses in compression efficiency.

We have proposed two approaches for further reducing TZS complexity in previous works, which presented a significant TZS processing time reduction and achieved high compression efficiency. The first approach, called e-TZS [6], is an early-termination scheme based on multiple decision trees, which were trained in an extensive process of data mining and machine learning techniques. The approach seeks to identify TZS steps in which the best block matching has higher chances to be found and then bypass the remaining steps. The second approach, called Octagonal-Axis Raster Pattern (OARP) [7], consists of a search pattern for the Raster Search step that exploits the regions in the search area with the highest probabilities of finding the best block matchings.

III. Early-Termination for TZS

This section presents the multiple early-termination scheme for TZS, named as e-TZS [6]. The scheme is based on a set of decision tree models that are executed after each step of TZS to decide whether or not the execution should be halted. An analysis of TZS presented by [6] showed that 87% of the best MVs are found after the Prediction Step, whereas only 11% are found after First Search. Raster step finds just 0.4% of the best MVs and Refinement finds 1.6%. A time analysis was also performed in [6] and showed that the Raster step took 81% of the total TZS processing time, even though it finds the smaller amount of best MVs. On the other hand, the Prediction and First Search steps represent only 2% and 10% of the total TZS time, respectively. Therefore, in the analyzed cases the cost of processing some steps of TZS is wasted, mainly for the Raster Search, since the major amount of best block matchings are found in the first two steps. This way, by identifying those cases, the final steps could be bypassed, with small penalties in coding efficiency.

The e-TZS algorithm proposed in [6] tries to predict the TZS stage in which the best block matching will be found, thus skipping posterior steps in an early termination approach. The predictions are performed based on a set of decision tree models trained after an extensive data mining process. The models were trained using the C4.5 algorithm implemented by the open-source and free software WEKA (Waikato

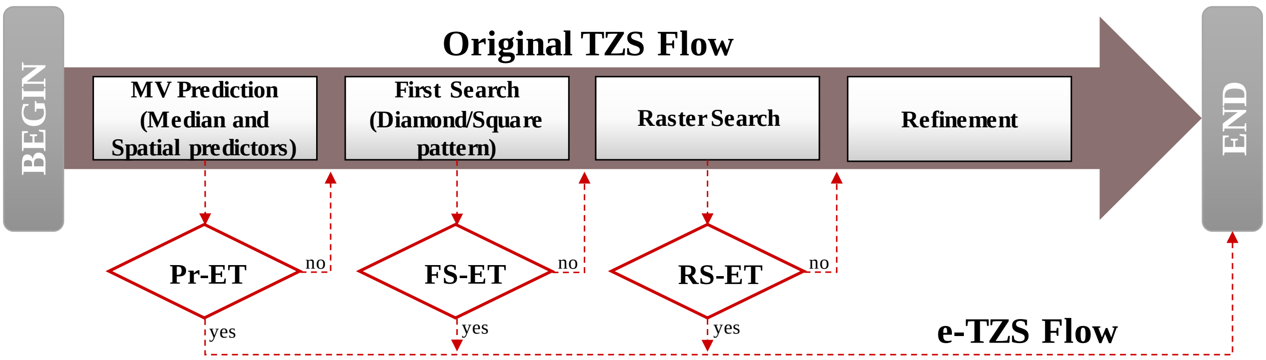


Fig 1: Flowchart of TZS and e-TZS algorithms.

Environment for Knowledge Analysis) [11]. More than 40 parameters were collected during the encoding process, generating 600,000 instances. The maximum depth of the obtained decision trees is equal to eight, which means that they are easily implemented without significant processing overhead. The red arrows in Fig. 1 present the flow of the e-TZS algorithm and show that the decision tree models are run after Prediction, First Search and Raster Search steps.

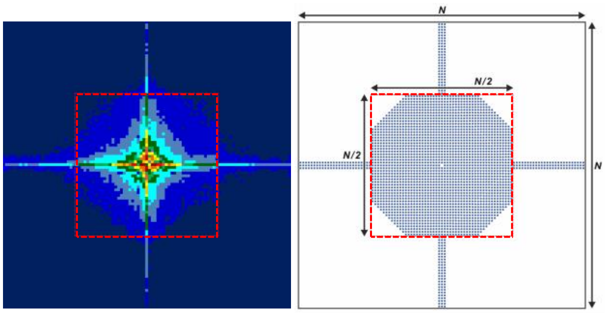
The e-TZS scheme proposed in [6] was implemented in the HEVC reference software (HM-16.14). In comparison with the original TZS, it achieved an average time reduction of 62.53% for TZS and 13.53% for the complete encoding process. A compression efficiency loss of 0.49%, measured in terms of Bjøntegaard Delta (BD-Rate) [8], was noticed when e-TZS is applied. The models achieved a precision of 94.2% on average, meaning that e-TZS took the right decision of stopping or continuing the search process in most of the tests. It is important to emphasize that the videos used to train the decision trees are different from those used in the experiments.

IV. Octagonal-Axis Raster Pattern

The Octagonal-Axis Raster Pattern (OARP) algorithm [7] is a search pattern developed specifically for the Raster Search, which is the most complex step of TZS. A distribution analysis of the best block matchings founds during the Raster Search step was performed in [7] and revealed a significant concentration of the most best block matchings at the center and axis of the search area. Fig. 2(a) shows the average distribution for the whole set of video sequences plotted as a heatmap, where the warmest colors represent search area positions with larger occurrence of best block matchings and the coolest colors represent positions rarely or never chosen.

Therefore, OARP optimizes the search area, reducing the number of points to 25% of the original by looking at the regions with higher probability of best matchings. The search area proposed by the OARP algorithm is represented in Fig. 2(b), in which 88% of the proposed search area is located within an octagonal central region, and 12% correspond to the horizontal and vertical axis. Considering the average distribution presented in Fig. 2(a), OARP is capable of covering 62.3% of the total best block matchings found in the original Raster Search.

To evaluate compression efficiency, OARP was implemented in the HEVC reference software (HM-16.14) in [7]. A time reduction of 60.91% was achieved in TZS, which led to a time reduction of 21.57% in the total encoding time, with a BD-Rate [8] increase of only 0.04%.



(a) (b)

Fig. 2: (a) Average block matching distribution for the whole set of video sequences tested [7]; (b) Octagonal-Axis Raster Pattern (OARP) [7].

V. Results for Combined Scheme

Even though e-TZS is able to skip some steps of TZS by employing a machine learning approach, the most computationally demanding step (Raster Search) cannot be always skipped when e-TZS is used. In its turn, OARP tries to optimize Raster Search by reducing the SA. By unifying these two solutions, this paper proposes a combined scheme to achieve more significant results in terms of complexity reduction by avoiding Raster Search as much as possible and, when necessary, by trying to do it more quickly.

The proposed combined scheme was also implemented in the HEVC reference software (HM-16.14). Search range was configured as 256, with maximum CU size of 64×64 and maximum CU partition depth equals to 4, as recommended in the Common Test Conditions (CTC) [12]. The video sequences were encoded with the Random Access temporal configuration and with Quantization Parameters (QPs) 22, 27, 32 and 37. All simulations have been carried on a workstation with an Ubuntu 14.04.5 OS, running on an Intel Xeon E5-2640v3@2.60GHz processor and with 32 GB of RAM.

Altogether, nine high-resolution video sequences are fully encoded: four from the A class of the CTC (*Traffic*, *PeopleOnStreet*, *NebutaFestival*, *SteamLocomotiveTrain*) with WQXGA resolution (2560×1600 pixels), and five from the B class of the CTC (*Kimono*, *ParkScene*, *Cactus*, *BQTerrace*, *BasketballDrive*) with Full HD resolution (1920×1080 pixels). It is important to emphasize that these sequences differ in frame rate, bit depth, spatial resolution and motion/texture content.

Table I shows the experimental results for the combined e-TZS and OARP schemes, as well as their individual results. As the analyzed videos sequences were not the same in e-TZS [6] and OARP [7] original papers, the full set of videos was tested once again in this work. The average results refer to the same videos sequences in the combined schemes and in the originals techniques. The proposed combined solution achieved an average time reduction (TR) of 75.16% in comparison with the original TZS. This value overcomes the original algorithms separately. Considering the entire encoding process, a time reduction of 16.18% was achieved, which is better than the results achieved by OARP and e-TZS algorithm. Compression efficiency results are presented in terms of BD-Rate [8], which is a metric used to compare the bitrate difference between two encoding solutions considering the same video quality. BD-Rate results present an increase of 0.1242% in bitrate, which is negligible when considering the achieved results in terms of time reduction.

By comparing the results in Table I, it is possible to notice that some videos presented better result than others. For example, *SteamLocomotiveTrain* presented the largest TR in the combined scheme, just like in the OARP algorithm. It happens because this sequence depends a lot on Raster Search and thus benefits from the strategy proposed in OARP. On the other hand, *BQTerrace* presents the worst results in TR in the combined scheme, just like in OARP, because it depends less Raster Search results. Thus, the results in this combined scheme are mostly influenced by Raster Search, which was expected because it is the most expensive step of TZS

Finally, Table II shows a comparison between this work and two related works found in the literature for complexity reduction of TZS. To allow for a fair comparison, the set of video sequences tested for the comparison in Table II is

TABLE I. PERFORMANCE OF E-TZS, OARP AND PROPOSED E-TZS + OARP COMBINED SCHEME.

Class	Sequence	BD-Rate (%)			Total TR (%)			TZS TR (%)		
		e-TZS [6]	OARP [7]	Proposed	e-TZS [6]	OARP [7]	Proposed	e-TZS [6]	OARP [7]	Proposed
Class B	*Kimono*	0.1510	-0.3004	0.1321	16.93	13.14	19.91	67.18	52.23	80.03
	ParkScene	0.1558	-0.1985	0.1268	8.43	5.89	10.32	58.67	38.18	69.54
	Cactus	0.1543	-0.2087	0.1434	14.02	12.73	17.86	59.18	53.65	78.61
	BQTerrace	0.1236	-0.0541	0.1570	6.77	3.86	7.18	57.74	28.90	63.65
	BasketballDrive	0.1015	-0.2654	0.1019	19.08	18.07	24.18	60.81	58.42	80.72
Class A	*Traffic*	0.3131	-0.1882	0.3466	7.23	3.62	8.26	61.47	30.37	69.85
	PeopleOnStreet	0.1869	-0.7118	0.0561	16.09	17.36	23.24	55.06	59.62	79.97
	NebutaFestival	0.0425	-0.0457	-0.0344	8.64	6.39	9.96	59.80	42.81	70.50
	SteamLocomotiveTrain	0.6568	-0.0245	0.0879	19.62	17.90	24.67	66.01	59.20	83.56
	Average	**0.2095**	**-0.2464**	**0.1242**	**12.98**	**11.00**	**16.18**	**60.66**	**47.05**	**75.16**

TABLE II. COMPARISON WITH RELATED WORKS.

Sequence	BD-Rate (%)			Total TR (%)		
	[9]	[10]	Proposed	[9]	[10]	Proposed
Kimono	1.91	1.51	0.13	26.49	27.11	19.91
ParkScene	1.07	0.74	0.13	18.69	19.18	10.32
Cactus	0.87	0.59	0.14	19.65	21.18	17.86
BQTerrace	0.45	0.29	0.16	19.85	20.04	7.18
BasketballDrive	2.50	2.15	0.10	23.62	26.66	24.18
Average	**1.36**	**1.06**	**0.13**	**21.60**	**22.85**	**15.89**

exactly the same as that listed in the two related works. In [9], several techniques based on search range adaptation are implemented to decrease the number of search points in the TZS process, whereas in [10] TZS is accelerated by decreasing the number of search points according to a spiral scan manner instead of a raster scan.

At first sight, it is possible to notice that both [9] and [10] achieve larger encoding time reductions than the scheme proposed in this work. However, these reductions come at the cost of much larger increases in BD-Rate [8], which means that compression efficiency is negatively affected. The ratio between BD-Rate and TR shows the loss in coding efficiency caused by each percentage point reduction in encoding time, which is a much fairer comparison metrics between different works. The obtained BD-Rate/TR ratio is equal to 0.063 for solution [9], 0.046 for solution [10] and 0.008 for the solution proposed in this work. Notice that the ratio achieved by this work is one order of magnitude smaller than the achieved in related works. Therefore, it is possible to conclude that the proposed scheme achieves the best tradeoff between encoding time reduction and compression efficiency when compared to related works.

VI. CONCLUSIONS

Although HEVC reaches high compression rates thanks to its advanced tools, the encoding process demands more processing power and time than its predecessor standards. The elevated complexity of HEVC impacts especially in battery consumption in embedded systems, which is an important challenge to be solved. This paper presented a combined scheme based on two previously proposed fast TZS algorithms with significant results in both complexity reduction and compression efficiency. The combined scheme aimed at achieving even better results to tackle the TZS complexity problem without implicating in significant compression efficiency losses. When combining the strategies, the original TZS encoding time was reduced to three quarters, with a negligible loss of 0.1242% in compression efficiency (in terms of BD-Rate [8]). Also, the proposed strategy can be jointly implemented with other approaches to achieve even more significant levels of complexity reduction.

The proposed combined solution can be implemented in embedded systems that require low complexity and high compression efficiency. By demanding less computational resources, extra hardware resources can be no longer necessary, thus decreasing the overall energy consumption, which is especially useful in battery-fed devices.

ACKNOWLEDGMENT

The authors would like to thank the National Council for Scientific and Technological Development (CNPq), the National Council for the Improvement of Higher Education (CAPES) and the Foundation for Research Support of Rio Grande do Sul State (FAPERGS) for the financial support, which allowed the development of this work.

REFERENCES

[1] CISCO. Global Mobile Data Traffic Forecast Update, 2016-2021. Cisco Visual Networking Index (VNI), 2017.

[2] G. J. Sullivan, J. Ohm, H. Woo-Jin, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1649-1668, 2012.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard” IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 560-576, 2003.

[4] G. Correa, P. Assuncao, L. Agostini, and L. 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, pp. 1899-1909, 2012.

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

[6] P. Goncalves, G. Correa, M. Porto, B. Zatt, L. Agostini, “Multiple Early-Termination Scheme for TZSearch Algorithm based on Data Mining and Decision Trees,” IEEE 19th International Workshop on Multimedia Signal Processing, 2017.

[7] P. Goncalves, M. Porto, B. Zatt, L. Agostini, G. Correa, “Octagonal-Axis Raster Pattern for Improved Test Zone Search Motion Estimation” IEEE International Conference on Acoustics, Speech and Signal Processing, 2018.

[8] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves" ed. Austin, Texas, 2001.

[9] P. Nalluri, L. N. Alves, and A. Navarro, “Complexity reduction methods for fast motion estimation in HEVC,” Signal Process.: Image Comm., vol. 39, no. A, pp. 280-292, Nov. 2015.

[10] R. Fan, Y. Zhang, and B. Li, "Motion Classification-based Fast Motion Estimation for High Efficiency Video Coding" IEEE Transactions on Multimedia, 2016.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update" SIGKDD Explor. Newsl., vol. 11, pp. 10-18, 2009.

[12] ISO/IEC-JCT1/SC29/WG11, "Common test conditions and software reference configurations" ed. Stockholm, Sweden, 2012.