Energy-Efficient Memory Hierarchy for Motion and Disparity Estimation in Multiview Video Coding

Felipe Sampaio[2], Bruno Zatt[1,2], Muhammad Shafique[1], Luciano Agostini[3], Sergio Bampi[2], Jörg Henkel[1]
[1]Karlsruhe Institute of Technology (KIT), Chair for Embedded Systems, Karlsruhe, Germany
[2]Federal University of Rio Grande do Sul (UFRGS), Informatics Institute, PPGC-PGMICRO, Porto Alegre, Brazil
[3]Federal University of Pelotas (UFPel), GACI, Pelotas, Brazil
{bzatt, bampi, felipe.sampaio}@inf.ufrgs.br, agostini@inf.ufpel.edu.br, {muhammad.shafique, henkel}@kit.edu

***Abstract*— This work presents an energy-efficient memory hierarchy for Motion and Disparity Estimation on Multiview Video Coding employing a Reference Frames-Centered Data Reuse (RCDR) scheme. In RCDR the reference search window becomes the center of the motion/disparity estimation processing flow and calls for processing all blocks requesting its data. By doing so, RCDR avoids multiple search window retransmissions leading to reduced number of external memory accesses, thus memory energy reduction. To deal with out-of-order processing and further reduce external memory traffic, a statistics-based partial results compressor is developed. The on-chip video memory energy is reduced by employing a statistical power gating scheme and candidate blocks reordering. Experimental results show that our reference-centered memory hierarchy outperforms the state-of-the-art [7][13] by providing reduction of up to 71% for external memory energy, 88% on-chip memory static energy, and 65% on-chip memory dynamic energy.**

***Index Terms* — Multiview Video Coding, MVC, 3D-Video, Low-Power Design, On-Chip Video Memory, Application-Aware DPM, Memory Hierarchy, Energy Efficiency, Motion Estimation, Disparity Estimation.**

I. INTRODUCTION

Increasing demands for immersive multimedia systems have driven the popularization of the 3D-video technology that embraces a wide range of applications such as cinema, automotive, telepresence, 3D (mobile) camcorders, etc. 3D videos are based on the multiview concept [1] where multiple independent cameras record the same 3D scene from different viewpoints. Multiple video streams represent huge amount of data that must be processed and encoded before storage or transmission. The Multiview Video Coding (MVC) standard [2] provides 20-50% increased coding efficiency in comparison to H.264/AVC [3]. This is due to the inter-view prediction using Disparity Estimation (DE), which results in significant increase in the encoding complexity and energy consumption. Along with the Motion Estimation (ME), the DE represents about 90% of the encoder energy consumption [13]. Therefore, ME/DE is the main optimization focus for energy reduction in the MVC encoders.

ME/DE is used to search an image region (candidate block) that presents the best matching in the reference frame (previously decoded frames). The search is performed within a *search window* using a search algorithm like TZ Search [16]. This search window is typically fetched from external memory and stored in an on-chip video memory. Even for fast search algorithms, frequent memory accesses and large on-chip memory requirements lead to high energy consumption. Moreover, since the memory energy contributes to approximately 90% of the total ME/DE energy consumption [13], on/off-chip memory energy reduction is mandatory to meet the constraints and design requirements posed by the mobile battery-powered devices.

Recent works have proposed solutions to reduce the number of external memory accesses by employing current macroblock (MB)-centered data reuse (MBDR) schemes [6][7]. However, candidate-based and search window-based MBDR data reuses suffer from increased external memory traffic and on-chip memory, respectively (mainly for MVC that demands multiple references and at least a 193x193 search window [11]). Looking forward to jointly address on/off-chip memory energy issues, asymmetric search window [11], search window follower [12], and dynamic search window formation [13] schemes were proposed. Asymmetric search window, however, might lead to undesirable quality losses in case of vertical motion/disparity scenarios while dynamic search window leads to irregular memory access and on-chip memory misses. A novel perspective to face ME/DE memory issues was proposed in [9] and [8]. In these solutions a reference frame region is fetched and multiple MBs are processed using the available data. It eliminates reference frame retransmissions at the cost of partial search results (motion/disparity vectors and similarity) storage due to out-of-order processing. This solution, however, does not properly consider the impact of partial results memory traffic and on-chip video memory size.

The main challenge is to *jointly minimize on-chip and off-chip energy consumption in order to reduce the overall energy related to ME/DE on MVC*.

Previous works also aimed to reduce the memory related energy consumption of ME/DE on MVC encoders. In [13], we presented a low power ME/DE architecture that uses the concept of search map and dynamic search window formation. The work [15] extends the previous work by employing a multi-sleep state model on-chip memory to reduce the leakage energy. Latest, in [14] we proposed an on-chip memory power management based on such techniques, like the search direction elimination, to reduce even more the on-chip memory energy. Now, in this work we proposed jointly on-chip and off-chip memory energy savings techniques. Besides, different from previous work, we proposed a regular off-chip memory access pattern to exploit the burst reads (regarding to the DDR memories) to be more energy-efficient than the irregular-pattern previous solutions.

A. Our Novel Contributions

This work proposes a novel reference-centered memory hierarchy for ME/DE on MVC targeting low-energy consumption at both on-chip storage and off-chip memory access. The memory hierarchy is composed of an on-chip video memory, an on-chip memory power gating control, and a partial search results compressor. Additionally, a customized search control unit is proposed to exploit the search behavior to achieve further energy reduction.

978-3-9815370-0-0/DATE13/©2013 EDAA

Our novel contributions are:

- **Reference-Centered Memory Hierarchy** that employs a Reference-Centered Data Reuse (RCDR) scheme. It makes the reference frames the center of processing order to avoid search window retransmission and to eliminate the need to simultaneously store on-chip multiple search windows. A novel memory access scheduling and an energy-efficient on-chip memory organization are proposed.
- **Statistics-Based Partial Results Compressor**: The out-of-order processing inherent to RDCR imposes partial results (motion/disparity vectors and SAD) storage. Statistically defined non-uniform quantization and Huffman coding are employed for partial results compression.
- **On-Chip Video Memory**: The on-chip memory is organized in multiple SRAM banks featuring line-level power gating capability. At run-time, search window regions that are less likely to be used are power-gated. Additionally, the candidate blocks coding order is rearranged to minimize on-chip memory line switching and, consequently, the dynamic energy consumption.

Paper Organization: Section II presents a motivational ME/DE memory case study. Section III describes the proposed memory hierarchy and data reuse scheme. In Section IV, the experimental results and comparison to state-of-the-art are presented. Section V concludes the paper.

II. ME/DE Memory Analysis: A Case Study

In this section a ME/DE memory analysis is presented to motivate the use of Reference-Centered Data Reuse (RCDR). Fig. 1. shows the number of accesses for one ME/DE search on one given reference frame. It is possible to note that some pixels in the reference frame are accessed more than 1000 times. The scenario becomes more challenging for MVC because its prediction structure demands the search on multiple reference frames to achieve efficient compression; see Fig. 2.

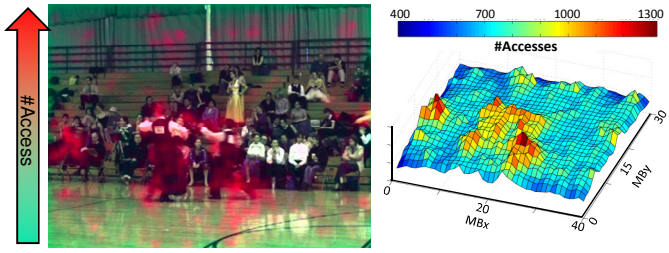

Fig. 1. Memory Access Analysis

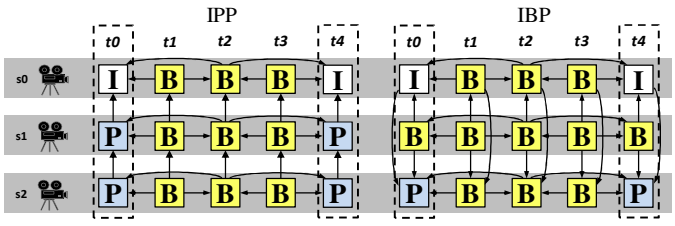

Fig. 2. MVC Prediction Structure, MB-Centered Perspective

Fig. 2. presents the MVC hierarchical prediction structure at the traditional current MB-centered perspective, where the origin of the arrows represent the current processed frame and the arrow head points to the requested reference frame. Examples for "IPP" and "IBP" view coding orders are provided. For simplicity, we use 3 views and Group of Picture (GOP, interval between I frames) equals 4; however, for real applications the GOP is typically higher and more views may be used [3]. Note that many frames are referenced multiple times to predict other frames. For instance, frame "S0T0" is referenced three and four times for "IPP" and "IBP" orders, respectively. It leads to a multiplication on the number of memory accesses (for MB-centered search) shown in Fig. 1. Moreover, as MVC requires multiple reference frames, they must be simultaneously stored on-chip. For instance, frame "S1T1" requires four references to perform the complete ME/DE.

Naturally, reference data is not fetched on demand from external memory every time it is required. Data reuse techniques like [6][7][11][12][13] reduce the external memory accesses by partially storing the data on-chip. In Fig. 3. the design spaces corner cases are presented for the on-chip vs. off-chip tradeoff considering 8 views, GOP=8, "IPB" and search window equals 193x193. In case no on-chip memory is employed, a huge external bandwidth is required. In contrast, if the all reference frames are stored on-chip, the on-chip memory grows drastically. Level-C [7] (a well known search window-based data reuse scheme) presents an intermediate compromise in this tradeoff but large on-chip memory and numerous external memory accesses are required due to MBDR limitation. To address this issue, our memory hierarchy employs a reference-centered data reuse scheme.

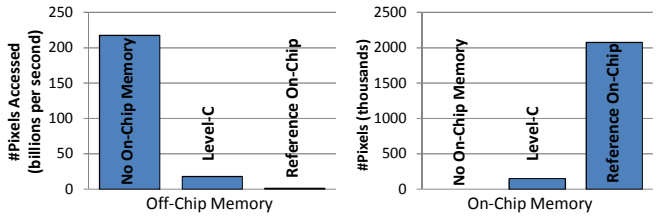

Fig. 3. Off-Chip vs. On-Chip Memory Tradeoff

III. Reference-Centered Memory Hierarchy

Fig. 4. presents the architecture of our video memory hierarchy employing reference-centered data reuse scheme. To control the ME/DE search and, consequently, the memory access pattern a Search Control unit is defined. Firstly, the Search Control sends search window requests to the memory. These requests are represented in video positions format. Therefore Address Generation Units (AGUs) are used to translate the requests to multiple external memory positions. Once the data is stored on-chip, a burst of candidate block positions is generated by the Search Control according to the TZ Search algorithm [16]. These candidate positions are rearranged by the energy-aware candidates merging unit in order to reduce the number of on-chip memory line switching. An on-chip memory power gating control monitors the search statistics and power-gates the on-chip memory lines accordingly. The candidates are processed by an array of processing elements (not described in this work). The best matching candidate and its SAD are forwarded to the search control. Due to the out-of-order processing inherent to RCDR, the temporary motion/disparity vectors and SAD (Sum of Absolute Differences) values must be stored for mode decision. These partial results are compressed using statistic-based non-uniform quantization and Huffman coding. As the partial results compressor employs variable-length coding, the partial results data is only sent to external memory once the local buffer is full. A specific AGU is implemented for partial results data.

Aware of the possible memory contention created by multiple AGUs requesting access to the external memory, we propose the

fixed memory access scheduling presented in Fig. 5. Firstly, the Current MBs are fetched in order to allow the ME/DE processing start (part of the search window is already on-chip) followed by the missing search window column reading. In the following, if the partial results buffer is full, partial results are sent to external memory. Finally, the memory is available for other MVC blocks. Note that the number of current MBs (*D*) and search window blocks (*n*) vary at frame level changing the duration of the schedule intervals. Still, the schedule is not affected.

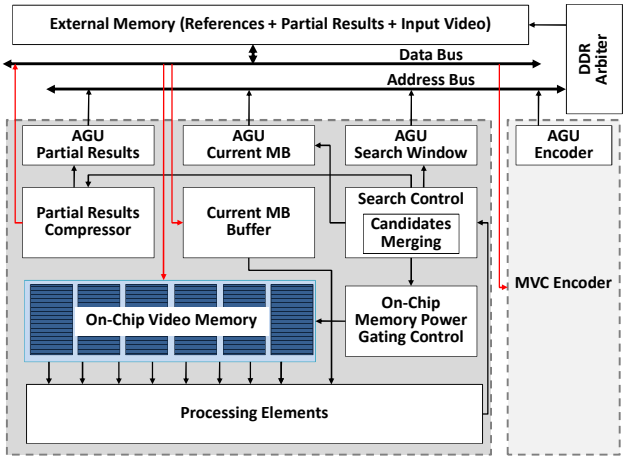

Fig. 4. Reference-Centered Memory Architecture

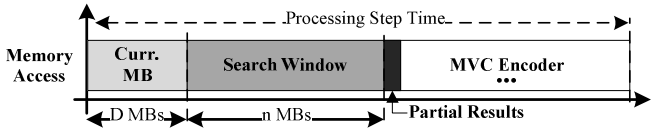

Fig. 5. External Memory Access Scheduling

A. Reference-Centered Data Reuse

The RCDR uses inverted dependence logic between reference frames and current MB. In this approach, the reference search window is fetched from external memory and those MBs requiring that specific data are processed. In other words, the reference data "calls" the MBs to be processed. For this reason, we define the term *dependent frames* for those frames "called" by a given reference frame.

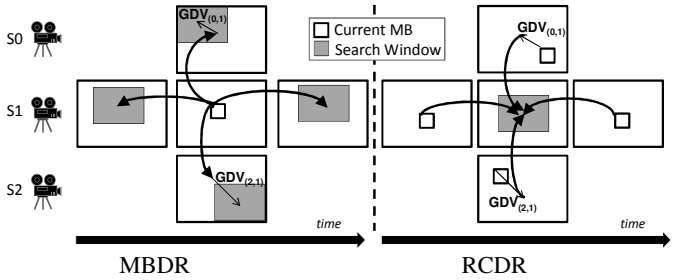

Fig. 6. MBDR vs. RCDR Data Reuses for ME/DE

Fig. 6. depicts the distinctions between search window-based MBDR and RCDR. Observe that in MBDR each current MB requests up to four search windows demanding 4x increased on-chip memory. Additionally, as those search windows are required more times in the future (to encode other frames), external memory data retransmission is needed. In contrast, on-chip storage of a single search window is required for RCDR resulting in reduced on-chip memory. Moreover, in RCDR the search window is requested and read from external memory a single time. Indeed, current MBs belonging to dependent frames are accessed multiple times. However, this represents a small impact for both on/off-chip memories, as demonstrated in results section. The GDV (Global Disparity Vector) is taken into consideration to locate the current MBs positions, as shown in Fig. 6.

Impact on External Memory Access: The equations presented below quantify the differences between MBDR and RCDR in terms of external memory accesses and on-chip storage. We consider square SWs with size multiple of 16 pixels (i.e., integer number of MBs). Eq. (1) and (2) represent the on-chip (*OC*) memory size, in number of MBs, for MBDR and RCDR, respectively. Where the search window size is *nxn*; *R* and *D* denote the number of reference or dependent frames, respectively. For MBDR, the quadratic factor related to search window size is further multiplied by the number of reference frames leading to a accentuated on-chip memory increase.

$$OC_{MBDR} = Rn^2 + 1 \quad (1)$$

$$OC_{RCDR} = n^2 + D \quad (2)$$

Eq. (3) and (4) show the external bandwidth (BW_{Step}), in number of MBs, for each *search step* after the on-chip memory is full. The search step is defined as one MB processing in MBDR case and a search window processing in case of RCDR. For each step MBDR reads *R* search window columns while RCDR reads one column plus *D* MBs. Note that MBDR suffers with the search window increase. The bandwidth for a frame line (BW_{Line}) is obtained applying Eq. (5) and (6). BW_{Line} has two components; the line initial read cost (between brackets) and the (*W-1*) steps red cost (curly brackets), where *W* denotes the number of MBs in a frame line. Note, that the *R* factor multiplies n^2 and *Wn* components in case of MBDR leading to strong bandwidth increase with the number of reference frames and search window size. To obtain the total bandwidth for a frame we multiply Eq. (5) and (6) by the number of MB is a frame column (*H*).

$$BW_{MBDR,Step} = Rn + 1 \quad (3)$$

$$BW_{RCDR,Step} = n + D \quad (4)$$

$$BW_{MBDR,Line} = [(Rn^2 + 1)] + \{(W - 1)(Rn + 1)\} \quad (5)$$

$$BW_{RCDR,Line} = [(n^2 + D)] + \{(W - 1)(n + D)\} \quad (6)$$

The third memory bandwidth component, omitted in Eq. (3)-Eq.(7), is the traffic spent to write partial results to the external memory. Although less representative in terms of amount of data, it accesses memory frequently leading to memory contention and efficiency loss. To address these issues, a partial results compressor with buffering is presented.

B. Statistics-Based Partial Results Compressor

The partial results are composed of two distinct data types, (i) motion/disparity vectors and (ii) SAD values, that present distinct numerical range and statistical behavior. For this reason, we discuss them separately. Fig. 7. a shows the histogram of the disparity vectors extracted from *Poznan Carpark* video sequence. Although the disparity vectors are mainly concentrated in DV_x=0, multiple disparity vectors are distributed in a wide value range complicating the compression step.

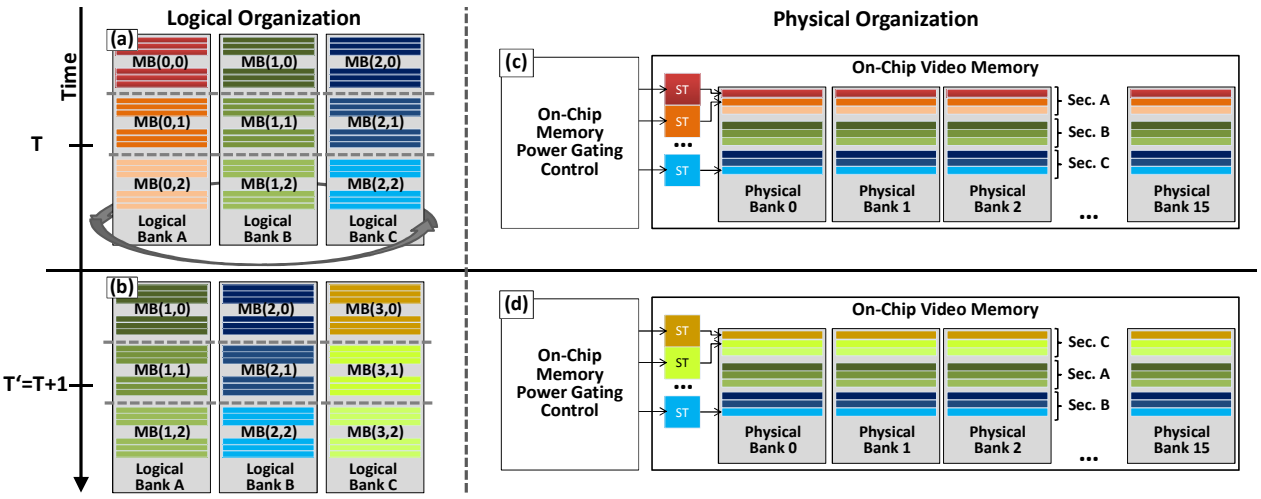

Fig 8. On-Chip Video Memory (a)(b) Logical and (c)(d) Physical Organization

To concentrate these vectors in a reduced range, distinct predictors were evaluated. A median spatial predictor (above and left MBs) provided the best results, as shown by the histogram and PDFs (Probability Density Function) in Fig. 7. b. The differential disparity vectors distribution is concentrated in a small range around DV_x=0 (see Fig. 7. b). For 54 values that represent μ±2σ=95.8% of differential vectors occurrences, a Huffman table was generated according to the techniques presented in [17]. The differential vectors values out of this range are represented by a special Huffman value followed by the vector value in 8-bit binary representation. Analogous statistical analysis and specific Huffman table definition were performed for differential motion vectors.

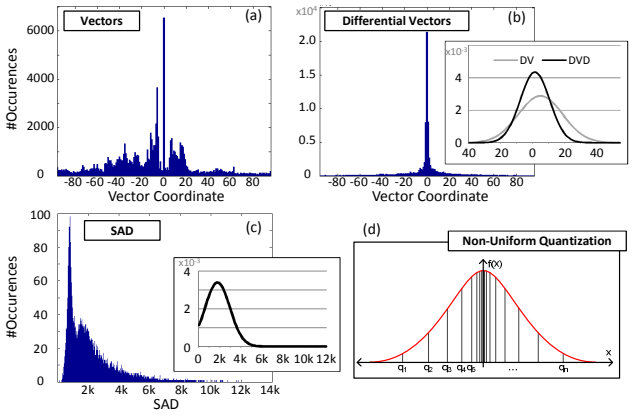

Fig. 7. Partial Results Compressor Statistics

In Fig. 7. c the histogram and PDF of SAD values for DE are presented. As can be noticed, SAD values are spread along a wide numerical range. Still, high concentration is observed around 1k. For such distribution, quantization is required. To reduce the impact of quantization errors, we employ a non-uniform quantization designed for a Gaussian distribution according to Lloyd algorithm and a Lloyd-Max refinement [18]. The quantizer employs 512 levels optimized for minimum mean square error (MMSE). The non-uniform quantization employs a reduced quantization step for regions with high occurrence (close to the average) and larger quantization step in regions with less occurrences (PDF tails), as represented in Fig. 7. d. After quantization the quantized SADs are encoded using a 189-entries Huffman table. SAD values out of range are encoded by a special Huffman value followed by the SAD value in 14 bits. After that, the partial results are concatenated in a 512-bits local buffer. Once this buffer is full, it is written to the external memory following the schedule defined in Fig. 5. The partial results AGU generates serial addresses in a memory region specific for partial results.

RCDR and partial results compression contribute mainly to external memory-related energy reduction. Below we present strategies to reduce on-chip memory energy consumption.

C. On-Chip Video Memory Organization

Our on-chip video memory is logically defined as a circular buffer organized in a 2D-array fashion to provide direct matching to video data. It is composed by *B* (where *B=n*) logical memory banks that rotate after each *search step* to avoid retransmission of overlapping SW. Fig. 8a shows a simplified example with 3x3-MBs search window (each MB is represented by a distinct color) in time instant *T*. The rotation for time instant *T'=T+1* (after a *search step*) is presented in Fig. 8b where the leftmost column of MBs (MB(0,x)) is dropped and a new column at the right is fetched (MB(3,x)). Columns MB(1,x) and MB(2,x) are reused. This organization, however, is not suitable for physical implementation once ME/DE requires MB parallel read.

The physical organization of our on-chip video memory is presented in Fig. 8c and Fig. 8d for time instants *T* and *T'=T+1*, respectively. It is composed of 16 parallel 128-bits wide SRAM banks to store 16 reference pixels per bank line. Each memory line stores and feed one complete MB in parallel. Each bank is further divided in sectors of *n* lines representing one search window column. The total number of lines is defined by the number of MBs in the search window (n^2). Note that differently from the logical organization, MBs columns are not shifted for every *search step*. For that the memory sectors are renamed accordingly, as depicted in Fig. 8. Line-level power gating is employed to support fine-grained power management; the power gating management is discussed in next section.

D. On-Chip Video Memory Power Gating

We propose a statistical power gating scheme that employs multiple SRAM sleep modes in order to reduce the static energy consumption due to the leakage current. Differently from related work solutions [13], this scheme does not require image properties extraction or MB-level memory access prediction in

order to provide a light-weight (but still efficient) solution. Four power states are implemented [19]: S0=OFF (Vdd=0), S1= Data Retentive (Vdd=Vdd*0.3), S2= Data Retentive (Vdd=Vdd*0.5) and S3= ON (Vdd= Vdd). Where each state has an associated wakeup energy cost ($WE_{S0} > WE_{S1} > WE_{S2} > WE_{S3}=0$). For this reason, regions that are frequently accessed are mapped to *S3*, unused regions to *S0*, and other regions are mapped to *S1-S2* according to run-time statistics.

1. **onChipPowerGating**(***n, v, CurrFrame***, offStatMap)
2. ***D*** *{ME, DE}* ← *getNumberDependentFrames*(***v, CurrFrame***);
3. $PowerMap_{SW}$ ← So; // PowerMap initialization
4. **For** *all* ***MB*** ∈ ***nxn*** // for all MBs in the *nxn* SW
5. **If** (*FirstFrame*) // if first frame
6. **Then** // use offline statistics
7. $StatMap_{SW}$ = ***D*** *{ME}**offStatMap$_{ME}$ + ***D*** *{DE}**offStatMap$_{DE}$;
8. **Else** // use statistics from previous encoded frames
9. onStatMap ← *getPrevFramesStat*(***v, CurrFrame***);
10. $StatMap_{SW}$ = ***D*** *{ME}**offStatMap$_{ME}$ + ***D*** *{DE}**offStatMap$_{DE}$;
11. **EndIf**
12. $PowerMap_{SW} = \begin{cases} S1 & If \quad \mu - 2\sigma < StatMap_{SW}(x,y) < \mu - 3\sigma \\ S2 & If \quad \mu - \sigma < StatMap_{SW}(x,y) < \mu - 2\sigma \\ S3 & If \quad Else \end{cases}$
13. **End For**
14. $PowerMap_{SW}$ ← *physicalMemPos*($PowerMap_{SW}$);
15. **For** *all* ***MB*** ∈ ***CurrFrame*** // for all MBs in the *frame*
16. PowerGate($PowerMap_{SW}$);
17. currStatMap ← perform*Search*();
18. **End For**
19. StoreCurrMap(***v, CurrFrame***, currStatMap); return;

Fig. 9. Pseudo-Code of the On-Chip Video Memory Power Gating

Fig. 9. presents our power-gating algorithm which is activated when a new frame processing starts. Firstly, the number of dependent frames is calculated (line 2) and the $PowerMap_{SW}$ is reset. $PowerMap_{SW}$ has one entry for each MB in the search window and if the used search window is smaller than the physical memory, all MBs exceeding the search window are fixed in *S0*. For each MB in the *nxn* search window (line 4), a weighted statistics map ($StatMap_{SW}$) is defined; offline statistics are used in case this is the first processed frame (line 7), otherwise statistics from the previously encoded frames are used (line 9-10). The weighting factors depend on the number of ME/DE ($D\{ME,DE\}$) dependent frames. It is required due to distinct memory access behavior between ME and DE. The $StatMap_{SW}$ is then converted to $PowerMap_{SW}$ by using statistically defined thresholds (line 12). The thresholds are calculated based on the memory access statistics (average and standard deviation) of each block within the search window. The $PowerMap_{SW}$ is finally mapped to the actual physical memory positions (line 14). For each MB in the frame the power gating signals are sent to the on-chip memory (line 16) and the ME/DE search is performed (line 17). At the end statistics are updated for further frames processing (line 19).

E. Energy-Aware Candidate Blocks Merging

Although static energy is becoming dominant in submicron on-chip memories, dynamic energy reduction significantly contributes to overall energy [20]. To avoid frequent on-chip memory line switching, we define an energy-aware candidate blocks merging strategy. As far as multiple dependent MBs are searching simultaneously in the same search window, multiple search points are requested multiple times. The abstract example using TZ search depicted in Fig. 10. shows the access pattern for MB A (left) and MB B (right). Note that for the first search step (dark gray blocks) all candidates are the same. Additionally, some candidates in the second step (black blocks) are repeated. In the figure center, bright blocks represent candidate blocks accessed by both MBs and dark blocks MBs accessed by a single current MB.

Our candidate blocks merger receives all search points generated by the search control and rearranges them in order to process together repeated candidates and avoid unnecessary SRAM line switching (address line switching, bitline pre-charge, sense amplifiers switching, output buffer switching). Moreover, the new processing order follows the left-right and up-down fashion so the processing can start even before the rightmost column is updated for each search step (Section C).

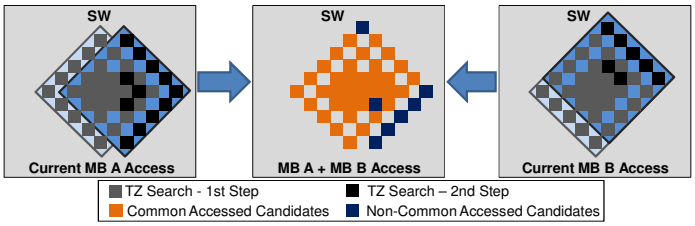

Fig. 10. **Example**: Candidate Blocks Merging

IV. Experimental Results

The experimental results were generated using real video sequences and coding settings recommended by JVT [4] running on the MVC reference software [5]. A customized energy simulator was used to measure the energy consumption of our approach and related solutions. The SRAM leakage and wake-up energies values were calculated based on [19]. The off-chip memory energy savings were evaluated by using the MT46H64M16LF LPDDR 1 Gigabit memory [21]. Note, the experimental results include the wakeup energy overhead.

Four video sequences including three video resolutions were used: *Ballroom* and *Flamenco2* (VGA-640x480), *Balloons* (XGA-1024x768), and *Poznan Carpark* (HD1080-1920x1920). The experiments included 4-views and 8-views sequences considering "IPP" and "IBP" view coding orders. Other settings are: CABAC, FREXt, QP={22,27,32,37}, GOP=8, TZ Search [2]. Note that among the algorithms proposed only SAD quantization may insert coding losses. However, due to the non-uniform quantization, no losses were observed in our experiments. Thus, coding results are omitted.

A. External Memory Energy Savings

Fig. 11. presents the off-chip energy savings for changing search window size and video resolution, under four distinct scenarios, compared to Level-C [7]. Observe that the energy savings scale well with the increase in number of views and search window. Additionally, our solution does not suffer with frame resolution increase. Higher savings happen in case of "IBP" due to more intense search window reuse, i.e., each search window is used by an increased number of current MBs. These results include energy reduction due to the partial results compression. Our compressor leads to 53.2% (average) external energy reduction for partial results communication (compared to the non-compression scenario), as detailed in Fig. 12. a.

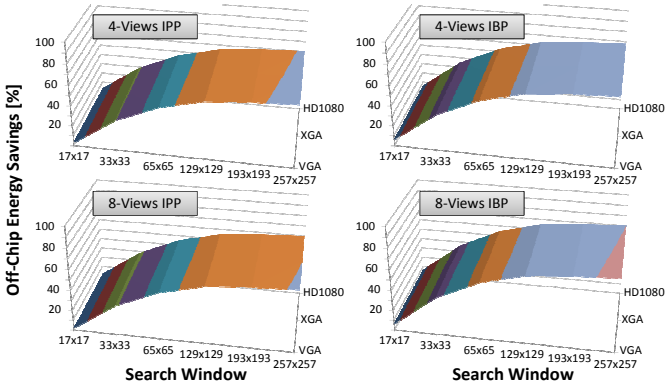

Fig. 11. Off-Chip Memory Energy Savings

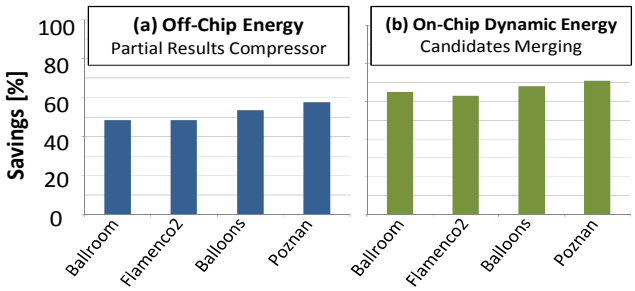

Fig. 12. Energy Savings due to (a) Partial Results Compressor and (b) Candidates Merging.

B. On-Chip Video Memory Energy Savings

Compared to the Level-C [7], our on-chip video memory size is significantly reduced (see Fig. 13.) because there is no need to simultaneously store multiple search windows on chip. Note that our on-chip memory grows smoothly with the search window increase. Moreover, compared to the search window storage, the cost (considered in Fig. 13.) for storing current MBs (that may reach 9 MBs for 8-views "IBP") is negligible. This cost is amortized as the search window increases. The reduced on-chip memory size directly leads to less static energy consumption, as shown in Fig. 14. Compared to [7], 77% energy reduction is reached without employing our power-gating technique. If the power-gating is used, further 88% of reduction is reached outperforming [13] in 61 %. These results refer to 4-views "IBP" scenario.

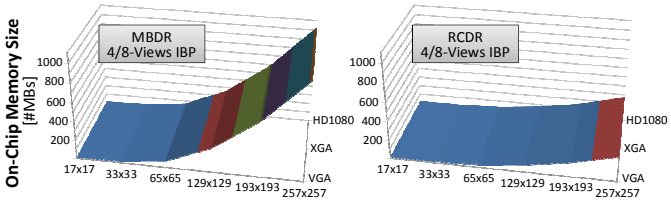

Fig. 13. On-Chip Memory Size Reduction

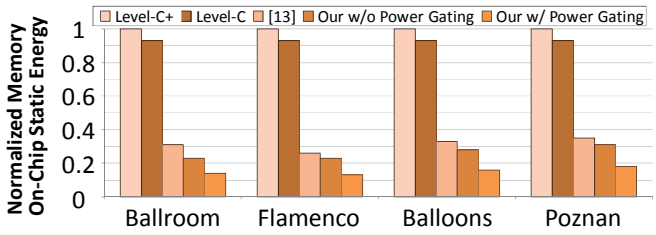

Fig. 14. On-Chip Static Energy Savings (Leakage)

In terms of on-chip dynamic energy, our candidate merging strategy reduces the energy consumption in 65%, as shown in Fig. 12. b. At the best of authors' knowledge, this is the first application specific technique to address the dynamic on-chip memory energy consumption for the ME/DE.

V. Conclusion

A memory hierarchy for Motion and Disparity Estimation on Multiview Video Coding was presented. It exploits a reference-centered data reuse scheme along with partial results compression and memory access scheduling in order to reduce external memory energy. An on-chip video memory organization with line-level power gating and candidates merging scheme is presented targeting on-chip energy reduction. Our memory architecture provides up to 71% off-chip memory energy reduction. On-chip memory-related energy is reduced on 88% and 65% for static and dynamic energies, respectively.

Acknowledgement

This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre "Invasive Computing" (SFB/TR 89); http://invasic.de.

References

[1] A. Smolic, et al. "Coding Algorithms for 3DTV - A Survey." In: IEEE TCSVT, v. 17, n. 11, pp. 1606-1621, nov. 2007.
[2] Joint Draft 8.0 on Multiview Video Coding, JVT-AB204, 2008.
[3] P. Merkle, et al. "Efficient Prediction Structures for Multiview Video Coding." In: IEEE TCSVT, v. 17, n. 11, pp. 1461-1473, nov. 2007.
[4] JVT. "Com. Test Cond. for Multiview Video Coding". JVT-T207, 2007.
[5] JMVC Reference Software , Sep. 2009.
[6] J.-C. Tuan, et al. "On the Data Reuse and Memory Bandwidth Analysis for Full-Search Block-Matching VLSI Architecture." In: IEEE TCSVT, v. 12, n. 1, p. 61-72, jan. 2002.
[7] C.-Y. Chen, et al. "Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders." In: TCSVT, v. 16, n. 4, p. 553-558, april. 2006.
[8] P.-K. Tsung, et al. "System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint". IEEE ISCAS p. 1001-1004, 2007.
[9] T.-C. Chen, et al, "Single Reference Frame Multiple Current Macroblocks Scheme for Multi-Frame Motion Estimation in H.264/AVC", In IEEE ISCAS, 2005, pp. 1790 – 1793.
[10] T.-Y. Kuo, et al. "A novel method for global disparity vector estimation in multiview video coding". In: IEEE ISCAS 2009.
[11] X. Xu, Y. He , "Fast disparity motion estimation in MVC based on range prediction," IEE ICIP, pp.2000-2003, 2008.
[12] S. Saponara, L. Fanucci, "Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems", IEE Comp. & Digital Tech., vol.151, no.1, pp. 51- 59, 2004.
[13] B. Zatt, M. Shafique, F. Sampaio, L. Agostini, S. Bampi, J. Henkel, "Run-time adaptive energy-aware motion and disparity estimation in multiview video coding", IEEE DAC, pp. 1026-1031, 2011.
[14] M. Shafique, B. Zatt, F. L. Walter, S. Bampi, J. Henkel, "Adaptive Power Management of On-Chip Video Mamory for Multiview Video Coding", IEEE DAC, pp. 866-875, 2012.
[15] B. Zatt, M. Shafique, S. Bampi, J. Henkel, "A Low-Power Memory Architecture with Application-Aware Power Management for Motion & Disparity Estimation in Multiview Video Coding", IEEE ICCAD, pp. 40-47, 2011.
[16] J. Yang et al., "Multiview video coding based on rectified epipolar lines", International CICSP, pp. 1-5, 2009.
[17] Huffman, D.A., "A Method for the Construction of Minimum-Redundancy Codes," IRE, vol.40, no.9, pp.1098-1101, Sept. 1952.
[18] Max, J.; , "Quantizing for minimum distortion," Information Theory, IRE Transactions on , vol.6, no.1, pp.7-12, March 1960.
[19] H. Singh et al., "Enhanced leakage reduction techniques using intermediate strength power gating", IEEE Transaction on Very Large Scale Integration, vol. 15, no. 11, pp. 1215-1224, 2007.
[20] S.l Rodriguez, B. Jacob,"Energy/power breakdown of pipelined nanometer caches (90nm/65nm/45nm/32nm", ISLPED, pp. 25-30, 2006.
[21] Micron. "1Gb: x16, x32 Mobile LPDDR SDRAM". Available at: www.micron.com. Last Accessed: December, 2012.