Complexity-Aware TZS Algorithm for High Efficiency Video Coding

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

Abstract—**Video applications are significantly growing in the last years, especially in embedded/mobile systems. Modern video compression algorithms and standards, such as the High Efficiency Video Coding (HEVC), achieved a high efficiency in compression ratio. However, such efficiency has caused an augment on the complexity to encode videos. This is a serious problem especially in mobile systems, which present restrictions on processing and energy consumption. This paper presents an enhanced Test Zone Search (TZS) algorithm, aiming at complexity reduction of the Motion Estimation (ME) process in the HEVC standard and focusing efficient hardware design for mobile encoders. The proposed algorithm is composed of two strategies: an early termination scheme for TZS, called e-TZS, and the Octagonal-Axis Raster Search Pattern (OARP). When combined and implemented in the HEVC reference 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. Besides, the approach presents an average block matching operation reduction of 80%, allowing hardware simplification and decreasing memory access.**

Index Terms—**Complexity Reduction, HEVC, Motion Estimation, Test Zone Search, Embedded Devices.**

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.

The embedded systems, by its own characteristics, demands some additional concerns like energy and data consumption. Batteries feed most embedded systems and the autonomy of such devices depend on both efficient hardware and software working together. A less complex software will demand less processing requirements, which could lead to an improved battery autonomy. Data consumption is another challenge that must be considered. Social networks with live streaming and security cameras connected to the Internet are examples of applications that demand a massive amount of data to be broadcasted, which has been increasing since many mobile devices can record videos on higher resolutions such as FULL HD (1920x1080) and 4K (3840x2160). However, mobile networks sometimes can be unstable. Besides, most mobile Internet service providers usually limit and block the user connection in case it surpasses a pre-specified data download/upload limit. Therefore, it is essential to develop efficient tools to compress videos with small impact on image quality and low overhead in the computational resources to allow the usage in embedded systems and improve the user experience by using these devices.

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 40-50% more efficient in terms of compression efficiency, using half of the bits needed to encode a video in comparison with its predecessor, the H.264/AVC standard, with the same video quality [2]. However, the coding process is up to five times more complex than the H.264/AVC in terms of time spent to encode a video sequence [3]. Much of this is due to the more flexible partitioning scheme implemented on HEVC, which allows both very large (64×64) and very small (4×4) block sizes. During the encoding process, the HEVC encoder divide each frame in an equal-sized block called Coding Tree Units (CTU). The default standard size of a CTU is 64x64, although the sizes 16x16 and 32x32 are also available. A CTU can be further recursively split into smaller blocks, called Coding Units (CUs) in a quad-tree structure. These CUs are then further divided into asymmetric blocks called Prediction Units (PUs) and Transformation Units (TUs) for prediction and transform proposes, 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.

The stage responsible for most of the compression gains in the HEVC encoder is the Motion Compensation (MC), which is used in inter-picture prediction. Part of the MC stage, the Motion Estimation (ME) is used to find vectors representing an estimated motion of a block in an image belonging to a sequence. To perform this, ME compares the blocks in the current frame with a reference frame previously encoded. Therefore, instead of transmitting or storing blocks with redundant data, the ME process allows encoding only the displacement of a block in the current frame to be reconstructed on the decoder side.

Composed by two steps, the ME is one of the most complex stages in the HEVC standard. In the first step, the Integer Motion Estimation (IME), a block matching algorithm

Digital Object Identifier 10.29292/jics.v14i3.60

(BMA) is used with the support of a matching criteria function like the Sum of Absolute Differences (SAD) and Rate Distortion (RD). The fractional motion estimation (FME) is the second and last step of the ME and it consist in an interpolation process around the best IME result. The gains obtained in this work are in the IME scope.

The Full Search (FS) among many BMAs available to solve the IME is the one that achieves the best result possible in terms of video quality and bitrate saving, but it is also the most complex solution because it compares the current block with all the candidates blocks within a search area. Therefore, in practice the FS solution cannot be used because of its high complexity. Test Zone Search (TZS) is the current state-of-the-art BMA, chosen to be implemented in the HEVC reference encoder during the standardization process [4]. However, despite being considered a fast IME approach, TZS is still one of the most demanding parts of the HEVC encoding process. 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.

This paper combines two different strategies to the TZS complexity reduction with negligible 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 the e-TZS [5], an early-termination scheme based on multiple decision trees, that seeks to identify TZS steps in which the best block matching has higher chances to be found and then bypass the remaining steps; and the OARP [6], that consist of a search pattern for the Raster Search step that exploits the regions with the highest probabilities of finding the best block matchings both of which are discussed in the next sections of this paper. When implemented in the reference HEVC encoder, the combined solution is able to reduce the time necessary to perform the IME process by 75.16%, which results in an overall time reduction of 16.18% and a negligible Bjøntegaard Delta (BD)-rate [7] increase of only 0.1242%.

A discussion of hardware usage and energy consumption savings of the combined approaches is presented. The obtained result demonstrates a decrease of 80% in the block matching operations, which has a direct impact for energy saving and external memory accesses in a hardware implementation.

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 [4]. TZS is a combination of four steps: (1) Motion Vector Prediction, (2) First Search, (3) Raster Search and (4) Refinement, as shown in Fig. 1.

A. Description of TZS algorithm

In the first step in Fig. 1, Motion Vector Prediction, also simply referred as Prediction, TZS algorithm employs four predictors: the left, up, and upper right predictors, as shows Fig. 2, ones to compute the median predictor of the corresponding block. The median is computed using the equation (1).

$$\begin{aligned} Median(A,B,C) = A + B + C \\ - Min(A, Min(B,C)) \\ - Max(A, Max(B,C)) \end{aligned} \quad (1)$$

The best predictor is chosen as the one that leads to lowest block matching criteria, usually using Sum of Absolute Difference (SAD), because SAD is the simplest-commonly used method among various matching criteria. SAD is defined in (2), where C represents the current block, R denotes the reference block, with block size of M×N pixels and (x, y) and (s, r) denote motion vector coordinates of reference and current blocks, respectively. The predictor with the smallest SAD value is selected as the starting point for the next algorithm step.

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(s+i, r+j) - R(x+i, y+j)| \quad (2)$$

After selecting the start position in the last step, the second step in Fig. 1 perform the first search around the Search

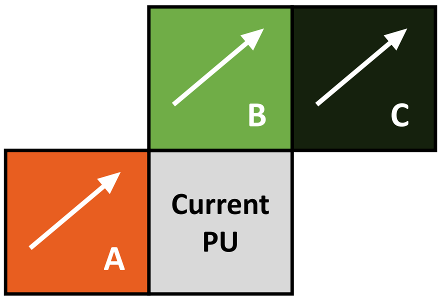


Fig. 2. Motion vectors adjacent to the current PU.

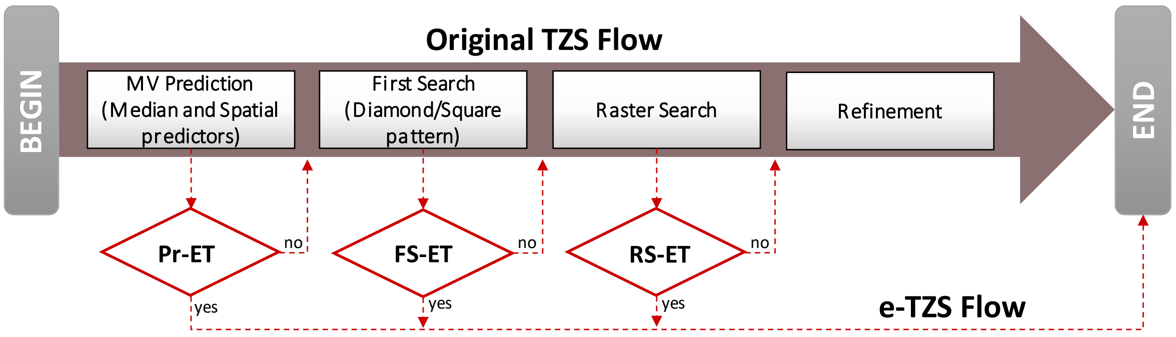
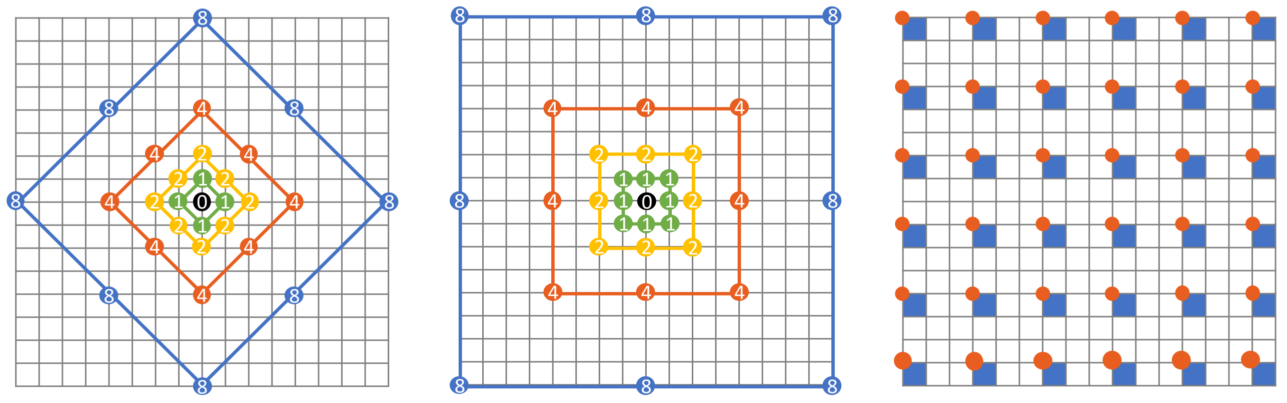

Original TZS Flow
BEGIN
MV Prediction (Median and Spatial predictors)
First Search (Diamond/Square pattern)
Raster Search
Refinement
END
Pr-ET
FS-ET
RS-ET
no
yes
e-TZS Flow

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



(a) (b) (c)

Fig. 3. Search patterns: (a) Diamond Search, (b) Square Search, (c) Raster Search.

Window (SW), in which a diamond or square search pattern could be executed as shown in Fig. 3(a-b).

This pattern is expanded in powers of 2 until a pre-defined limit of the SW is reached. The best point is again the one with lowest SAD. The distance obtained for the best point is stored in the "*bestdistance*" variable, which will be evaluated in further steps. If after three expansions no point becomes better than central point, the execution of this step is interrupted.

The third step in Fig. 1 is a sub-sampled full search, performed in a limited Search Range (SR). Raster Search is the most complex step of the TZS algorithm, taking 81% of the total TZS execution time [5]. The Raster Search is performed when the difference between the motion vectors obtained from the previous step and the starting position is too large. By default, if the *bestdistance* stored in the previous step is smaller than *iRaster* (a constant equal to 5 by default), the Raster Search is skipped. The *iRaster* also define the sub-sampling factor performed both horizontally and vertically in the Raster Search, resulting in a total of n_{points} candidates blocks to be compared, defined in (3).

$$n_{points} = \left[\frac{1 + (SR \times 2)}{iRaster}\right]^2 \qquad (3)$$

Fig. 3(c) illustrate the Raster Search step performed to a SR equals to 8, i.e., with a span of [-8, +8] positions, where a total of 36 candidates blocks is compared. However, larger SR sizes are usually used, like 64, 128 or 256. Due to the exponential growth of the formula, this increase in candidate blocks compared makes Raster Search the most computationally expensive step of the TZS algorithm.

The Refinement step in Fig. 1 is the last step of TZS, and involves a greater accuracy of the motion vector obtained from the previous steps. 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 SAD is found, the Refinement step is stopped.

B. Related Works

In order to reduce the block matching complexity in ME algorithms, several authors purpose techniques to speed up this process maintaining the coding efficiency, with low complexity distortion measures, such as the Partial Sum of Absolute Differences (PSAD) [8], or using early-termination schemes [9]. Besides, also by modifying the search patterns in the ME process and decreasing the number of search points through dynamic search range, or proposing fast mode decision to pruning the partitioning structure, the authors in [10, 11] achieve greater encoding time reduce. However, the incremental gains in time reduction provided by some these approaches incur in significant losses in compression efficiency or, in other cases these gains are still small compared to the expressive complexity of TZS algorithm. Besides, due to the lack of regularity, most of these approaches turn out to be of difficult implementation in hardware.

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 [5], 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) [6], 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 [5]. The scheme is based on a set of decision tree models that are executed after each step of the TZS workflow to decide whether the execution should be halted. By predicting in which TZS stage, the best block matching will be found, the e-TZS algorithm could skip posterior steps of the original TZS workflow in an early termination approach.

An analysis of the TZS original algorithm performed by [5] showed that 87% of the best MVs are found after the Prediction Step, whereas only 11% are found after First Search. Raster Search step finds just 0.4% of the best MVs and Refinement finds 1.6%. A time analysis was also performed by [5] and showed that the Raster Search 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 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 Environment for Knowledge Analysis) [12]. More than 40 parameters were collected during the encoding process, generating 600,000 instances. To evaluate the relevance of each parameter in the model, the method Information Gain Attribute Evaluation (IGAE) from WEKA was used. The three decisions trees were trained with the J48 tool from WEKA with the same 600,000 instances after the definition of best attributes. The output of the model are two classes: “Yes”, means that the execution flow could stop in that moment, while “No” indicates that the next step should be performed. The model’s implementation is a sequence of nested IF-ELSE with the number of conditions tested are given by the depth of the tree. The maximum depth of the obtained decision trees is equal to eight, which means that they are easily implemented without significant processing overhead.

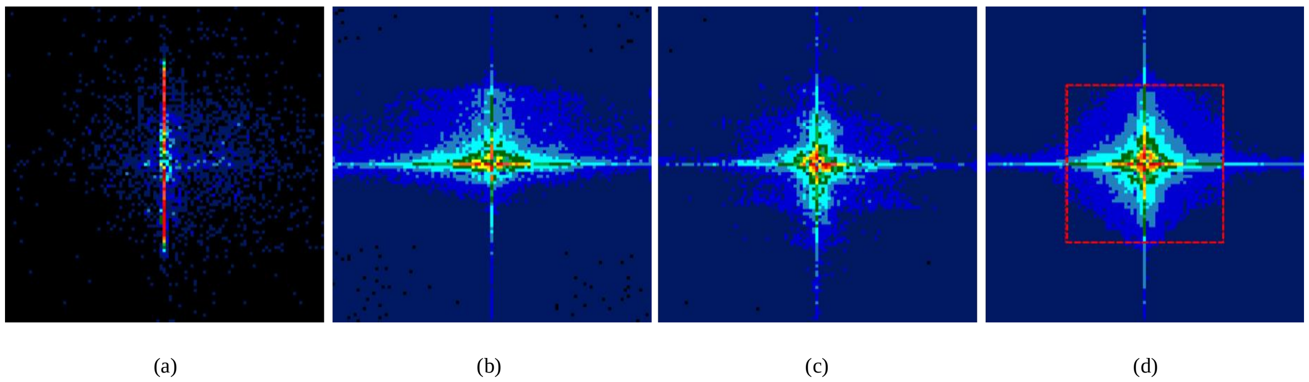
The red arrows in Fig. 1 present the flow of the e TZS algorithm and show that the three decision trees were placed after Prediction (Pr-ET), First Search (FS-ET) and Raster Search steps (RS-ET). This means that e-TZS is capable to terminate the TZS algorithm in three different stages. Notice that for Pr-ET only a couple of parameters are available, because only a step of TZS was performed. This implies in the decision tree with less accuracy among all. In the other hand, RS-ET is found after the execution of three steps of the TZS algorithm. Consequently, the accuracy of RS-ET is the largest of all, because it uses the information of all previous steps. Thus, the accuracy of Pr-ET, FS-ET and RS-ET is 72.4%, 98.4% and 98.6%, respectively.

Therefore, the average accuracy of e-TZS is of 89.8%, this means that the scheme is unable to decide correctly between terminating or keeping the execution of TZS in only 10.2%, on average. However, it is important to notice that this misclassification occurs in only 5.8% of the inaccurate decisions when TZS is terminated but should keep running. The remaining 4.4% of the inaccurate decisions do not cause any misclassification since the e-TZS indicates that TZS should keep running, but the best MV has already been found. These cases only incur in the lack of complexity reduction, with no penalties in coding efficiency. Due to these characteristics, the e-TZS scheme is capable of bypassing some TZS steps speed up the algorithm execution, reducing its complexity with negligible losses in coding efficiency, through an efficient exploration of the attributes extract from the data mining process.

IV. Octagonal-Axis Raster Pattern (OARP)

The Octagonal-Axis Raster Pattern (OARP) algorithm [6] is a search pattern developed specially for the Raster Search, which is the most complex step of TZS. A distribution analysis of the best block matchings found during the Raster Search step was performed in [6]. For this, during the execution of Raster Search step, all search points are within the Search Window (SW), which has a fixed maximum size defined by the Search Range (SR) parameter. The default value of SR in HM was used in the experiments, setting the SW to [-256, +256], thus composed of 512×512 positions. Therefore, considering the *iRaster* sub-sampling step, a total of 10,609 position were tested during the execution of Raster Search step.

For this distribution analysis, the point where the best block matching was found was stored after each execution of Raster Search. The results were plotted in the heat maps presented in Fig. 4, where the warmest colors represent regions of the SW with larger occurrence of best block matchings, and the coolest colors represent regions rarely or never chosen. Fig. 4(a)-(c) present this analysis for *BQTerrace*, *YachtRide* and *Cactus* video sequences, respectively. *BQTerrace* (Fig. 4(a)) and *YachtRide* (Fig. 4(b)) are corner cases that represents the two most uncommon distributions among all videos analyzed. In *BQTerrace*, a constant cam-


(a) (b) (c) (d)
Fig. 4. Block matching distribution within the [-256, +256] search area for (a) *BQTerrace*, (b) *YachtRide*, and (c) *Cactus* sequences; (d) average block matching distribution for the whole set of video sequences tested.

era movement in the upright direction is performed. This temporal characteristic makes perceptible a significant concentration in the central vertical axis of the SW. The *YachtRide* is a video sequence mainly compose of a boat floating from left to right, with some upright oscillations. Due to these characteristics, a uniform distribution is perceived, but the activity above the central horizontal axis is more accentuated than bellow. In *Cactus*, the distribution follows a pattern recurrent in most remaining sequences, with a clear concentration around the central point and along the horizontal and vertical axis.

Fig. 4(d) shows the average obtained for all video sequences analyzed. Notice that besides the high concentration within the inner 256×256 square (in red), a distribution around the central axes is significant, even in areas far from the central point. Besides, is important to notice that even the atypical cases previous presented are covered in the average pattern observed in Fig. 4(d).

This analysis of best block matchings in the Raster Search step shows that, in most of points, the computational effort to perform the search during this step is wasted, since the majority of compared positions outsides the 256×256 square and the axes rarely lead to a best matching. Therefore, in order to speed up the Raster Search step with small penalties in coding efficiency, these search points could be removed from the search.

OARP is a search pattern proposed to optimize the search window and exploit the characteristics of motion vector distribution efficiently, aims to reduce the complexity of Raster Search in TZS. Fig. 5 presents the OARP, which was developed to perform the search within the square that represents 25% of the original number of points. It is important to notice that, although most of the OARP search is performed within the inner square, the axes were developed to reach the boundaries of the search window. Since, OARP is capable of exploit the regions of high probability of best block matchings. Considering the average distribution presented in Fig. 4(d), is noticeable that the four corners of the inner search square rarely lead to best block matchings, while the horizontal and vertical axes are well represented in the heat maps, but not totally covered by the inner square. This way, OARP replaces the search points that compose the four corners by the exact same amount of points in the axis regions. Due to this octagonal-plus-axis search area, OARP is capable of covering 62.3% of the total best block matchings found in the original Raster Search.

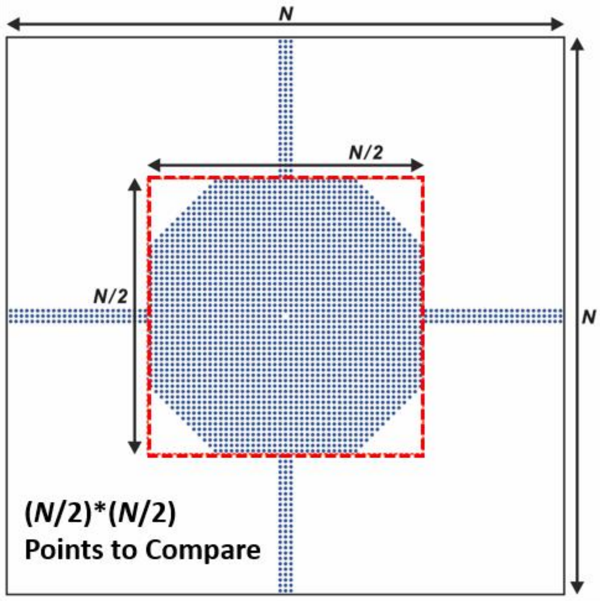


Fig. 5. Octagonal-Axis Raster Pattern (OARP).

Besides, is important to notice than OARP is suitable for the Raster Search step and can be easily integrated to others fast ME algorithms for TZS, like e-TZS, because it does not need neighboring cost information for directing the search. Besides, its fixed number of search points allows execution regularity, which is especially important for hardware implementations of ME algorithms.

V. Results and Comparisons

This paper proposed an efficient TZS algorithm suitable for mobile devices, with low computational overhead. The algorithm is composed of the combination of the techniques e-TZS [5] and OARP [6]. This section presents both the results for each individual technique and for the combined solution.

The proposed techniques were implemented in the HEVC reference software (HM-16.14). For all video sequences, 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) [13]. 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. The experiments were conducted aiming at single general-purpose processor, in order to obtain comparable results with related works. This methodology is employed by several authors aiming at complexity reduction in video encoders, such as [3], [5], [6], [10], [11], [14] and [15].

As the analyzed videos sequences were not the same in e TZS and OARP original papers, the full set of videos was tested once again in this work. 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, allow-

Table I. Individual impact of each model in e-TZS.

Tree	BD-rate (%)	TZS TR (%)	Total TR (%)
Pr-ET	-0.0458	13.76	2.29
FS-ET	0.0422	57.1	12.28
RS-ET	-0.0972	1.49	0.75
Pr+FS	0.1909	60.51	13.22
Pr+FS+RS	0.2095	60.66	12.98

Table II. Performance of e-TZS, OARP and proposed e-TZS + OARP combined scheme.

Class	Sequence	BD-Rate (%)			TZS TR (%)			Total 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	67.18	52.23	80.03	16.93	13.14	19.91
	ParkScene	+0.1558	-0.1985	+0.1268	58.67	38.18	69.54	08.43	05.89	10.32
	Cactus	+0.1543	-0.2087	+0.1434	59.18	53.65	78.61	14.02	12.73	17.86
	BQTerrace	+0.1236	-0.0541	+0.1570	57.74	**28.90**	**63.65**	06.77	**03.86**	**07.18**
	BasketballDrive	+0.1015	-0.2654	+0.1019	60.81	58.42	80.72	19.08	18.07	24.18
Class A	*Traffic*	+0.3131	-0.1882	+0.3466	61.47	30.37	69.85	07.23	03.62	08.26
	PeopleOnStreet	+0.1869	-0.7118	+0.0561	55.06	59.62	79.97	16.09	17.36	23.24
	NebutaFestival	+0.0425	-0.0457	-0.0344	59.80	42.81	70.50	08.64	06.39	09.96
	SteamLocTrain	+0.6568	-0.0245	+0.0879	66.01	**59.20**	**83.56**	19.62	**17.90**	**24.67**
	Average	**+0.2095**	**-0.2464**	**+0.1242**	**60.66**	**47.05**	**75.16**	**12.98**	**11.00**	**16.18**

ing a fair evaluation of the proposed method for different input condition.

Compression efficiency results are presented in terms of BD-Rate [7], which is a metric used to compare the bitrate difference between two encoding solutions considering the same video quality.

A. Results for e-TZS

As discussed in the previous sections, the e-TZS algorithm uses three decision trees to predict if the best block matching has already been chosen and in case affirmative, the strategy allows the interruption of the TZS execution. The decision trees were placed after Prediction (Pr-ET), First Search (FS-ET) and Raster Search (RS-ET) steps from TZS.

An analysis to evaluate the individual impact of each decision tree in the e-TZS was performed. On the three firsts lines of Table I, the average individual results for each tree in the e-TZS algorithm are presented. On the line four, the impact of the combined Pr-ET and FS-ET is computed and on line five, the impact of the three trees together, where is demonstrate the e-TZS approach.

When only the Pr-ET tree is executed, the e-TZS is able to reduce the encoding time in 13.76% with a gain in compression efficiency of 0.0458% in comparison with the original TZS algorithm. A considerable part of best MVs are located in the neighborhood of the analyzed block. As the prediction step looks in blocks very close to the current PU it can achieve satisfactory results on finding the best MV, which permits bit-rate saving in the encoding process. In this sense, by bypassing the posterior steps, besides of a time reduction, it can avoid some false positive on the next steps that could move away from the best block.

The FS-ET tree represents the biggest time reduction in comparison to the other decisions trees while, it was the unique decision tree that has caused losses in compression efficiency. This result was already expected once the FS-ET is placed before Raster step. The positive time reduction value is related with the non-execution of Raster. At the same time, the decrease of the compression efficiency demonstrates the importance of Raster in finding the best block matching in the TZS flow.

The RS-ET is located just after Raster and immediately before Refinement. The achieved time reduction is very small. However, when only this decision tree is enabled, a considerable result in terms of BD-Rate is noticed. The reason is similar as explained for the first tree: In some cases, the taken decisions from the refinement step could deviate the found MV from the real MV.

The result when Pr-ET and FS-ET are enabled together is close to the complete e-TZS algorithm, where the three decision trees works together. The most TR gains occurs in the FS-ET while RS-ET have a very small contribution. In the complete e-TZS algorithm, RS-ET presents few contributions to the TR result. For the individual analysis, the impact values were evaluated with the others two trees disabled. Considering all the trees working together, the impact of each one in the e-TZS algorithm can be different, once a decision taken previously by one tree maybe result in a different action later taken by another. Besides, the last trees use parameters acquired from the previous ones and many parameters are just available in the last steps of the TZS workflow.

The complete e-TZS approach, used in the combined scheme on this work, achieves an average time reduction of 60.66% in comparison with the original TZS algorithm and 12.98% for the whole encoding process, with a BD-Rate increase of only 0.2095%. The most gains in TR and the losses in BD-Rate are strictly connected with the reduction of the number of executions of the Raster step during the encoding process.

B. OARP Results

The e-TZS tries to skip the Raster execution when it is possible. However, to keep the good compression efficiency values from HEVC, sometimes it is necessary to execute the Raster step. The OARP approach is the other technique used in this paper and it presents a new search pattern to Raster step that was developed looking for the regions of the SA with higher concentration of best blocks matchings. The new SA focused on the center and axis of the SA can cover 62.3% of the total blocks contained inside the original SA searching in an area 75% smaller. For the default size of SA of the HM, a square of 512×512 positions, the method executes a total of 2652 comparisons instead of 10609 in case the entire SA were analyzed.

By reducing the SA, the number of comparisons and calculations reduce as well, what explain these time reduction values. As the pattern contains most of the best block matchings of the original SA, the bit-rate increase caused by the reduced SA is negligible. Considering OARP being always executed (the original TZS algorithm), a time reduction of 60.91% was achieved in the TZS processing time, which led to a time reduction of 21.57% in the total encod-

Table III.Comparisons with related works.

	Sequence	*Kimono*	*ParkScene*	*Cactus*	*BQTerrace*	*Basket-ballDrive*	Average
BD-rate (%)	[14]	1.91	1.07	0.87	0.45	2.5	1.360
	[15]	1.51	0.74	0.59	0.29	2.15	1.056
	Proposed	0.132	0.127	0.143	0.157	0.102	0.132
Total TR (%)	[14]	26.49	18.69	19.65	19.85	23.62	21.66
	[15]	27.11	19.18	21.18	20.04	26.66	22.83
	Proposed	19.91	10.32	17.86	7.18	24.18	15.89
BD/TR ratio	[14]	0.072	0.057	0.044	0.023	0.106	0.063
	[15]	0.056	0.039	0.028	0.014	0.081	0.046
	Proposed	0.007	0.012	0.008	0.022	0.004	0.008

ing time, with a BD-Rate of 0.04%.

C. Results for Combined Scheme

Table II shows the experimental results for the combined e-TZS and OARP schemes. The proposed combined solution achieved an average time reduction (TR) of 75.16% in comparison with the original TZS. 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, separately. BD-Rate results present an increase of 0.1242%, which is negligible when considering the achieved results in terms of time reduction.

By comparing the results in Table II, 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 on Raster Search results. Thus, the results in this combined scheme are mostly influenced by execution or not of the Raster Search, accordingly of decision took by the e-TZS part on the combined scheme.

D. Comparison with Related Works

Finally, Table III shows a comparison between this work and two related works found in the literature for complexity reduction of TZS algorithm. To allow a fair comparison, the set of video sequences used in the comparison in Table III is exactly the same as that listed in the two related works. Besides, the experimental setup is similar, since the two related works present results targeting a single general-purpose processor and following the recommendations in the CTC [13]. In [14], several techniques based on search range adaptation are implemented to decrease the number of search points in the TZS process, whereas in [15] 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 [14] and [15] 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 [7], 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 be-tween different works. The obtained BD-Rate/TR ratio is equal to 0.063 for solution [14], 0.046 for solution [15] 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. Hardware Savings Estimation

In HEVC, the inter-frame prediction is responsible for a significant share of the overall encoding performance, while taking 74% of the total encoding time [16]. Besides being costly in terms of computational effort, the Motion Estimation (ME) process is one of the most energy demanding operations in a video encoder. This energy consumption is a critical factor especially in mobile devices, which operate on limited battery capacity. Due to this issue, multiple hardware solutions for ME are found in the literature, such as [17] and [18].

A block matching operation (BMO) requires write and read tasks of a large amount of data from external memory and the application of a similarity criteria measurement, such as the Sum of Absolute Differences (SAD), commonly used in HEVC. In order to evaluate the efficiency of e-TZS and OARP when combined in terms of hardware performance, a BMO analysis was performed. Five frames of each sequence were encoded for QPs values 22, 27, 32 and 37, with the same configurations described in the previous section.

Fig. 6 shows the average BMO reduction considering the four QPs for each video. It is noticeable that the reduction in the number of BMO is directly related to the encoding time reduction presented in the previous section. *ParkScene* and *NebutaFestival*, which achieved smaller reductions in encoding time also, presented the lower BMO reduction, while *BasketballDrive* and *Kimono* provide the highest reductions in both BMO and encoding time. This is due to the content characteristic of the sequences, which perform the Raster Search step fewer times, but benefits from the e-TZS. Therefore, the e-TZS+OARP solution is capable of reducing the number of BMO in 80%, on average, when compared to the TZS algorithm.

The analysis of BMO reduction allows the estimation of some hardware optimizations that can be performed when employing the e-TZS+OARP approach. Firstly, as BMO is directly related to SAD operations, we can consider archi-

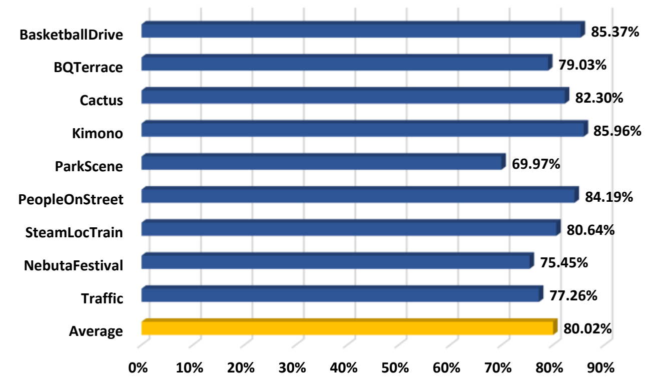


Fig. 6. Average BMO reduction.

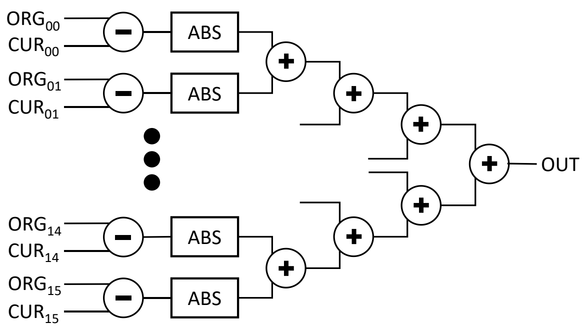


Fig. 7. SAD unit calculation (adapted from [20]).

tectures that optimize these operators. In [19] and [20], an architectural exploration applied to SAD units is proposed to reduce energy consumption while introducing a minimum impact on the coding efficiency.

Targeting at real-time processing of high definition videos, the authors in [19] present a fully combinational scheme that is able to process all the supported block sizes of current video encoders. Fig. 7 presents a combinational and fully parallel SAD architecture designed to operate over 4x4 blocks. The architecture is divided into five addition levels. The first level provides the absolute difference values between the samples from the original block (ORG_n) and the current block (CUR_n). The next levels compose an adder tree for the partial values coming from the first level. To process real-time video with 1920×1080 pixels at 60 frames per second, the authors in [19] propose an architecture with 13 SAD units (see Fig. 7). Four processing cores compose each SAD unit. The target frequency was defined to 497.66MHz, intending to process video sequences with 1920×1080 pixels at 60 frames per second [19]. The obtained SAD architecture presents a power dissipation and silicon area of 21.40mW and 179.6 gates, respectively, when synthesized for 45nm@1.1V Nangate standard cells technology [19].

In this sense, the proposed e-TZS+OARP approach can reduce the number of SAD calculations in 80% on average. Therefore, to provide the same conditions (targeting real-time processing), an architecture that considers e-TZS+OARP can be thoroughly simplified by decreasing the number of processing cores and SAD units. Based on the BMO reduction and considering the same target frequency (497.66MHz), a reduction from 52 to 11 processing cores and, consequently, from 13 SAD units to 3 can be estimated. Thus, disregarding the control area, this architecture can reduce power dissipation in up to 4.28mW and the silicon area in up to 35,920 gates. When included in a simplified HEVC encoder (e.g., mobile purposes), the e-TZS+OARP implementation can further reduce energy consumption with minimal impacts in encoding efficiency.

The second issue related to BMO operations is the number of memory accesses performed by the ME, which is responsible for about 90% of the energy consumption in this encoding step [21]. Most of this consumption is because ME requires frequent external memory accesses to perform the block matching algorithms. Since the number of BMO is reduced in 80%, the amount of external memory access will be also significantly reduced when the proposed strategy is employed.

Since both reductions in SAD calculations and memory access are capable of reducing energy consumption, we claim that the proposed e-TZS+OARP is an intelligent and efficient approach that minimizes hardware usage and allows simplifications to improve energy savings for hardware design in both processing and memory access.

VII. Conclusions

The HEVC standard achieves high compression efficiency, which is very helpful for storage and video broadcasting in higher resolutions over the Internet. However, the HEVC encoding process demands more processing power and time than its predecessor standards. The elevated complexity of HEVC affects mobile devices due to its energy consumption limitations. Thus, to better attend these devices, energy-aware solutions should be developed.

In this sense, this paper presented a combined scheme based on two fast TZS algorithms with significant results in complexity reduction without compromising the compression efficiency. When combining the strategies, the original TZS encoding time was reduced by 75.16%, with a negligible loss of 0.1242% in compression efficiency (measured in BD-Rate). The solution reduced the number of block matching operations in 80%, which has a direct impact on energy savings in hardware implementations, allowing the number of SAD unities to be reduced or the operation frequency of the ME module to be decreased. Moreover, the proposed method can significantly reduce external memory access, which represents an important impact on energy savings.

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] CISCO, "Global Mobile Data Traffic Forecast Update 2016-2021," in *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] 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.

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

[5] 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.

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

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

[8] Cheung, Kwan C., Po, L.M., “Normalized partial distortion search algorithm for block motion estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 417– 422, 2000.

[9] Pan, Y. Zhang, S. Kwong, X. Wang, and L. Xu, “Early termination for TZSearch in HEVC Motion Estimation,” *IEEE Int. Conf. on Acous. Speech and Signal Process. (ICASSP)*, pp. 1389–1393, 2013.

[10] F. Belghith, H. Kibeya, H. Loukil, M. A. B. Ayed, and N. Masmoudi, "A new fast motion estimation algorithm using fast mode decision for highefficiency video coding standard," *Journal of Real-Time Image Processing*, vol. 11, pp. 675-691, 2016.

[11] Pan, Y. Zhang, S. Kwong, X. Wang, and L. Xu, “Early termination for TZSearch in HEVC Motion Estimation,” *IEEE Int. Conf. on Acous. Speech and Signal Process. (ICASSP)*, pp. 1389–1393, 2013.

[12] 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.

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

[14] 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.

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

[16] V. Afonso et al., “Hardware implementation for the HEVC fractional motion estimation targeting real-time and low-energy,” *J. Integr. Circuits Syst.*, vol. 11, no. 2, pp. 106–120, Apr. 2016.

[17] M. E. Sinangil, V. Sze, Z. Minhua, A. P. Chandrakasan, “Cost and coding efficient motion estimation design considerations for High Efficiency Video Coding (HEVC) standard,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1017-1028, 2013.

[18] P. Nalluri, L. Alves; A. Navarro. “High speed SAD architectures for variable block size motion estimation in HEVC video coding,” *IEEE International Conference on Image Processing*, pp 1233-1237, 2014.

[19] R. Porto, L. Agostini, B. Zatt, M. Porto, N. Roma and L. Sousa, "Energy-efficient motion estimation with approximate arithmetic," 2017 *IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, Luton, 2017, pp. 1-6.

[20] R. Porto, L. Agostini, B. Zatt, N. Roma and M. Porto, "Power-Efficient Approximate SAD Architecture with LOA Imprecise Adders," *2019 IEEE 10th Latin American Symposium on Circuits & Systems (LASCAS)*, Armenia, Colombia, 2019, pp. 65-68.

[21] B. Zatt, M. Shafique, F. Sampaio, L. Agostini, S. Bampi and J. Henkel, "Run-time adaptive energy-aware Motion and Disparity Estimation in Multiview Video Coding," *48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, New York, pp. 1026-1031, 2011.