Journal of Real-Time Image Processing (2021) 18:2495–2510
https://doi.org/10.1007/s11554-021-01138-3

ORIGINAL RESEARCH PAPER

Low-energy motion estimation memory system with dynamic management

Dieison Soares Silveira[1] · Lívia Amaral[2] · Guilherme Povala[2] · Bruno Zatt[2] · Luciano Volcan Agostini[2] · Marcelo Schiavon Porto[2] · Sergio Bampi[3]

Received: 20 January 2021 / Accepted: 29 May 2021 / Published online: 11 June 2021

Abstract

The digital video coding process imposes severe pressure on memory traffic, leading to considerable power consumption related to frequent DRAM accesses. External off-chip memory demand needs to be minimized by clever architecture/algorithm co-design, thus saving energy and extending battery lifetime during video encoding. To exploit temporal redundancies among neighboring frames, the motion estimation (ME) algorithm searches for good matching between the current block and blocks within reference frames stored in external memory. To save energy during ME, this work performs memory accesses distribution analysis of the test zone search (TZS) ME algorithm and, based on this analysis, proposes both a multi-sector scratchpad memory design and dynamic management for the TZS memory access. Our dynamic memory management, called neighbor management, reduces both static consumption—by employing sector-level power gating—and dynamic consumption—by reducing the number of accesses for ME execution. Additionally, our dynamic management was integrated with two previously proposed solutions: a hardware reference frame compressor and the Level C data reuse scheme (using a scratchpad memory). This system achieves a memory energy consumption savings of 99.8% and, when compared to the baseline solution composed of a reference frame compressor and data reuse scheme, the memory energy consumption was reduced by 44.1% at a cost of just 0.35% loss in coding efficiency, on average. When compared with related works, our system presents better memory bandwidth/energy savings and coding efficiency results.

Keywords Video coding · Motion estimation · Test zone search · Dynamic memory management · Energy optimization

1 Introduction

Recent hardware technology advances have strong impacts in the IT market, where consumer electronics gadgets are short-lived. Digital devices, such as smartphones, cameras, and tablets, are released and soon surpassed by new increasingly powerful devices able to perform a larger number of tasks. This evolution brought higher quality and diversification to multimedia services, such as audio/video quality and the ability to easily record and transmit multimedia content, among others.

To quantify this trend, a forecast by Cisco shows that the internet video traffic will grow fourfold from 2017 to 2022, accounting for 82% of total internet traffic by the end of the forecast period [1]. Furthermore, with the unexpected COVID-19 pandemic, the internet traffic increased in all regions, mainly from streaming video and videoconferencing traffic [2]. Moreover, battery-powered devices are ever more ubiquitous, especially the ones that handle digital videos such as smartphones. These devices will account for 44% of total IP traffic in 2022, up from 18% in 2017 [1], which is relevant given the limited memory and energy availability in mobile devices. In turn, digital

Dieison Soares Silveira and Livia Amaral are equally-credited first authors.

✉ Dieison Soares Silveira
dieison.silveira@canoas.ifrs.edu.br

[1] Federal Institute of Education, Science and Technology of Rio Grande do Sul and Graduate Program in Microelectronics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

[2] Federal University of Pelotas, Pelotas, Brazil

[3] Informatics Institute at the Federal University of Rio Grande do Sul, Porto Alegre, Brazil

video content has also become omnipresent. By 2022, a million minutes of video content will cross the network every second [1]. To satisfy users' demand and allow them to capture and to store videos, mobile devices must be able to handle quality videos and ever-increasing resolutions. However, when dealing with battery-powered devices, serious battery-related restrictions must be considered. To prolong the battery lifetime of battery-powered devices, the reduction of memory related energy consumption becomes an essential task [3].

In this context, video coding is a key enabling technology. Without compression, videos would require a prohibitive data usage to be represented, stored, and, eventually, transmitted. Video encoders incorporate many encoding tools that are constantly in development. With these tools, digital videos are represented with much smaller volume of data at the cost of heavy processing and some controlled losses in visual quality.

The evolution in the encoding tools generated different video coding standards, such as the high-efficiency video coding (HEVC) [4]. HEVC introduces plenty of innovations to video processing, such as new coding structures, larger prediction units, variable size transforms, and features to enhance parallel processing capability [5]. Thus, HEVC promoted 50% bitrate reduction for similar objective video quality in comparison to its ancestor, the H.264/AVC [6].

However, the new tools implemented in HEVC promoted a significant increase in computational effort of the encoder ranging from 9 to 502%, depending on the HEVC configurations and the used HEVC Test Model (HM) [7]. Besides, the volume of data accessed in memory is more than twice as high as required in H.264/AVC [6], which leads to more complex hardware designs. The tool that most contributes to this heavy load is the ME since the ME process takes around 80% of the total encoding time and 50% of the total HEVC encoder energy consumption [8, 9], even using a fast algorithm as TZS [10]. Besides, approximately 65% of the HEVC encoder memory accesses are made by the ME [11].

ME demands intense memory communication, which leads to high energy consumption in video coding systems. As a result, about 70–90% of ME energy consumption is spent on accessing internal and external memories [12, 13]. As memory communication significantly affects energy consumption and performance, this is a major bottleneck in a video coding system. Hence, reducing memory energy consumption is mandatory for energy-efficient high-performance video encoders.

1.1 Novel contributions

This work proposes a dynamic memory management system to better explore the relationship between coding efficiency and energy consumption when processing the ME. This system consists of a SRAM-based scratchpad memory (SPM), which allows power gating at the sector level. This memory is used to store two search areas (SA) of neighbor blocks using a Level C data reuse strategy [14], allowing data reuse among these two neighbor blocks. The system also includes the double differential reference frame compressor (DDRFC) [15], which was developed in our group. The proposed system reduces the communication with external DRAM memory when ME is requesting data from reference frames, reducing the energy required to perform the ME.

The proposed memory manager turns off SRAM memory sectors when the accesses made to them become irrelevant to the TZS. The inactive memory sectors reduce the static energy consumption in the internal memory proportionally with the number of powered off SRAM cells. A reduction in the dynamic energy consumption was also achieved since inactive memory sectors imply fewer blocks available for access during the TZS process. Data unavailable in memory will not be delivered to the TZS, even if requested, reducing memory access. The main contributions of this article are summarized as follows:

1. *A statistical analysis of the TZS memory accesses*, which allows better understanding of the TZS behavior within the SA and the definition of the most/least accessed regions.
2. *A multi-sector scratchpad memory design* that employs power gating at the sector level to reduce the static energy consumption of the internal memory.
3. *A dynamic management algorithm*, which is able to dynamically control the ME process according to the number of memory accesses required by the TZS.
4. *A memory system for the ME* that combines reference frame compression, Level C data reuse, and dynamic management, to reach high memory bandwidth/energy reduction.

The rest of this article is organized as follows: Sect. 2 presents related works; Sect. 3 presents the TZS algorithm; Sect. 4 explains in detail the TZS statistical analysis, the proposed dynamic management system, the memory access behavior of the TZS, and the multi-sector SPM design proposed. Section 5 presents the energy consumption model developed to calculate the energy consumption savings by the proposed system. The energy consumption and coding efficiency results are shown in Sect. 6, then conclusions are presented in Sect. 7.

2 Related works

The approaches most often used to reduce external memory accesses in video encoders are: (1) reference frames compression; (2) data reuse; and (3) dynamic control of search range for the block matching during ME. In the following subsections, related works with these approaches are briefly reviewed.

2.1 Reference frame compression

This approach consists of two phases: (1) compression of the reference frame blocks before they are stored in the external memory; and (2) decoding process of previously compressed blocks, since they will be used as future references during the coding process. Lossy or lossless algorithms can be herein employed.

Lossy compression approaches generally achieve higher compression ratio than lossless solutions, since they use quantization in the compression algorithm. However, lossless solutions preserve the video quality, but they insert non negligible computational cost by employing various combined techniques to perform the compression, which leads to increased power dissipation and latency.

In works presented by Lian et al. [16] and Willeme et al. [17], lossy reference frame compression schemes are proposed. In these works the authors use techniques such as quantization, differential coding, variable length coding, and transforms to compress the reference frame. Whereas, in works presented by Lee et al. [18], Lian et al. [19] and Silveira et al. [15] the authors proposed lossless reference frame compression approaches, which maintain the video quality.

2.2 Data reuse

Data reuse schemes reach significant memory bandwidth reduction when the memory accesses are sequential. The data are brought in a specific order from the external memory to the internal memory (SRAM cache), reducing the access latency and external memory bus contention, all leading to less memory bandwidth required.

The work proposed by Tuan et al. in [14] presents the Level C data reuse scheme, which explores the global locality in the search area strip among different search areas. This way, Level C works as follows: an entire SA is used in internal memory for coding a given block N. After block N is coded, its neighboring block $N+1$ will also be coded. As both blocks share the main part of the SAs, the overlapped area will not be removed from memory. However, areas that only correspond to the SA of block N will be replaced by information pertaining only to the SA of block $N+1$. By bringing only the different information, there is an important reduction in redundant memory access. Level C will be detailed in Sect. 3.

In [20], a reuse scheme based on overlapped SAs among neighboring current blocks is used. This work proposes a zigzag scan called Level C+. This scheme requires special coding orders, which is not convenient for hardware implementation since the control complexity is increased. However, Level C data reuse is a hardware-friendly solution, which achieves a large reduction in external memory bandwidth requirement and is widely employed in video encoder architectures.

2.3 Dynamic search range

Dynamic search range schemes employ dynamic SA adjustment algorithms. The adjustment is usually performed based on the sizes of the motion vectors found for the previous frame.

Dai et al. [21] propose the use of Cauchy distribution to improve the effectiveness in predicting the distribution of the motion vectors sizes in the frame being encoded. In Du et al. [22] and Chien et al. [23], the authors propose an adaptive search range algorithm that, considers the prediction unit (PU) [5] motion. The relationship between the motion vector predictor, the difference between PUs, and the SA size are considered by Li et al. [24].

Ji et al. [25] modeled a deviation metric of the motion vector predictor to predict the relationship between the size of the SA and the difference of the motion vector. In Pakdaman et al. [26] and Singh et al. [27] the authors propose a new fast ME algorithm to reduce the number of search points in the ME process.

The aforementioned works did not present evaluations that focus on the number and location of accesses within the SA when a fast algorithm is used. Furthermore, there is a need for a memory system able to integrate the three discussed approaches to improve energy savings, mainly considering fast search algorithms.

3 Motion estimation and the TZS algorithm

ME is present in all major video coding standards and its goal is to reduce the temporal redundancies among neighbor frames [5]. For each block of the current frame, the ME searches for a block (in previously encoded reference frames) that presents the highest similarity with the current frame block [5]. The displacement between the best-matching block and the current block is represented by a motion vector.

Considering the HEVC standard, the frame being encoded is divided into blocks called coding tree units (CTU) [5]. The ME search is performed within a SA in order to reduce the ME computation cost. The size of the SA considers the block size and the search range (SR) size. The SR defines the maximum horizontal and vertical distances (in relation to the current block position) the search is allowed. These distances are measured in samples. Equation (1) defines the SA size, where BS_h and BS_v are the block horizontal and vertical sizes and SR_h and SR_v indicate the horizontal and vertical SRs.

$$SA = (2SR_h + BS_h) \times (2SR_v + BS_v) \tag{1}$$

One of the most efficient ME algorithms proposed in the literature is the TZS algorithm [10]. This algorithm divides the search process into four steps: motion vector prediction, first search, raster search, and refinement. With this strategy the TZS reaches results close to the optimum, but without the prohibitive cost of an exhaustive search done by the Full Search algorithm [28]. TZS is the current state-of-the-art ME algorithm and compliant with any video encoder standard. The TZS is adopted as default by the HEVC reference software [29]. Since the TZS is the focus of this work, this section will detail the four steps that compose the algorithm.

The first step of TZS consists of the motion vector prediction (MVP) phase. This step is responsible for relocating the SA to the most promising location in the frame, based on previously encoded blocks [10]. While the MVP step contributes to decrease coding time, the fact that the SA may not be always formed around the collocated block hinders hardware implementations and the use of efficient data reuse strategies [30].

The First Search is the second step of the TZS algorithm. It starts at the center of the SA defined by the MVP and expands toward the edges of the SA using a square or diamond format [10]. The expansion step increases in power of two. The First Search has two stopping criteria. The first is when it reaches the border of the SA. The second limits the number of consecutive expansions (typically three expansions) unable to return a better result.

If none of the compared points of the three first expansions has a better result than the central block, the First Search stops and returns block 0 as a best-match result. Since the three next expansions did not return a better result and the fourth expansion reached the border of the SA, the First Search finishes the search and returns the block as the best one.

The third TZS step is the Raster Search. However, the Raster Search only occurs when the motion vector from the First Search is greater than the *iRaster* constant, which has a value 5 by default [10]. When executed, the Raster Search applies a block subsampled Full Search scan throughout the SA, returning the best-matching block found. The *iRaster* constant also defines the horizontal and vertical block subsampling. Then, by default, only one at each five horizontal and five vertical blocks are compared. Even with this subsampling, this is the most computationally intensive TZS step [10].

Finally, the fourth and last step is refinement. The refinement takes the best-matching block, from either the First Search or the Raster Search (when performed), and executes the same pattern of First Search. However, while the First Search starts in the center and goes for the edges of the SA, the refinement always updates its center with the best result found in its last iteration. The Refinement step has two stop conditions: (1) when the expansion reaches the edge of the SA, and (2) when no block more similar than the one being encoded is found after a two-level expansion [10].

The use of fast algorithms such as TZS and SA smaller than the complete frame reduce the computational cost to perform the ME. However, HEVC uses tools that lead to better coding results at the cost of a significant increase in memory accesses and complexity [31].

Therefore, all efficient ME architectures use some form of data reuse. Several repetitive accesses are made to the exact pixel locations in advanced video coding standards due to excessive block matching operations.

In this sense, Level C data reuse is one of the most used in ME architectures and presents a hardware-friendly implementation, providing a high memory bandwidth reduction rate. Level C data reuse explores the data locality in the SA strip across neighboring SAs.

Figure 1 illustrates the Level C data reuse process between two SA. In this figure, SR means search range and *N* is the size of the block. The gray area represents the reused area between two SA. Equation (2) shows the amount of samples reused for each new SA when Level C is used.

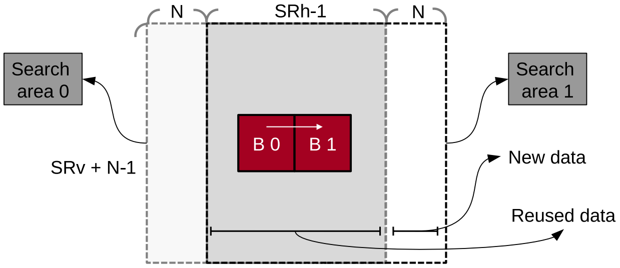


Fig. 1 Level C data reuse process, which explores the global locality between SA 0 and SA 1

$$\mathrm{LevelC}_{\mathrm{SamplesReused}} = (\mathrm{SR}_{\mathrm{v}} + N - 1) \times (\mathrm{SR}_{\mathrm{h}} - 1) \tag{2}$$

To quantify the ME external memory bandwidth Eq. (3) can be used. In this equation, the external memory bandwidth (MemB) using Level C is considered. In Eq. (3), FS is the frame size and W and H are the width and height of the frame and block, respectively.

$$\mathrm{MemB} = \mathrm{RF} \times \mathrm{FPS} \times \frac{\mathrm{FS_H}}{\mathrm{BS_{SH}}} \times \left(\mathrm{SA_W} \times \mathrm{SA_H} + \left(\frac{\mathrm{FS_W}}{\mathrm{BS_W}} - 1 \right) \times \mathrm{BS_W} \times \mathrm{SA_H} \right) \tag{3}$$

As one can notice in Eq. (3), the factors that most impact external memory bandwidth are the frame size, the number of reference frames (RF), the frame rate per second (FPS), and the block size (BS). The block size determines the amount of SA that will be brought from the external memory.

Thus, the encoder design must consider that a major bottleneck lies in the communication between external memory and processing units. Therefore, there is a strong need for techniques to reduce external memory communication, especially during the ME process.

4 Proposed ME memory system with dynamic management

This section presents the analysis and the development of the proposed low-energy TZS memory system with dynamic management. The system proposed in this article employs three strategies to reduce the external memory accesses and energy consumption. These strategies are: (1) the DDRFC reference frame compressor; (2) the Level C data reuse scheme, and (3) a dynamic control to power gate memory sectors. These three solutions are explored together to reduce the TZS energy consumption with low coding efficiency loss. This system is described in Fig. 2.

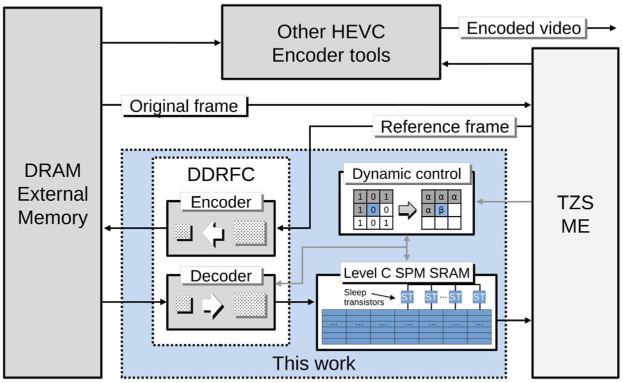


Fig. 2 The evaluated dynamic management system integrated with the video encoding unit, with an external memory and connected by the reference frame compressor

As one can notice in Fig. 2, the DDRFC operates in both read and write operations in the external memory. This approach consists of two phases: compression of the reference frame blocks before they are stored in the external memory; and decompression of the previously compressed blocks, since they will be used as future references during the coding process. The DDRFC was presented in our previous work and performs intra-block double differential coding over 8 × 8 sample blocks followed by a static Huffman coding [15]. The DDRFC is a lossless reference frame compressor and guarantees block-level random access by avoiding data dependencies between neighboring blocks.

The Level C data reuse scheme was implemented in a SPM SRAM memory. The SPM was implemented as a circular buffer where the beginning of the SA is indicated by a flag. This allows the Level C to avoid unnecessary write operations every time a SA shift occurs. Furthermore, the memory banks of the SPM can be power gated using a single sleep-state transistor [32]. The dynamic management module controls the power gating and the flag offset along the memory banks.

The dynamic management module developed in this work uses information from the neighboring CTUs to define if a sector can be turned off or on. The dynamic management intends to find the best relation between energy consumption and video coding efficiency.

The following sections will describe the dynamic management memory system proposed in this work. Section 4.1 presents a statistical analysis that motivated the dynamic management development. In Sect. 4.2 the details of the scratchpad memory design are shown. In Sect. 4.3 the dynamic management strategy of TZS SPM will be discussed.

4.1 TZS experimental analysis

To support the design of our memory hierarchy, an analysis of the samples accessed by TZS inside the SA was done through experiments using the HEVC reference software, the HM version 16.6 [29]. The first 100 frames from six HD 1080p test sequences were used to quantify the number of times each sample is required within the SA. These video sequences were recommended by the HEVC common test conditions (CTCs) [33] and are called *Basketball Drive*, *BQTerrace*, *Cactus*, *Kimono*, *ParkScene* and *Tennis*. These sequences were encoded with the four quantization parameters (QP) defined by the CTCs: 22, 27, 32 and 37.

The low delay P configuration was adopted, since this configuration is the one recommended by the HEVC CTCs.

As the proposed dynamic controller focuses only on the TZS stage, this module was isolated in the HM software to better evaluate the impact of the dynamic control. This way, the early skip [5] and the occurrence of intra blocks in inter frames [5] were disabled. Besides, the support to bi-directional prediction [5] was disabled and the search was limited to one reference frame.

The TZS has five MVP predictors to select the SA position in the reference frame: one based on the collocated block and four predictors based on neighboring blocks. By applying predictors based on neighboring blocks, TZS can move the SA to a distant region from the collocated block position, hinders a predictive behavior of the external memory accesses, and generates irregular access to the external memory. These two consequences of the MVP use limit efficient memory management schemes in hardware, increasing the energy consumption and limiting the ME throughput [27].

This way, to achieve a better trade-off between the performance and hardware cost, only the prediction based on the collocated block was maintained in TZS for all evaluations. This approach is also adopted by other hardware solutions proposed in the literature. As presented by [27], the consequence of not using all five MVP predictors is 0.25% in BD-rate. Then, the SA was defined around the collocated block inside the reference frame. Besides, a CTU of size 64×64 samples was defined. A SR with $[-64, +64]$ samples were assumed, totaling a SA of 192×192 samples, according to Eq. (1). This SR is the HM default value. Finally, the HM 16.6 [29] was modified to obtain the information required for this evaluation into a trace file. Given that non-square blocks are less frequently used than square blocks [5], non-square blocks were disabled in the ME process for these experiments.

Figure 3 shows histograms of the number of accesses made to each sample within the SA considering two corner case video sequences (BQTerrace and Tennis sequences in Fig. 3a,b, respectively), considering the average of the four QPs of each video. Moreover, Fig. 3 shows the histogram of the average accesses for the six videos.

As one can notice in Fig. 3, the most accessed region in the SA is the central one, from where the expansion of TZS starts during the First Search. This happened for all video sequences. However, videos with features such as high motion or texture, such as the Basketball Drive, Kimono and Tennis (Fig. 3b), presented more accesses closer to the edge of the SA than videos that have low motion or texture, where the accesses are more concentrated in the central region.

These histograms reassure that the occurrence of most accesses is in the small central region of the SA, i.e., peripheral parts of the SA receive few accesses. From the average of accesses generated for each video, it was possible to obtain a relation between number of accesses and the proportion of the used area within the SA. This relation is shown in Fig. 4. It shows that 50% of the most accessed samples within the SA lie in a region that corresponds to only 15.81% of its total size. Furthermore, 95% of the accesses are carried out in a region corresponding to 66.65% of the SA. The relation expressed in this graph is important since it motivates a multi-sector memory design

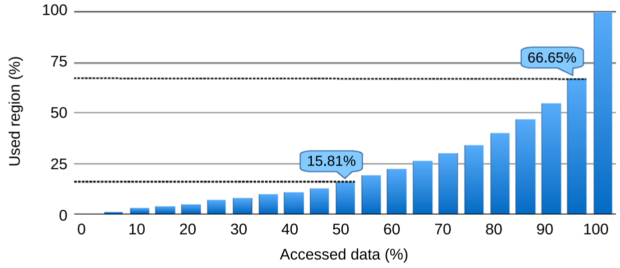


Fig. 4 Relation between most accessed data and used region within the SA

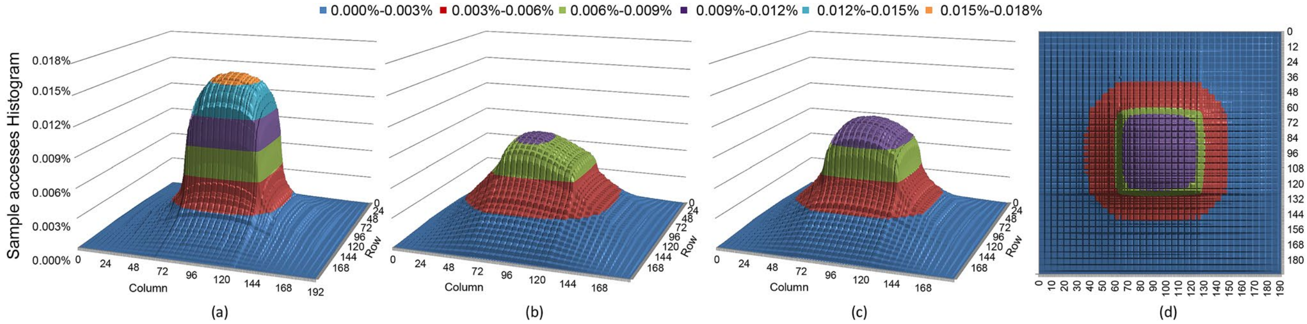
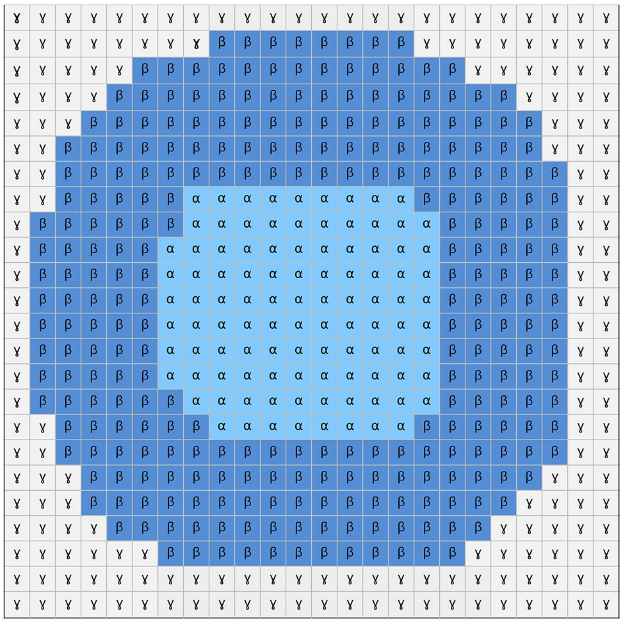


Fig. 3 Histogram of TZS memory accesses: **a** BQTerrace histogram shows the highest central region accesses; **b** Tennis histogram presents the lowest central region accesses. **c**, **d** present different points of view for the histogram with the average TZS memory accesses for the six videos evaluated



γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ
γ	γ	γ	γ	γ	γ	γ	γ	β	β	β	β	β	β	β	β	γ	γ	γ	γ	γ	γ	γ	γ
γ	γ	γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ	γ	γ	γ
γ	γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ	γ
γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ
γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ
γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ
γ	γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	β	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	β	β	β	β	β	β	α	α	α	α	α	α	α	α	α	α	β	β	β	β	β	γ	γ
γ	γ	β	β	β	β	β	β	α	α	α	α	α	α	α	α	β	β	β	β	β	β	γ	γ
γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ
γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ
γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ	γ
γ	γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ	γ	γ
γ	γ	γ	γ	γ	γ	β	β	β	β	β	β	β	β	β	β	β	β	γ	γ	γ	γ	γ	γ
γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ
γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ

Fig. 5 Search area divided in three sectors, where sector α has 103 blocks, sector β has 281 blocks and sector γ has 192 blocks

and the dynamic memory management developed in this work.

4.2 Multi-sector scratchpad memory design

To build the proposed SPM with dynamic control, the original SA was divided in sectors. Initially, several sizes and number of sectors were investigated and the approach employing three sectors showed the best coding efficiency results: α (central), β (intermediate), and γ (border). The division of the SA into these three sectors is presented in Fig. 5. In this figure, each of the 24×24 positions represents a block of 8×8 samples, totaling a SA of 192×192 samples. The sector division was made considering the information obtained from the evaluation presented in the previous section and presented in Figs. 3 and 4. The sectors α and β represent the same area size of a SR $[-32, +32]$. However, due to the diamond shape of sectors α and β, this arrangement achieves better quality and coding efficiency results than the square area of the SR $[-32, +32]$.

Table 1 Average results of the TZS memory accesses by sector

Video	Sector α (%)	Sector β (%)	Sector γ (%)
SA percentage	17.89	48.78	33.33
BasketballDrive	50.84	39.23	9.93
BQTerrace	84.55	13.34	2.10
Cactus	84.23	13.47	2.30
Kimono	57.78	35.41	6.80
ParkScene	85.14	12.51	2.36
Tennis	48.87	41.61	9.53
Average	68.57	25.93	5.50
Std. Dev.	17.86	12.95	3.72

The next step was a statistical evaluation to obtain the list of memory accesses by sector when running the TZS. This evaluation was done encoding the same six videos used in the previous section, with QP 32. In Fig. 5, Sector α has the smallest slice, which corresponds to 17.89% of the SA size. Sector β, constitutes a region with 48.78% of the SA. Finally, sector γ, represents an area of 33.33% of the SA. The list of accesses by sector is shown in Table 1, where the results are presented for six video sequences, as well as the average and the standard deviation. In this table, one can notice that most accesses for all videos is made in sector α, from where the search of TZS starts during the First Search phase. This result is even more evident for low movement or texture videos where more than 80% of accesses were made in this sector.

High motion videos had about 50% of average accesses in sector α and about 40% in sector β. Sector γ was the one with the lowest number of accesses. In slow motion videos, the number of accesses in this sector was lower than 3% and for high motion videos, this number did not reach 10%.

Given that sector γ has 33.33% of the SA samples and that it has a very low rate of accesses (5.5% on average), the first optimization presented in this work was to remove sector γ from the SA. The decrease in the SA size allowed the reduction of the SPM size, from 576 8×8 sample blocks as shown in Fig. 5, to 384 8×8 sample blocks. The coding efficiency impacts of this design decision will be discussed later in this article.

Then, just two sectors were used in this work: α and β. Sector α is always active, and sector β can be power-gated according to the developed dynamic control. The SPM design allows a sector-level memory control, using the power gating in groups of cells corresponding to a specific sector. This allows a static energy consumption reduction, resulting from the SRAM cells' shutdown and

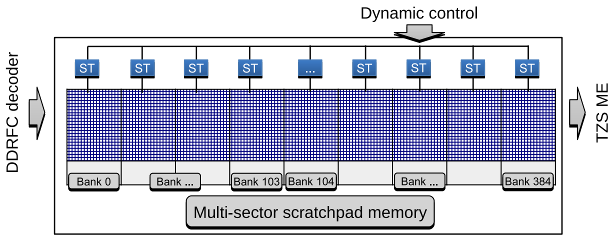


Fig. 6 Design of the multi-sector SPM architecture composed of 384 memory banks with 64 bytes each.

a dynamic energy consumption reduction, with a reduction in the number of samples compared during the coding process. However, an efficient control must be developed to achieve a good relationship between energy consumption reduction and coding efficiency. For this, different strategies for dynamic memory management are addressed in the next section.

Figure 6 presents the architecture design of the proposed multi-level SPM. The SPM contains 384 memory banks, where each bank can store one 8 × 8 sample block. The memory banks are composed of 16 rows and 32 columns, reaching 64 bytes of capacity each bank. The SPM can store two sectors, α and β. When just sector α is turned on, only 103 memory banks are necessary.

On the other hand, both sectors are turned on, and all the 384 memory banks are used. All memory banks can be power gated by the sleep-state transistors (ST), which are used as switches to shut off power supplies to memory banks [32]. When the memory banks corresponding to the sector β are turned off, the stored data are lost since this data will no longer be necessary. However, only memory blocks that will not be used by the next SA are turned off, blocks that will compose the next SA remain turned on into the SPM.

In addition, with each new SA change, new data will be brought from the external memory to the SPM. However, only data from the region that is not overlapped with the region already stored in the SPM is brought, avoiding bringing the same data back to the internal memory. This occurs with both sectors α and β, but if the new SA has sector β turned on, all the memory banks of the SPM will be turned on.

Furthermore, the SPM presented in Fig. 6 operates as a circular buffer. When a new block is being coded, the flag indicates the memory bank with the beginning of the new corresponding SA. Besides, common SA blocks from both sectors α and β are maintained, if sector β was used, and new blocks are read from the external memory. Thus, the Level C scheme works optimally, avoiding unnecessary writing operations each time a SA shift occurs. The dynamic management module controls both power gating and flag offset along the memory banks. Besides, the SPM receives data from the DDRFC decoder, stores them into the cells, and sends them to the TZS ME under request.

4.3 Developed neighbor management SPM control

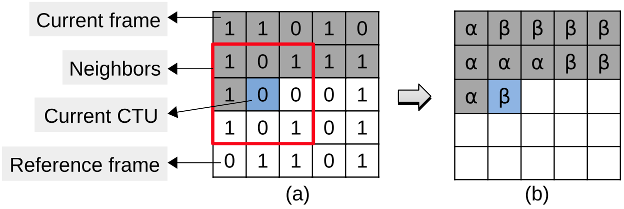
In this section, the dynamic management strategy of TZS SPM memory will be discussed. The memory control proposed in this work intends to find the best relation between energy consumption savings and coding efficiency losses. Thus, the system evaluated in this section has a memory with dynamic management, aiming to activate or deactivate sector β when convenient.

The decisions are done using an auxiliary binary matrix that represents the requests of each CTU inside a frame to access sector β. In an HD 1080p video, the auxiliary matrix will have 510 positions, since each frame with this resolution has 17 rows and 30 columns of 64 × 64 CTUs. Each position in the auxiliary matrix is updated after TZS encodes each CTU. Thus, a position in the matrix may receive the value "1" when the corresponding CTU requests the activation of sector β or the value "0" otherwise. The auxiliary matrix is updated at each processed CTU and it will be used to encode the next CTU. All CTUs of the first frame will be encoded considering that the sector β is active.

The TZS dynamic management strategy developed in this work is called *Neighbors management* (*NM*). The *NM* consults the auxiliary matrix for each CTU being encoded. If the collocated CTU requests sector β, the CTU being encoded will have sector β active during its encoding. A second criterion was also defined: if most of the eight neighboring positions have the current CTU request sector β active, this CTU will have sector β enabled during its coding.

As the auxiliary matrix is always updated after encoding the current CTU, management based on neighbors considers neighboring CTUs that have already been encoded in the current frame (frame N) and CTUs encoded in the previous frame (frame $N-1$).

Figure 7 presents an example of how the decision to activate or not sector β is made in *NM*. In this figure, the gray color indicates CTUs already encoded in the current frame, the white color indicates CTUs that have not yet been encoded in the current frame but have been encoded in the previous frame. The blue color indicates the current CTU being encoded. In this example, even that the collocated CTU is not requesting the use of sector β (position with "0" in the auxiliary matrix presented in Fig. 7a), five of the eight neighbors of the current CTUs are requesting sector β (as presented in Fig. 7a), then, sector β will be active to encode this CTU (as presented in Fig. 7b). Only



1	1	0	1	0
1	0	1	1	1
1	0	0	0	1
1	0	1	0	1
0	1	1	0	1

α	β	β	β	β
α	α	α	β	β
α	β			

Fig. 7 Illustration of the NM solution: **a** auxiliary matrix and **b** activated sectors

after encoding the current CTU, the auxiliary matrix in the corresponding position will be updated.

5 Energy consumption model

This section presents the energy consumption model developed to calculate the total energy consumption savings resulting from the dynamic management system. The energy consumption model used in this work follows the model presented by Amaral et al. [34]. This model can calculate the total energy consumption of the video encoding systems with different configurations, such as data reuse strategies, memory hierarchies, and reference frame compressors. The input of this model is the TZS algorithm trace, with detailed information of all blocks requested during TZS execution. The model output consists of the energy consumption of the evaluated solutions. The system addressed in this work and evaluated by this model involved the joint use of the DDRFC [15], Level C data reuse, and the dynamic TZS controller, as presented in Fig. 2 and previously discussed.

As this work proposes a TZS dynamic management from an internal memory that allows power gating at the sector level, the monitoring of the static consumption of this memory becomes relevant. This is because, by turning off memory sectors, energy savings in the static energy consumption of internal memory are obtained. To quantify this savings, the energy consumption model calculates the static energy consumption over the cycles required to code each CTU.

This work considers that the processing core of the TZS hardware will receive 512 bytes per clock cycle as input and will operate at 100 MHz using as reference a previous TZS architecture designed in our research group [30]. This target frequency is sufficient to process 4K videos at 30 frames per s. It is important to note that when the sector β is turned on or off, the frame rate of the system remains the same. This way, only energy consumption will change. The number of cycles required to process a block follows Eq. (4), where: (1) TB refers to the block size, (2) nLines is the number of lines in the block, and (3) nCand is the number of blocks compared by TZS.

$$\mathrm{Cycles} = 1 + \log_2 \mathrm{TB} + \mathrm{nLines} + \log_2 \mathrm{nCand} \tag{4}$$

Thus, the internal memory was designed to receive and deliver 64 bytes (8 × 8 sample blocks) per cycle and to operate at a frequency of 800 MHz, establishing the input of the adopted TZS hardware. With this frequency the system guarantees a processing rate enough for encoding 4K videos at 30 frames per s. This input size also matches the granularity of the DDRFC. Moreover, 8 × 8-sample block is the smallest square block size used in HEVC TZS. Higher TZS block sizes are divided in 8 × 8 samples to be processed by the DDRFC.

The calculation of static energy consumption considers the number of cycles required to process each CTU and also, the processing time in which the requested sector was active in the internal memory. Equation (5) shows the calculation performed to obtain the total static energy consumption of the internal memory, obtained from the sum of the static energy consumption of each sector. In this equation: (1) CT refers to the cycle time of the TZS hardware, (2) SEC represents the static energy consumption of the number of SRAM cells active in memory and (3) n is the maximum number of sectors that can be activated in memory.

$$\mathrm{Staticconsumption} = \sum_{i=0}^{i<n} (\mathrm{CT} \times \mathrm{Cycles}_i \times \mathrm{SEC}_i) \tag{5}$$

The following equations are used to calculate the dynamic energy consumption. The energy consumption (EC) related to the DRAM memory for reading (Re) and writing (Wr) operations for a number of words (D) when data reuse scheme is not considered, is defined as:

$$\mathrm{EC}_{\mathrm{Dram/TRe}} = E(\mathrm{DRAM}_{\mathrm{Re}}) \times D(\mathrm{Algorithm}_{\mathrm{TZS}}) \tag{6}$$

$$\mathrm{EC}_{\mathrm{Dram/TWr}} = E(\mathrm{DRAM}_{\mathrm{Wr}}) \times D(\mathrm{Frame}) \tag{7}$$

In Eqs. (6) and (7), E is the energy cost for an operation (read/write) in a specific memory. $\mathrm{Algorithm}_{\mathrm{TZS}}$ is the TZS algorithm used by the ME. Frame is the size of the frame (e.g., 1920 × 1080 pixels). The total EC, when data reuse scheme is not considered, is the sum of Eqs. (6) and (7), as in Eq. (8):

$$\mathrm{EC}_{\mathrm{Dram}} = \mathrm{EC}_{\mathrm{Dram/TRe}} + \mathrm{EC}_{\mathrm{Dram/TWr}} \tag{8}$$

The EC pertinent to the DRAM and SRAM memories for reading and writing operations when only data reused (DR) is considered is defined in Eqs. (9)–(11). In these equations, $\mathrm{Level}_{\mathrm{C}}$ is the Level C data reuse. The total EC considering the use of a data reuse scheme is summarized in Eq. (12).

$$\mathrm{EC}_{\mathrm{Dram}_{\mathrm{Re}}} = E(\mathrm{DRAM}_{\mathrm{Re}}) \times D(\mathrm{Level}_{\mathrm{C}}) \tag{9}$$

$$\mathrm{EC}_{\mathrm{Sram}_{\mathrm{Wr}}} = E(\mathrm{SRAM}_{\mathrm{Wr}}) \times D(\mathrm{Level}_{\mathrm{C}}) \tag{10}$$

$$\mathrm{EC}_{\mathrm{Sram}_{\mathrm{Re}}} = E(\mathrm{SRAM}_{\mathrm{Re}}) \times D(\mathrm{Algorithm}_{\mathrm{TZS}}) \tag{11}$$

$$\begin{aligned}\mathrm{EC}_{\mathrm{DR}} = {} & \mathrm{EC}_{\mathrm{Dram}_{\mathrm{Re}}} + \mathrm{EC}_{\mathrm{Sram}_{\mathrm{Wr}}} \\ & + \mathrm{EC}_{\mathrm{Sram}_{\mathrm{Re}}} + \mathrm{EC}_{\mathrm{Dram/TWr}}\end{aligned} \tag{12}$$

The EC related to the read operations for the complete solution including reference frame compressor and data reuse scheme, is defined by Eqs. (13) and (14). For write operations the EC is defined in Eqs. (15)–(17). Total EC of this scheme is obtained by Eq. (18).

$$\mathrm{EC}_{\mathrm{Dram}/C_{\mathrm{Re}}} = E(\mathrm{DRAM}_{\mathrm{Re}}) \times D(\mathrm{RFC}_{\mathrm{Dec}}) \tag{13}$$

$$\mathrm{EC}_{\mathrm{RFC}_{\mathrm{Dec}}} = E(\mathrm{RFC}_{\mathrm{Dec}}) \times D(\mathrm{Level}_{\mathrm{C}}) \tag{14}$$

$$\mathrm{EC}_{\mathrm{RFC}_{Enc}} = E(\mathrm{RFC}_{\mathrm{Enc}}) \times D(\mathrm{Frame}) \tag{15}$$

$$\mathrm{EC}_{\mathrm{C}_{\mathrm{Wr}}} = E(\mathrm{DRAM}_{\mathrm{Wr}}) \times D(\mathrm{FrameCod}) \tag{16}$$

$$\mathrm{EC}_{\mathrm{Dram}/\mathrm{C}_{\mathrm{Wr}}} = \mathrm{EC}_{\mathrm{RFC}_{\mathrm{Enc}}} + \mathrm{EC}_{\mathrm{C}_{\mathrm{Wr}}} \tag{17}$$

$$\begin{aligned}\mathrm{EC}_{\mathrm{system}} &= \mathrm{EC}_{\mathrm{Dram}/\mathrm{C}_{\mathrm{Re}}} + \mathrm{EC}_{\mathrm{RFC}_{\mathrm{Dec}}} \\ &\quad + \mathrm{EC}_{\mathrm{Sram}_{\mathrm{Wr}}} + \mathrm{EC}_{\mathrm{Sram}_{\mathrm{Re}}} + \mathrm{EC}_{\mathrm{Dram}/\mathrm{C}_{\mathrm{Wr}}}\end{aligned} \tag{18}$$

In these equations, FrameCod represents a frame encoded by the reference frame encoder. $\mathrm{RFC}_{\mathrm{Enc}}$ and $\mathrm{RFC}_{\mathrm{Dec}}$ represent the reference frame encoder and decoder energy from a reference frame coding.

6 Results and comparison with related works

This section presents the energy consumption and the coding efficiency results for the memory-aware dynamic control proposed in this work. Moreover, we compare the NM dynamic control strategy proposed in Sect. 4.3 against two static schemes called Static Sector Outer (SSO), where booth sectors β and α are always turned on, and Static Sector Inner (SSI), where only the sector α is turned on. Since NM strategy dynamically defines when sector β will be turned on or turned off, it is expected that the dynamic strategy will reach an energy consumption better than SSO and a coding efficiency better than SSI.

In the analysis of the TZS algorithm six HD 1080 video sequences were used. To validate the solution, new sequences were used together with the sequences used for the evaluation analysis. A new set of HD 1080p resolution video sequences were used and also a set of ultra-high definition (UHD) 2160p (4096 × 2160 and 3840 × 2160 pixels) videos were used. The set of UHD videos used in this section comprises videos of classes A1 and A2 of the Common Test Conditions (CTCs) [35]. These videos are: *Campfire*, *Drums*, *Tango*, *ToddlerFountain*, *CatRobot*, *DaylightRoad*, *RollerCoaster* and *TrafficFlow*. The additional 1080p sequences used were *BlueSky*, *InToTree*, *PedestrianArea*, *RushHour*, *Sunflower* and *Tractor*.

Table 2 65 nm CMOS SRAM-based SPMs hardware characteristics

	Sector α	Sectors $\alpha+\beta$	SR [−64, +64]
Number of banks	103	384	576
Bank size (B)	64	64	64
Access time (ns)	0.62	0.67	0.75
Cycle time (ns)	1.18	1.22	1.21
Dynamic energy (pJ/B)	0.08	0.08	0.09
Static energy (mJ/s)	0.71	2.64	4.56
Total capacity (KB)	6.4	24	36

6.1 NM energy consumption reduction

To generate the energy consumption results, a low power double data rate synchronous dynamic random-access memory (LPDDR SDRAM) model from Micron Technology was considered [36] as external memory. This memory consumes, per byte, 119.7 pJ and 116 pJ for reading and writing operations, respectively. The HP Cacti 6.5 tool [37] was used to simulate the 65 nm SRAM-based SPMs, which are used as internal memories.

The details of the SRAM-based SPM memory model are found in Table 2. The first column represents size, timing and energy when considering α sector only, which is able to store 17.89% of the SA. The second column represents sector α plus sector β, where sector α has 103 memory banks and sector β has 281 memory banks. Finally, the last column represents the complete SA, including 194 memory banks from sector γ.

As one can notice in Table 2, when sector β is turned off and only sector α remains active, the static energy consumed is reduced from 2.64 to 0.71 mJ/s, according to the change in the number of active SRAM cells in memory. The dynamic consumption is not changed, since it refers to the switching of the accessed cells and their buses and amplifiers.

With these results, it was possible to estimate the total energy consumption of the static and the proposed dynamic control strategy. These results are presented in Table 3 and were generated considering the 20 video sequences. In this table, the results of the dynamic and static management strategies are compared with the naive scheme (Naive column), i.e., the SR [−64, +64] does not employ any technique to reduce the memory bandwidth or energy consumption. The naive approach is not realistic in the current video coding scenario. However, we presented these results to highlight the contributions of the proposed

Table 3 Total energy consumption results (mJ)

Video	Naive ($\times 10^3$)	L.C RFC ($\times 10^3$)	L.C RFC SSI ($\times 10^3$)	L.C RFC NM ($\times 10^3$)	L.C RFC SSO ($\times 10^3$)
BasketballDrive	43.49	0.084	0.017	0.053	0.055
BQTerrace	10.87	0.063	0.019	0.028	0.043
Cactus	8.91	0.059	0.016	0.025	0.039
Kimono	31.63	0.076	0.017	0.047	0.051
ParkScene	9.84	0.061	0.017	0.026	0.041
Tennis	42.41	0.082	0.017	0.054	0.054
BlueSky	37.93	0.076	0.022	0.048	0.053
InToTree	51.83	0.088	0.022	0.057	0.060
PedestrianArea	29.97	0.072	0.019	0.040	0.049
RushHour	16.84	0.060	0.017	0.030	0.040
Sunflower	47.47	0.086	0.023	0.054	0.060
Tractor	49.65	0.093	0.027	0.061	0.066
Average 1080p	31.74	0.075	0.019	0.044	0.051
Std. dev.	15.60	0.012	0.003	0.013	0.008
CampfireParty	87.53	0.247	0.055	0.127	0.160
Drums	93.60	0.283	0.066	0.154	0.188
Tango	151.98	0.319	0.061	0.190	0.206
ToddlerFountain	152.64	0.359	0.084	0.232	0.241
CatRobot	37.80	0.217	0.053	0.086	0.142
DaylightRoad	112.42	0.287	0.062	0.156	0.187
RollerCoaster	204.47	0.359	0.065	0.223	0.232
TrafficFlow	52.78	0.234	0.055	0.100	0.151
Average 2160p	111.65	0.288	0.063	0.159	0.188
Std. dev.	52.10	0.051	0.009	0.050	0.034
Total average	63.70	0.160	0.037	0.090	0.106
Std. dev.	52.60	0.110	0.022	0.065	0.071

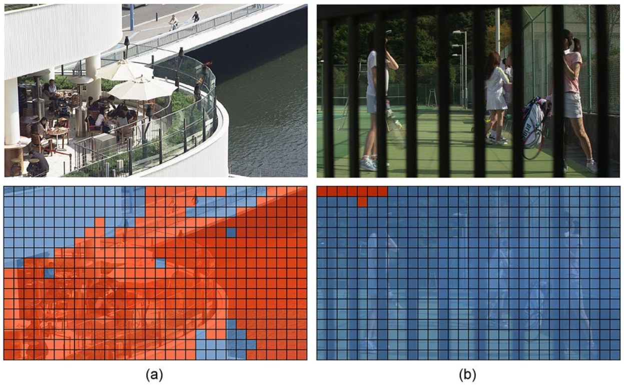


Fig. 8 NM CTU decision for frame 50 of the videos **a** BQTerrace and **b** Tennis. The red squares indicate where only sector α was used, while the blue squares indicate that sectors α and β were used.

approach and also to allow a fairer comparison with the related works that used this approach.

Furthermore, the results are compared to a baseline approach (L.C + RFC column), which employs the original SA (SR $[-64, +64]$) using Level C data reuse and DDRFC. Table 3 firstly presents the results for HD 1080 videos followed by the results for UHD videos. Table 3 also presents the average results for HD 1080 and UHD videos, the total average, and the standard deviation.

As one can notice in Table 3 the static schemes, SSI and SSO, and the NM dynamic scheme were integrated with the Level C data reuse and with the DDRFC. Thus, the real gains of these solutions can be observed.

Since SSI and SSO are static schemes they reached the best and the worst consumption energy gains, respectively, when compared with the original SR. This occurs because the static schemes cannot exploit the TZS context to define when sector β must be turned on or turned off. Thus, once the SSO system has the largest SA available, allowing a higher number of comparisons between blocks during the ME process, it presents the lowest energy consumption reduction when compared to the baseline scheme, 54.4 mJ or 33.9% on average. Likewise, the SSI system, that has the smallest SA available in the internal memory, it is the one with the higher energy consumption reduction. On average, the SSI reduces 123.6 mJ or 77.1% of energy consumption when compared to the baseline approach.

When compared to the raw data the SSO and SSI present an energy consumption reduction of 99.83% and 99.94%, respectively.

Figure 8 presents the energy corner case video results. Besides, an example of the NM CTU decision made to frame 50 of the videos BQTerrace (Fig. 8a) and Tennis Fig. 8b. In Fig. 8 the red squares indicate where sector α was used, while the blue squares indicate that sectors α and β were used. In Fig. 8a 78.4% of the BQTerrace frame was encoded with sector β turned off, achieving high memory energy savings. However, Fig. 8b shows that just in 1.5% of the cases, the sector β was turned off, i.e., the energy savings are practically the same as those obtained by the SSO scheme. As one can notice in Table 3 the difference between NM and SSO schemes for the Tennis sequence is 0.24%. This occurs because the Tennis sequence presents characteristics such as high motion and texture, which lead to less access to the SA central region than other videos. This effect can also be seen in Fig. 3 (Sect. 4.1).

As expected, the system with NM dynamic control scheme presents average energy consumption results between the two static systems: lower than SSO (−16.4 mJ) and higher than SSI (+52.8mJ. Compared to the baseline scheme, the NM system reaches energy consumption savings of 70.8 mJ or 44.1% on average. Besides, when compared to the naive approach, the NM system reaches an energy consumption savings of 99.86% on average. However, it is important to note the techniques that enable energy reduction are likely to lead to coding efficiency losses, as will be discussed in the next section.

6.2 Coding efficiency results

Since the proposed scheme controls which sectors of the memory will be turned on or turned off, the consequence in terms of coding efficiency is that the TZS algorithm will not have the whole original SA available in the search process. This restriction, naturally, will cause some impacts in terms of coding efficiency and this section presents an evaluation of these impacts.

The coding efficiency results use the Bjøntegaard Delta Bitrate (BD-rate) metric, which can be interpreted as how much lower or higher (in percentage) is the bitrate required to represent the encoded video with some encoder alteration, when compared with the original encoder, considering the same objective quality in both cases [38]. The coding efficiency results are shown in Table 4. Again, these results were compared with the baseline approach. As in Table 3, Table 4 presents the results for HD 1080 videos and the results for UHD videos. Table 4 also presents average results for HD 1080 and for UHD videos, as well as the total average, the minimum and maximum values and the standard deviation.

Table 4 BD-rate results for the static (SSI and SSO) and NM dynamic scheme

Video	Level C DDRFC SSI (%)	Level C DDRFC NM (%)	Level C DDRFC SSO (%)
Basketball Drive	4.78	0.360	0.210
BQTerrace	0.36	0.300	−0.120
Cactus	1.10	−0.130	−0.040
Kimono	2.10	−0.030	−0.050
ParkScene	0.18	−0.520	−0.490
Tennis	4.98	0.140	0.100
BlueSky	2.73	0.490	−0.420
InToTree	15.02	0.000	2.380
PedestrianArea	8.27	0.320	0.550
RushHour	2.69	0.440	0.000
Sunflower	4.81	0.180	−0.030
Tractor	3.28	0.000	0.500
Average 1080p	4.19	0.130	0.220
Std. dev.	3.93	0.273	0.716
CampfireParty	5.18	1.340	0.970
Drums	1.80	0.650	0.450
Tango	10.49	0.740	0.560
ToddlerFountain	2.67	0.510	0.610
CatRobot	0.95	0.050	0.090
DaylightRoad	13.96	0.760	0.660
RollerCoaster	27.03	−0.380	−0.520
TrafficFlow	11.07	1.830	−0.370
Average 2160p	9.140	0.690	0.310
Std. dev.	8.12	0.644	0.491
Total average	6.17	0.350	0.250
Min	0.18	−0.520	−0.520
Max	27.03	1.830	2.380
Std. dev.	6.44	0.530	0.640

The results of SSI scheme in Table 4 presented a highest loss in coding efficiency, as expected, reaching an average increase of 6.17% in BD-rate. The higher is the video resolution and the higher is the video movement, the higher is the BD-rate degradation caused by the SSI scheme. These videos need to use large motion vectors in the TZS to best predict the current block. Since SSI deals with a very limited SA (only sector α is available), the prediction results are far from the optimal ones and the degradation in coding efficiency is higher.

In Table 4 one can notice that *SSO* scheme shows the smallest degradation in the coding efficiency with an average increase of 0.25% in BD-rate. This degradation occurs because even with sectors α and β turned-on, the external sector is not considered, as explained in Sect. 4.2. In some situations, the SSO scheme presented small gains in coding efficiency (negative values of BD-rate), and this occurs mainly because the largest motion vectors are not allowed since the external sector is not available. With small vectors, the motion vector prediction tools [27] will require fewer bits to represent these vectors and, then, the impact in the bit rate can be reduced even with a prediction result a little bit worse.

The NM dynamic management strategy achieved better coding efficiency results than SSI for all videos, reaching an average BD-rate degradation of 0.35%. This result was expected since the NM dynamic control allows the use of sector β when the neighbor CTUs required the use of this sector, and the SSI scheme always considers sector β is turned off. Some NM results are even better than those reached by the SSO. The TZS is a heuristic algorithm and, by imposing some restrictions, it is possible that the results are slightly better.

The small granularity of NM control allows higher independence of sector β switching on/off between CTUs and presents a higher occurrence of BD-rate values outside the range presented by SSO and SSI. In addition, as this solution is based on information from neighboring CTUs already encoded in the current and past frames, to decide the situation of sector β for the CTU being encoded, this solution can reach low variations in BD-rate.

Analyzing the relationship between the energy consumption and the coding efficient, the NM presents important gains in terms of energy consumption (99.86% on average) with a very small coding efficiency degradation (0.35% on average). This means that *NM* reduces, on average, 15.5% of the energy consumption when compared with SSO at a cost of only 0.1% in BD-rate on average. On the other way, NM has a coding efficiency degradation 17.6 times lower than the SSI, with an increase of 2.4 times in the energy consumption. The coding efficiency degradation reached by SSI is prohibitive in a current video coding scenario.

Therefore, the results of energy consumption reduction combined with the BD-rate results show that the TZS dynamic management strategy improved the video encoding process. Once exploring the video characteristics during the video encoding phase, it is possible to turn off a memory sector when it becomes less relevant to the TZS process. Thus, dynamically turning off a memory sector, it was possible to obtain a good relationship between energy consumption and coding efficiency.

Table 5 Comparison with related works

Sol.	Energy saving	BD-rate loss	BW saving	ME Alg.	#Test Seq.
NM	99.8%	0.35%	94%	TZS	20
[22]	NA	0.70%	NA	TZS	22
[26]	NA	1.10%	78%	TZS	7
[27]	NA	1.80%	54%	MHGS	4
[21]	NA	0.20%	NA	FS	4
[23]	NA	0.42%	43%	FS	5
[24]	NA	0.17%	91%	FS	4
[25]	NA	0.30%	90%	FS	4

6.3 Comparison with related works

This section presents a comparison among related works presented in Sect. 2 and our system with dynamic memory management (*NM*). Only solutions implemented in HEVC test model were considered. Table 5 presents the comparative results with seven related works. In this comparison, five metrics were used: (1) energy saving, (2) BD-rate, (3) memory bandwidth saving (BW saving), (4) ME algorithm (ME Alg.), and (5) amount of test sequences used (#Test Seq.).

As one can notice in the second column of Table 5, the state-of-the-art related works lack energy consumption figures, thus precluding a fair energy comparison presentation. However, our system reaches 99.8% (or 500 times) of memory energy savings over the simplest scheme, considering the 20 test sequences.

When the BD-rate results are compared, our approach achieves competitive results, once the related works present BD-rate loss from 0.02 to 1.8%, on average. Compared with works that employ TZS ME algorithm, such as [22] and [26] our system with the *NM* scheme presents the smallest BD-rate loss, 0.35% for NM, against 0.70% and 1.1% in [22] and [26], respectively.

The memory bandwidth savings results are presented in the fourth column of Table 5. The proposed system achieves a memory bandwidth savings of 94% on average for the test sequences. When compared to [26], which is the only related work that uses the TZS algorithm and presents memory bandwidth savings results, our system presents an improvement of 16% in memory bandwidth savings. Besides, the proposed system achieves 40% more memory bandwidth savings than [27]. Furthermore, the proposed system presents a coding efficiency three times lower than [26] and our video test set is almost three times higher than [26], including HD 1080p and UHD test sequences.

Moreover, our system achieves better memory bandwidth savings than related works that apply dynamic search range algorithms with the full search algorithm. This is because we combined three techniques (reference frame compression, data reuse, and dynamic management) to achieve high memory bandwidth savings.

7 Conclusions

This work presented a search memory design and TZS dynamic memory management, which explores power gating at the memory sector level. To better understand the TZS access behavior, this work presented a statistical analysis of the pattern of memory accesses performed by the TZS algorithm during the ME process for video encoding. From this quantitative study, we confirmed that most accesses are made in the central region of the SA, as initially expected. Interestingly, 95% of the most accessed samples are in a region that corresponds to 66.65% of the SA. Based on this analysis, we perform reductions in the SA to verify the impact on the coding efficiency when removing some of the less accessed samples from the SA. This evidence led to our proposal of multi-sector scratchpad memory, which can perform power gating when accesses to sector β become not relevant to the ME process. To substantially save energy, this work presented the neighbors management, a dynamic memory management strategy to decide to turn off or not sector β. This dynamic management was integrated with a reference frame compressor and a Level C data reuse scheme, both presented in prior works. The results were obtained from 20 test video sequences, 1080p and 2160p, and compared with a naive approach and with a baseline scheme, which employs the reference frame compressor and Level C data reuse scheme. Our system with dynamic management obtained a coding efficiency loss of just 0.35% and an energy consumption reduction of 99.8% (or 500 times reduction) when compared with the naive approach, and 44.1% when compared to the baseline scheme. Our system presents better memory bandwidth/energy savings and coding efficiency results when compared to related works.

Acknowledgements This study was supported by the Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS), Fundação CAPES Finance Code 01, FAPERGS, and CNPq Brazilian agencies for R&D support.

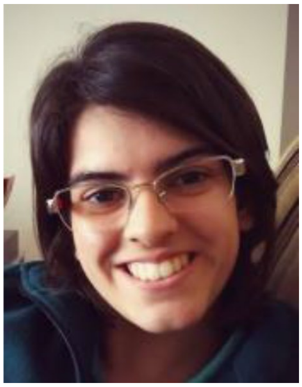
References

1. Cisco: Cisco visual networking index: forecast and trends, 2017–2022 (2018). https://www.cisco.com/. Accessed 20 Dec 2019
2. Nokia: Network traffic insights in the time of COVID-19 (2020). https://www.nokia.com/blog/network-traffic-insights-time-covid-19-april-9-update/. Accessed 6 July 2020
3. Mutlu, O., Subramanian, L.: Research problems and opportunities in memory systems. Supercomput. Front. Innovat. **1**(3), 19–55 (2015)
4. H.265: ITU-T recommendation H.265: high efficiency video coding, audiovisual and multimedia systems (2013). https://www.itu.int/rec/T-REC-H.265. Accessed 20 Dec 2019
5. Sullivan, G.J., Ohm, J., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. Trans. Circ. Syst. Video Technol. **22**(12), 1649–1668 (2012)
6. Shafique, M., Garg, S., Henkel, J., Marculescu, D.: The EDA challenges in the dark silicon era: temperature, reliability, and variability perspectives. In: Design Automation Conference (2014). https://doi.org/10.1145/2593069.2593229
7. Correa, G., Assuncao, P.A., Agostini, L.V., da Silva Cruz, L.A.: Fast HEVC encoding decisions using data mining. Trans. Circ. Syst. Video Technol. **25**(4), 660–673 (2015). https://doi.org/10.1109/TCSVT.2014.2363753
8. Tikekar, M., Huang, C., Juvekar, C., Sze, V., Chandrakasan, A.P.: A 249-Mpixel/s HEVC video-decoder chip for 4K ultra-HD applications. J. Solid-State Circ. **49**(1), 61–72 (2014)
9. Khan, M.U.K., Shafique, M., Henkel, J.: AMBER: adaptive energy management for on-chip hybrid video memories. In: International Conference on Computer-Aided Design (2013). https://doi.org/10.1109/ICCAD.2013.6691150
10. Tang, X., Dai, S., Cai, C.: An analysis of TZSearch algorithm in JMVC. In: International Conference on Green Circuits and Systems (2010). https://doi.org/10.1109/ICGCS.2010.5543008
11. Mativi, A., Monteiro, E., Bampi,S.: Memory access profiling for HEVC encoders. In: Latin American Symposium on Circuits Systems (2016). https://doi.org/10.1109/LASCAS.2016.7451055
12. Zatt, B., Shafique, M., Sampaio, F., Agostini, L., Bampi, S., Henkel, J.: Run-time adaptive energy-aware motion and disparity estimation in multiview video coding. In: Design Automation Conference (2011). https://doi.org/10.1145/2024724.2024950
13. Shafique, M., Zatt, B., Walter, F.L., Bampi, S., Henkel, J.: Adaptive power management of on-chip video memory for multiview video coding. In: Design Automation Conference (2012). https://doi.org/10.1145/2228360.2228516
14. Tuan, J.-C., Chang, T.-S., Jen, C.-W.: On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture. Trans. Circ. Syst. Video Technol. **12**(1), 61–72 (2002). https://doi.org/10.1109/76.981846
15. Silveira, D., Povala, G., Amaral, L., Zatt, B., Agostini, L., Porto, M.: Efficient reference frame compression scheme for video coding systems: algorithm and VLSI design. J. Real-Time Image Process. **16**(2), 391–411 (2019)
16. Lian, X., Liu, Z., Zhou, W., Duan, Z.: Parallel content-aware adaptive quantization-oriented lossy frame memory recompression for HEVC. Trans. Circ. Syst. Video Technol. **28**(4), 958–971 (2018)
17. Willème, A., Macq, B., Descampe, A., Rouvroy, G.: Power-aware HEVC compression through asymmetric JPEG XS frame buffer compression. In: International Conference on Image Processing (2018). https://doi.org/10.1109/ICIP.2018.8451539
18. Lee, Y.-H., Chen, C.-C., You, Y.-L.: Design of VLSI architecture of autocorrelation-based lossless recompression engine for memory-efficient video coding systems. Springer Circ. Syst. Signal Process. **3**(2), 459–482 (2014)
19. Lian, X., Liu, Z., Zhou, W., Duan, Z.: Lossless frame memory compression using pixel-grain prediction and dynamic order entropy coding. Trans. Circ. Syst. Video Technol. **26**(1), 223–235 (2016)
20. Chen, C.-H., Huang, C.-T., Chen, Y.-H., Chen, L.-G.: Level C+ data reuse scheme for motion estimation with corresponding coding orders. Trans. Circ. Syst. Video Technol. **16**(4), 553–558 (2006). https://doi.org/10.1109/TCSVT.2006.871388
21. Dai, W., Au, O.C., Li, S., Sun, L., Zou, R.: Adaptive search range algorithm based on Cauchy distribution. In: Visual Communications and Image Processing (2012). https://doi.org/10.1109/VCIP.2012.6410741

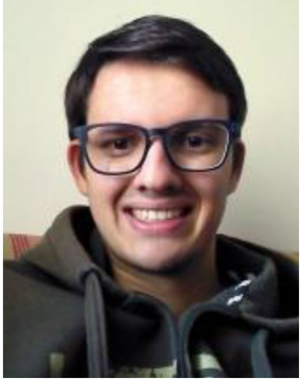
22. Du, L., Liu, Z., Ikenaga, T., Wang, D.: Linear adaptive search range model for uni-prediction and motion analysis for bi-prediction in HEVC. In: International Conference on Image Processing (2014). https://doi.org/10.1109/ICIP.2014.7025745
23. Chien, W., Liao, K., Yang, J.: Enhanced AMVP mechanism based adaptive motion search range decision algorithm for fast HEVC coding. In: International Conference on Image Processing (2014). https://doi.org/10.1109/ICIP.2014.7025750
24. Li, Y., Liu, Y., Yang, H., Yang, D.: An adaptive search range method for HEVC with the k-nearest neighbor algorithm. In: Visual Communications and Image Processing (2015). https://doi.org/10.1109/VCIP.2015.7457794
25. Ji, X., Jia, H., Liu, J., Xie, X., Gao, W.: Computation-constrained dynamic search range control for real-time video encoder. Image Commun. **31**, 134–150 (2015)
26. Pakdaman, F., Gabbouj, M., Hashemi, M.R., Ghanbari, M.: Fast motion estimation algorithm with efficient memory access for HEVC hardware encoders. In: European Workshop on Visual Information Processing (2018). https://doi.org/10.1109/EUVIP.2018.8611766
27. Singh, K., Rafi Ahamed, S.: Low power motion estimation algorithm and architecture of HEVC/H.265 for consumer applications. Trans. Consumer Electron. **64**(3), 267–275 (2018). https://doi.org/10.1109/TCE.2018.2867823
28. Huang, Y.-W., Chen, C.-Y., Tsai, C.-H., Shen, C.-F., Chen, L.-G.: Survey on block matching motion estimation algorithms and architectures with new results. J. VLSI Signal Process. Syst. Signal Image Video Technol. **42**, 297–320 (2006)
29. High Efciency Video Coding (HEVC): Reference software (2018). https://hevc.hhi.fraunhofer.de. Accessed 15 Dec 2019
30. Afonso, V., Conceição, R., Saldanha, M., Braatz, L., Perleberg, M., Corrêa, G., Porto, M., Agostini, L., Zatt, B., Susin, A.: Energy-aware motion and disparity estimation system for 3D-HEVC with run-time adaptive memory hierarchy. Trans. Circ. Syst. Video Technol. **29**(6), 1878–1892 (2019)
31. Bossen, F., Bross, B., Suhring, K., Flynn, D.: HEVC complexity and implementation analysis. Trans. Circ. Syst. Video Technol. **22**(12), 1685–1696 (2012)
32. Jeong, K., Kahng, A.B., Kang, S., Rosing, T.S., Strong, R.: MAPG: memory access power gating. In: Design, Automation Test in Europe Conference (2012). https://doi.org/10.1109/DATE.2012.6176651
33. Bossen, F.: Common test conditions and software reference configurations. Document JCTVC-L1100, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Joint Collaborative Team on Video Coding (JCT-VC) (2013)
34. Amaral, L., Povala, G., Porto, M., Silveira, D., Bampi, S.: Memory energy consumption analyzer for video encoder hardware architectures. In: International Conference on Electronics, Circuits and Systems (2016). https://doi.org/10.1109/ICECS.2016.7841203
35. Sharman, K., Suhring, K.: Common test conditions for HM. Document JCTVC-Z1100, ITU-T SG16 and ISO/IEC/JTC1/SC29/WG11 Joint Collaborative Team on Video Coding (JCT-VC) (2017)
36. Micron: Micron MT46H64M16LF: 1 Gb DDR SDRAM (2019). https://www.micron.com/. Accessed 15 Dec 2019
37. CACTI: HP CACTI 6.5 (2018). https://www.hpl.hp.com/research/cacti/. Accessed 15 Dec 2019
38. Bjontegaard, G.: Improvements of the BD-PSNR model. In: VCEGAI11, ITU-T SG16/Q6 VCEG 35th meeting, Berlin, Germany, pp. 16–18 (2008)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dieison Soares Silveira received the B.S. and M.S. degree in Computer Science from Federal University of Pelotas (UFPel), Pelotas, RS, Brazil, in 2013 and 2015, respectively. He is a Professor at Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS), Campus Canoas, Brazil, and he is pursuing the Ph.D. degree in microelectronics from the PGMICRO (Graduate Program on Microelectronics), Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil. Also, he is a collaborate researcher of the Video Technology Research Group (ViTech) and the Group of Architectures and Integrated Circuits (GACI). His research interests include algorithms, VLSI architectures, approximate computing, and energy management for video coding systems.

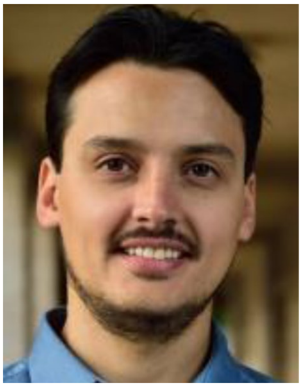


Lívia Amaral received the B.S. and M.S. degrees in computer science from the Federal University of Pelotas (UFPel), Brazil, in 2015 and 2018, respectively. She is a member of the Video Technology Research Group (ViTech) and the Group of Architectures and Integrated Circuits (GACI). Her research interests include algorithms and architectures for image and video compression focusing in energy consumption reduction.



Guilherme Povala received the B.S. and M.S. degrees in computer science from the Federal University of Pelotas (UFPel), Brazil, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in Computer Science from UFPel, Brazil, and the Ph.D. degree in Biochemestry from Federal University of Rio Grande do Sul (UFRGS), Brazil. He is a member of the Zimmer Lab, the Video Technology Research

Group (ViTech) and the Group of Architectures and Integrated Circuits (GACI). His research interests include algorithms for image and video compression. Currently, he works with neuroimaging (analysis and processing of medical PET and MRI images) and with the application of artificial intelligence techniques for the diagnosis and prediction of events related to neurodegenerative diseases, such as Alzheimer and Parkinson.



Bruno Zatt received his B.E. and M.S. in Computer Engineering from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2006 and 2008, respectively. He has received his Ph.D. degree on Microelectronics from the PGMICRO (Graduate Program on Microelectronics) at the same university in 2012 with "summa cum laude"' distinction. Currently, Bruno Zatt is a Professor at the Federal University of Pelotas (UFPel), Pelotas, Brazil, and a member of the Group of Architectures and Integrated Circuits (GACI) and the Video Technology Research Group (ViTech). He has 15+ years research experience on algorithms and hardware architectures for video processing including 3 years as an intern researcher at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, and 1-year experience as a visiting professor at the University of California, Irvine, USA. He has published over 100+ papers in international journals/conferences and one book named "3D video coding for embedded devices". He holds the status of CNPq (National Council for Scientific and Technological Development) productivity research fellow. Prof. Zatt also contributes as a member of the Brazilian Committee on Audio, Image, Multimedia and Hypermedia Coding.

Luciano Volcan Agostini received the M.S. and Ph.D. degrees in Computer Science from Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2002 and 2007 respectively. He is a professor since 2002 at Federal University of Pelotas (UFPel), Brazil, where he leads the Video Technology Research Group (ViTech) and the Group of Architectures and Integrated Circuits (GACI). He is advisor at the UFPel Master and Doctorate in Computer Science courses. He was the Executive Vice President for Research and Graduate Studies of UFPel from 2013 to 2017. He has more than 200 published papers in respected international journals and conferences. His research interests include 2D and 3D video coding, algorithmic optimization, arithmetic circuits, and dedicated hardware design. Dr. Agostini is a Member of SBC and SBMicro Brazilian societies. He is also a Brazilian Distinguished Researcher through a CNPq PQ-1D grant.

Marcelo Schiavon Porto received the M.S. and Ph.D. degrees in computer science from Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2008 and 2012, respectively. He is currently a Professor with the Federal University of Pelotas (UFPel), Brazil, and a member of the Video Technology Research Group (ViTech) and the Group of Architectures and Integrated Circuits (GACI). He is currently the Coordinator of the Postgraduate Program in Computing (PPGC) of UFPel. He also has been holding the status of CNPq (National Council for Scientific and Technological Development) Productivity Research Fellow, since 2016. His research interests include video coding, motion estimation algorithms, complexity reduction, and energy-efficient VLSI design for video coding.

Sergio Bampi received the B.Sc. degree in electronics and the B.Sc. degree in physics from Federal University of Rio Grande do Sul (UFRGS) in 1979, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, in 1982 and 1986, respectively. He was the Technical Director of the Microelectronics Center CEITEC from 2005 to 2008, and was the President of the FAPERGS Research Funding Foundation. He was a Visiting Research Professor at Stanford University from 1998 to 1999, and the Director of the National Supercomputer Center from 1993 to 1996. He is currently a Full Professor with the UFRGS Informatics Institute. His research interests include IC design and modeling, mixed signal and RF CMOS circuit's design, architectures and SoCs for image and video processing, nano-CMOS devices, ultra-low power digital CMOS design, and dedicated complex algorithms and architectures. He has coauthored more than 390 papers in the above fields and in MOS devices, circuits, technology and CAD. He served as the Council Member of the Brazilian Computer Society, and the Brazilian Society for Advancement of Science. He served as the President of the scientific societies SBPC and SBMICRO from 2002 to 2004.