Two-Level Scalable Motion Estimation Architecture with Fractional-Pixel Accuracy and Efficient Data Re-Usage

Tiago Dias
tdias@sips.inesc-id.pt

Nuno Roma
nuno.roma@inesc-id.pt

Leonel Sousa
las@inesc-id.pt

Instituto Superior Técnico / INESC-ID
Rua Alves Redol, 9 - 1000-029 Lisboa - PORTUGAL

Abstract

A scalable and highly configurable architecture for motion estimation with fractional-pixel accuracy is proposed in this paper. This circuit is highly modular and efficient, providing a flexible integration and implementation of different sub-pixel motion-estimation VLSI circuit. The adopted pipeline processing structure provides the overall architecture with the capability to compute motion vectors with any accuracy without significant penalization on its throughput. Redundant accesses to the main memory are also completely avoided by re-using and transferring the required pixels between the several modules of the architecture. Experimental results show that the proposed multi-level architecture is suitable to implement real-time motion estimators with fractional pixel accuracy on configurable devices, such as FPGAs. These results were obtained for high quality video encoders and 4CIF image format, using half-pixel accuracy.

1. Introduction

Motion-compensation is a fundamental technique to exploit interframe redundancy in video coding [1], in order to achieve higher compression rates and to reduce the required bit rate. In most systems, the set of displacement vectors used for motion compensation is estimated by comparing each block of the current frame with the candidate blocks of the corresponding search area, defined within the previously processed frame and using a given matching criteria. Due to its simplicity and its satisfactory results, the Sum of Absolute Differences (SAD) similarity measure has been extensively preferred for video coding.

The displacement of objects between two consecutive frames in a video sequence is rarely an integer number of pixels. In fact, in order to efficiently exploit the temporal redundancies between successive frames, most modern video coding standards [7, 8, 5, 10] have been incorporating optional motion compensation modes with fractional-pixel accuracy (FPA), by making use of motion vectors that can take values in a given sub-pixel resolution.

However, the introduction of FPA on standard integer-pixel accuracy (IPA) VLSI architectures usually implies the re-design of all processing structure. Most architectures that have been proposed either compute the matching function at the whole sub-pixel level or compute the best integer accuracy vector and then use this coarse estimate to refine the search, by retrieving the required data from the frame memory.

In this paper, a new and modular architecture for FPA motion estimation is proposed. This architecture is based on an efficient class of processing structures for integer-accuracy Full-Search Block-Matching (FSBM) motion estimation (ME), that provides minimum latency, maximum throughput and a full utilization of the hardware resources. The required configurability and scalability properties are attained by integrating the FPA motion-estimation module as an embedded co-processor on the base IPA structure. Such module makes use of an integer coarser estimate, obtained from the first stage, to compute the fractional accuracy motion vector (MV). Furthermore, to minimize the required memory bandwidth, the performance of the base FSBM architecture was further enhanced by incorporating a new and dedicated data re-use block of the search area pixels.

The paper is organized as follows. In Section 2 it is presented the IPA Full-Search Block-Matching architecture that computes a coarse estimate of the motion vector. It is also presented a new and dedicated data re-use block of the search area pixels that was developed and incorporated into this base structure in order to minimize the required memory bandwidth. In Section 3 the developed FPA motion estimation module is presented, together with its internal functional units. Section 4 presents the obtained implementation results and Section 5 concludes the presentation.

2. Base Architecture

2.1. Integer Accuracy Motion Estimator

Several FSBM structures for ME have been proposed over the last few years [11, 17]. Recently, a new class of parameterizable hardware architectures that provides minimum latency, maximum throughput and a full and efficient utilization of the hardware resources has been proposed [14, 15]. In this architecture, just like in any other type-I bidimensional structure [9], each pixel of the reference macroblock is assigned to one processor element (PE)

of the N^2 *active PEs* that compute the SAD similarity measure. However, unlike other structures, the main feature of the architecture proposed in [14, 15] is the usage of only one *passive block* with $(2p-1)\times N$ *passive PEs* to displace the search area pixels. To do so, the passive PEs located on the right margin of the passive block are connected to the active PEs of the left margin of the active block, giving rise to an innovative cylindrical structure, as shown in Figure 2(a). Search area pixels are displaced in such cylindrical structure by adopting the same zig-zag processing scheme proposed by Vos [17]. With this approach, not only is this architecture able to efficiently displace the search area pixels but it also overcomes the major drawback presented by other architectures (see [17]) that require $N\times(2p-1)$ redundant passive PEs. Moreover, with this new processing structure [14, 15], it is also possible to keep all PEs busy at every time instant, thus computing in parallel similarity measures for different candidate blocks [2].

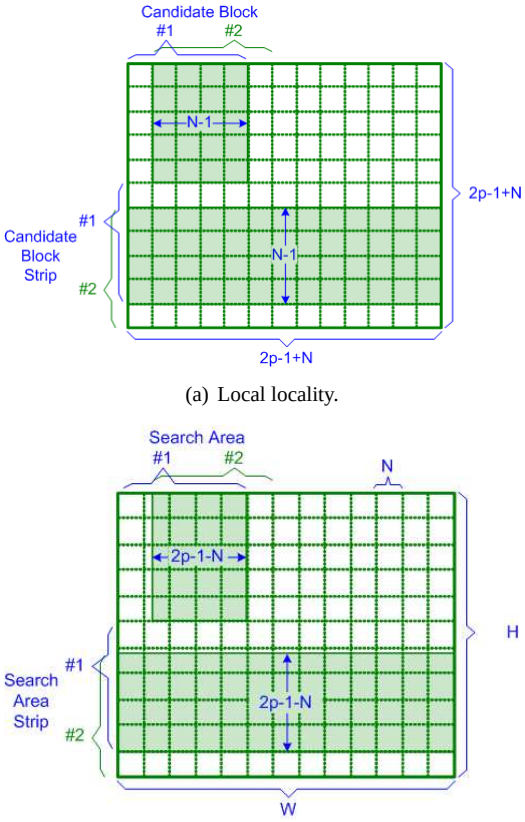
2.2. Data Re-usage

Although this peculiar and efficient cylindrical structure, together with the zig-zag processing scheme, minimizes the latency and maximizes the data throughput, this optimum performance can only be attained if the processing of the video data is never halted. Such condition is only achieved if reference area (RA) and search area (SA) pixels are continuously feedthrough into the processing structure. Consequently, a significant memory bandwidth is often needed to process high-quality video. To circumvent this strict requisite, ME architectures have to exploit the locality and the redundancy properties of video data accesses in order to alleviate such restrictions.

To evaluate the efficiency of memory accesses in the FSBM ME algorithm, i.e., the degree of redundancy in memory accesses, Tuan et. al. [16] defined a *Redundancy Access factor* (Ra) representing the average access count per pixel in the FSBM processing, as shown in Equation (1). As a result, the memory bandwidth requisites of ME video encoding systems can be expressed as a function of Ra, therefore depending on the degree of data re-use of the system, as shown in Equation (2), where f represents the frame rate and Ra_{CP} and Ra_{SP} the Ra values for the current and search pictures, respectively. Four different degrees of data re-use can be specified depending on the locality of data (see figure 1):

- *Level A* – exploits locality of data within the candidate block strip;
- *Level B* – exploits locality of data among adjacent candidate block strips;
- *Level C* – exploits locality of data within the search area strip;
- *Level D* – exploits locality of data among adjacent search area strips.

$$Ra = \frac{total\ number\ of\ memory\ accesses\ in\ task}{pixel\ count\ in\ task} \quad (1)$$



(a) Local locality.

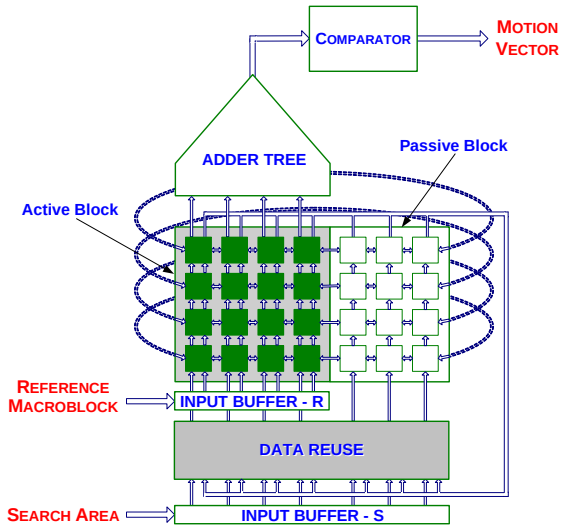
(b) Global locality.

Figure 1. Locality of data in the search picture.

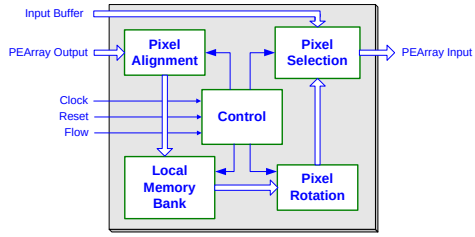
$$BW = f\times W\times H\times Ra_{CP} + f\times W\times H\times Ra_{SP} \quad (2)$$

In the architecture proposed in [14, 15], the $N\times N$ pixels of the current block are stored in local memory, namely, in the reference area registers inside the active PEs, as depicted in Figure 2(a). This corresponds to the maximum possible saving in terms of memory accesses for the current picture. A less efficient data re-use approach was adopted for the SA, since the zig-zag processing scheme only allows for the re-use of SA data within a single search area. Hence, this class of architectures only exploits the locality properties of RA data (data re-use level B [16]) which is not enough to comprise with the memory bandwidth requirements for high-quality video ME. Thus, it seems clear that the data re-use level must be increased to levels C or D if we want to decrease such bandwidth requisites. However, although data re-use level D provides the best solution in terms of memory bandwidth, its hardware requirements significantly increase with the size of the image. Furthermore, a change from data re-use level C to D provides the smallest decrease in the memory bandwidth among all other transitions between adjacent levels [16]. Consequently, the ME architecture proposed in this paper incorporates a new and dedicated data re-use block in the search area processing circuit in order to adopt a data re-use level C, since it provides the best trade-off in terms of memory bandwidth and local memory size [3].

The developed data re-use circuit is composed by four distinct blocks (see Figure 2(b)):



(a) Base architecture.



(b) Data re-use circuit.

Figure 2. Simplified block diagram of the proposed ME processor.

- *Pixel Alignment* block, responsible for the first misalignment correction carried out on the search area pixels leaving the top of the array, due to the odd number of shifts $(2p-1)$ performed by the zig-zag processing scheme;

- *Local Memory Bank*, composed by $(N+2p-1)\times(2p-1)$ registers organized in a $(2p-1)$ lines FIFO structure in order to store the pixels that are common to the two adjacent search areas;

- *Pixel Rotation* block, which performs the required displacement of the SA pixels in each line, between the position that they occupied in the previously processed search area and their new position in the current search area;

- *Pixel Selection* block, responsible for selecting and compositing each new line of search area pixels that is fed into the array with data stored in the *Local Memory Bank* (re-used pixels) and/or data retrieved from the main frame memory (new pixels).

The data flow between the several blocks of this circuit is controlled by a dedicated unit that guarantees the synchronization and the re-alignment between the pixels leaving the array and those re-entering in the processing structure.

3. Multi-Level ME Architecture with Fractional-Pixel Accuracy

The architecture proposed in [14, 15] provides the best matching between the current block and all candidate blocks of the SA using integer-pixel accuracy. However, as it was previously referred, it is widely accepted that the displacement of objects between two consecutives frames is rarely an integer number of pixels. Therefore, in order to efficiently exploit the temporal redundancies between successive frames, most modern video coding standards [5, 10, 7, 8] adopt ME with fractional-pixel accuracy, in which MVs can point to prediction blocks placed at half-pixel locations of the SA [5, 7], or even at quarter-pixel locations for the most recent standards [10, 8].

To overcome this limitation, the efficient architecture recently developed by the authors of this work [14, 15, 13] had to be provided with all the required means to comply with such technology advances. However, such enhancements were performed by taking into account that not only is the fractional-pixel accuracy estimation an optional and extended part of most video coding standards, but also the part that none of the standards specify the algorithm that should be used to perform the accuracy extension. Consequently, a modular and highly configurable multi-level structure, capable of incorporating a wide range of different fractional-pixel ME circuits to compute MVs with different fractional-pixel accuracies, was adopted. By using such a flexible structure, the fractional-pixel ME module can be regarded as a co-processor of the main integer-pixel accuracy ME circuit. When embedded in the system, it may provide the overall architecture with the capability to compute MVs with any pixel accuracy, depending on the specific algorithm implemented in the interpolation and fractional-pixel circuits (see figure 3).

The proposed architecture estimates the fractional-pixel MVs using a two-step search procedure, thus providing significant savings in terms of hardware and power consumption due to the subsequent reduction in the number of computations. In the first step, executed in the first level of the architecture, the best matching block at integer locations of the SA is found. Then, in the second level, the SA surrounding the best integer candidate block is interpolated into a higher resolution and the former MV is refined into a fractional-pixel accuracy. The modularity and configurability properties of the architecture are guaranteed by simple and efficient protocols that are used to transfer data

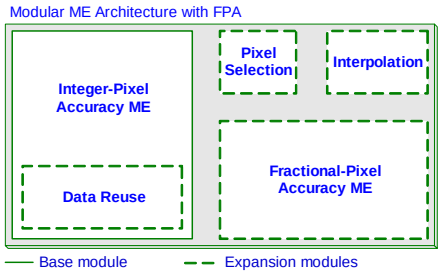


Figure 3. Block diagram of the proposed multi-level architecture.

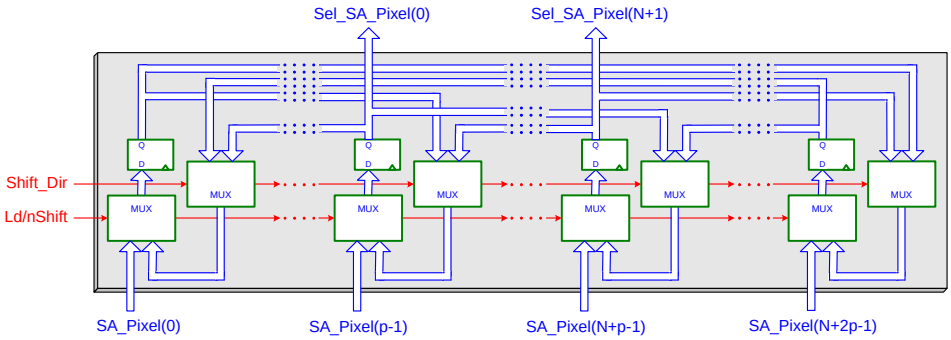


Figure 4. Proposed pixel selection module.

between the several blocks of the architecture, which provides an easy insertion/removal and development of new and different interpolation modules.

Another important feature of this multi-level architecture is that it does not require any additional accesses to the main frame memory: RA and SA pixels flow through one level to another and from one module to another within the structure. Hence, the memory bandwidth requirements of the proposed multi-level architecture are minimized. Nevertheless, this data flow requires that all levels of the architecture are kept synchronized, i.e., use the same clock signal. Consequently, a special attention has to be paid in the design of all new modules of the architecture so that the performance of the original integer-pixel ME base architecture is not compromised. Figure 3 depicts the block diagram of the proposed multi-level fractional-pixel ME architecture. This architecture is composed by an integer-pixel ME unit, a pixel selection unit, an interpolation unit and a fractional-pixel ME unit.

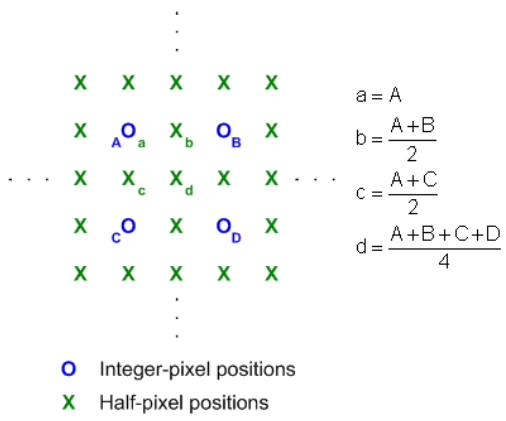
3.1. SA integer-pixels selection

To avoid redundant accesses to the main frame memory, the SA pixels required by the interpolation process are retrieved from the integer-pixel accuracy ME circuit. Although this circuit holds the entire $(N+2p-1)^2$ pixels of the integer accuracy SA, only the set of $(N+1)^2$ pixels surrounding the best candidate block are needed in the next refinement step. This subset of pixels is retrieved from the integer-pixel ME circuit by a dedicated pixel selection unit that was designed to select the block of $(N+1)^2$ pixels as a function of the coordinates of the IPA MV. The operation of this selection circuit, whose architecture is depicted in figure 4, is quite straightforward: the horizontal coordinate of the MV is used to rotate the line of pixels outputted by the integer-pixel ME circuit so that the required $N+1$ pixels become aligned at the centre of the line, while the vertical coordinate of the MV selects the lines to be sent to the interpolation unit. The line selection is done by asserting the data valid output signal of this unit.

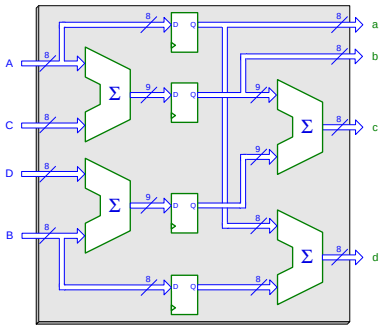
3.2. Interpolation unit

The interpolation circuit computes higher resolution blocks of the original SA surrounding the best candidate block that was found by the integer motion estimation circuit, according to the adopted interpolation algorithm. Considering that numerous algorithms have been proposed in the last few years [**?**], one of the premises that guidelined the design of the proposed multi-level architecture was to provide it with the capability to implement any generic interpolation algorithm. To accomplish such objective, not only was the architecture structured in a flexible and modular way, but it makes also use of two interface protocols to transfer the data from the IPA motion estimator to the FPA module. Such protocols are used to acknowledge a valid SA line of pixels at the output of the IPA motion estimator and to synchronize the two pipelined systolic circuits.

In figure 5(b) it is presented one possible implementation of the interpolation unit using the bilinear interpolation algorithm, the most used in video coding hardware ar-



(a) Algorithm implementation.



(b) Proposed circuit.

Figure 5. Half-pixel prediction using bilinear interpolation.

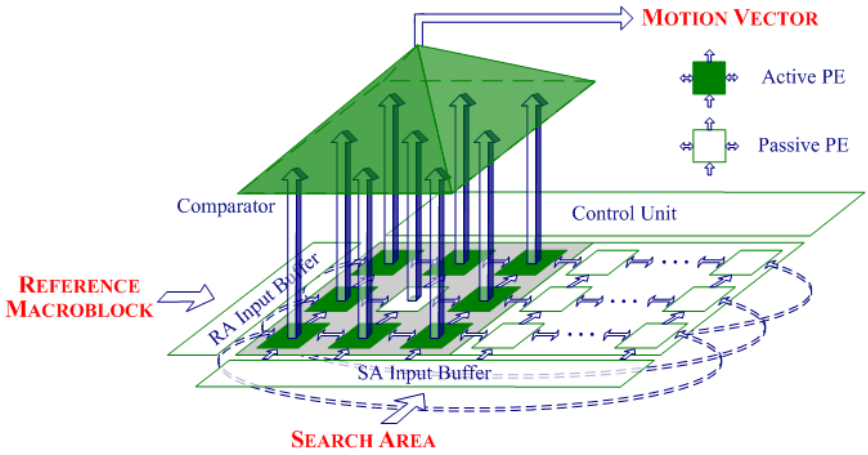


Figure 6. Proposed fractional-pixel ME circuit.

chitectures. From figure 5(a), which depicts the computation of the bilinear interpolation algorithm, it can be seen that pixels with half-pixel accuracy can be computed in two stages: in the first stage pixels *b* and *d* are calculated, while in the second stage the values of pixels *a* and *c* are found by using the two previously computed pixel values. Since the processor with integer-pixel accuracy outputs a line of SA pixels every $(p-1)$ clock cycles, the N pixels of the interpolated SA lines can therefore be computed serially in a pipeline structure, which minimizes the hardware requirements.

3.3. Fractional-pixel motion estimator

The proposed architecture for the ME circuit with FPA, whose block diagram is presented in figure 6, is an improved version of the type-2 architecture proposed by Vos [17]. In this structure, one active PE is used for each displacement vector. In each clock cycle all PEs process the same pixel of the RA, but for different displacements. Hence, after N^2 clock cycles, all pixels of the RA have been processed and a new set of $\left\{\left[\left(\frac{1}{k}-1\right)\times 2+1\right]^2-1\right\}$ similarity measures with FPA are simultaneously available, where k is the FPA factor. This set of values is fed into a binary tree comparator, outputting a fractional-pixel accurate MV after $\left\lceil \log_2\left(\left[\left(\frac{1}{k}-1\right)\times 2+1\right]^2-1\right)\right\rceil$ clock cycles. Hence, provided that the RA and the SA pixels are available at the data inputs of the circuit, the global latency of the FPA motion estimator is $N^2+\left\lceil \log_2\left(\left[\left(\frac{1}{k}-1\right)\times 2+1\right]^2-1\right)\right\rceil$.

As it can be seen from figure 6, the proposed architecture is composed by two sets of processing elements: *i)* those that compute the SAD similarity measure, designated by *active PEs*; and *ii)* those used to displace the interpolated SA pixels, designated by *passive PEs*. Similarly to the integer-pixel ME circuit, these PEs are interconnected using the same highly efficient cylindrical structure. The adopted zig-zag processing scheme is also the same used in the motion estimator with IPA, although in this FPA estimator it is only used to displace the SA pixels within the array of PEs. By doing so, it is guaranteed that the FPA ME circuit inherits all the eficiency characteristics of the integer ME circuit [15, 17] and that all PEs are kept busy at any time instant.

The active PEs are composed by four main blocks, as shown in figure 7: the SA displacement circuit (A), the RA pixels load circuit (B), the absolute difference arithmetic unit (C) and the accumulation unit (D). Since the passive PEs are only responsible for the displacement of SA pixels, their internal structure is the same as the SA transfer circuit of the active PE (block A).

The SA pixel used in the computation of the distance measure is selected from the set of three pixels supplied by the PEs located below, on the left and on the right of the considered active PE and is transferred to an internal

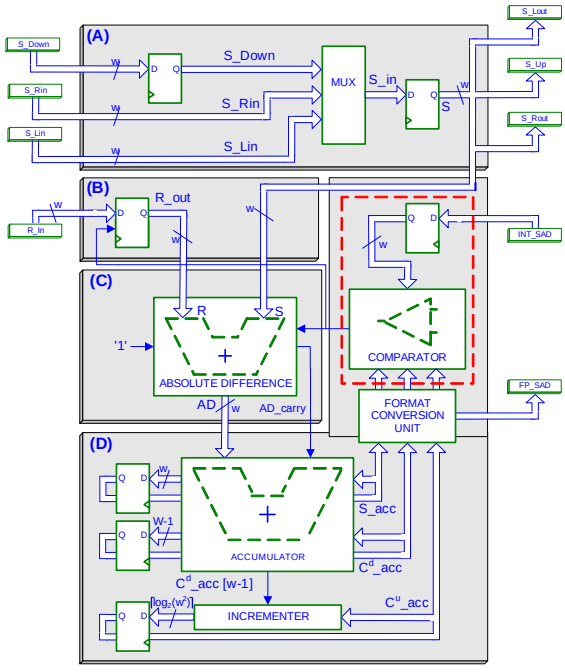


Figure 7. Active processor element circuits: (A) Search area displacement; (B) Reference pixel storage; (C) Absolute difference arithmetic unit; (D) Accumulator unit.

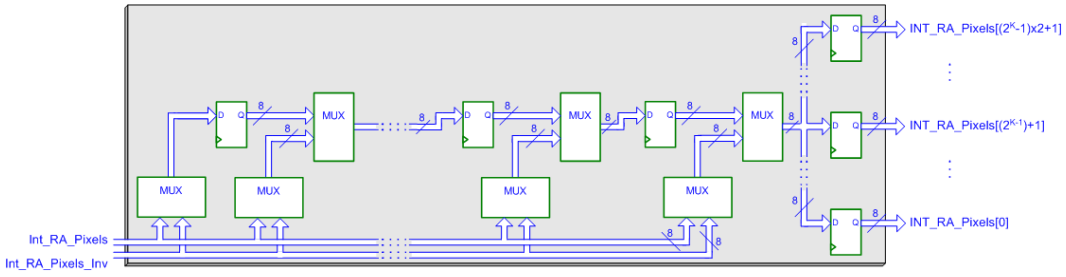


Figure 8. RA input buffer of the fractional pixel motion estimator.

standing-data register. In contrast, the RA pixel is transferred from the running-data registers that feed the lines of active PEs to the standing-data register inside the active PEs. The SAD distance measure is then computed in the absolute difference arithmetic unit and the obtained result is added with the partial sum computed in the previous clock cycles in the accumulation unit.

To reduce the power consumption of the circuit, the accumulation unit has also an extra expansion module that can be embedded in the processor. The circuit consists of a standing-data register, to hold the distance measure of the best integer-pixel candidate block, and a comparison circuit that disables the computation of the fractional-pixel SAD similarity measure whenever the partial accumulated value exceeds the integer resolution similarity measure. In such situation, the previously computed MV with integer-pixel accuracy is outputted by the circuit.

The RA and SA pixels are loaded into the processor array by means of one parallel-in-serial-out (PISO) and one parallel-in-parallel-out (PIPO) input buffers, respectively. The purpose of the PIPO buffer is to hold the line of SA pixels generated in the interpolation block until these pixels can be transferred, in parallel, to the PEs array, i.e., when the processing of the current line of pixels is completed. In contrast, the PISO buffer stores the RA pixels outputted by the IPA motion estimator and transfers them serially to each of the $\left\{\left[\left(\frac{1}{k}-1\right)\times 2+1\right]^2-1\right\}$ PEs that compose the processing array. Although this buffer could be implemented by a set of N cascaded registers, it was implemented using $\left[N-1+\left(\frac{1}{k}-1\right)\times 2+1\right]$ cascaded registers, in which the inputs of the last $\left[\left(\frac{1}{k}-1\right)\times 2+1\right]$ registers are all connected to the output of register $(N-1)$, as shown in figure 8. By doing so, each line of active PEs in the array is fed with the output of one of this registers, thus minimizing the fan-out of the overall circuit, due to the broadcast of the RA pixel value to all active PEs in the array.

4. Experimental Results

The proposed multi-level architecture for FSBM ME with FPA was completely described using both behavioural and fully structural parameterisable VHDL. Several setups of these descriptions were synthesised in a general purposed VIRTEX XCV3200E-7 FPGA, using the Xilinx Synthesis Tool from ISE 6.1.3i. The particular implementation setup presented in this paper considers an IPA module using a maximum displacement in each direction of the search area (p) equal to the macroblock width (N) and a FPA module using the bilinear interpolation algorithm for half-pixel accuracy. This set of configurations corresponds to the set of parameters more frequently adopted by the ITU-T H.26x [6, 7, 8] and ISO MPEG video coding standards [4, 5, 12], i.e., using macroblocks composed by 8×8 ($N = 8$) and 16×16 pixels ($N = 16$).

Table 1. Percentage of CLB slices, LUTs and maximum operating frequencies for the considered implementations of the proposed multi-level ME architecture in a VIRTEX XCV3200E-7 FPGA.

Architecture Type	CLBs	LUTs	F (MHz)
IPA	15.8%	8.6%	129.4
IPA + DR	21.5%	10.4%	129.4
IPA + DR + FPA	27.9%	14.6%	117.8
IPA + DR + FPA (LP)	28.4%	15.2%	117.8

(a) $N = p = 8$.

Architecture Type	CLBs	LUTs	F (MHz)
IPA	62.8%	34.7%	124.1
IPA + DR	83.9%	38.4%	124.1
IPA + DR + FPA	95.1%	45.6%	124.1
IPA + DR + FPA (LP)	95.8%	46.2%	124.1

(b) $N = p = 16$.

Table 1 presents the experimental results obtained with these implementations. When compared with the base architecture with the data re-use block (IPA+DR), the FPA unit using the SA data re-use block requires few extra hardware resources for the implementation of the multi-level architecture in a FPGA device. This fact is emphasized for the case $N = p = 16$ and can be explained by taking into account the FPGA internal structure (a FPGA device is composed by a set of slices, each one containing two flip-flops and two LUTs) and the hardware requirements of the data re-use and FPA modules. While the data re-use module is implemented mostly with flip-flops and uses few combinatory logic (LUTs), the FPA circuit requires mainly combinatory logic, which makes these two units complementar in what concerns the required FPGA resources. Consequently, the synthesis and routing tools can optimize the allocation and sharing of the used CLB slices, thus minimizing the hardware requirements of the multi-level architecture.

Moreover, albeit the ideal configuration of the multi-level ME architecture would be the one using the IPA,

Table 2. Hardware resources and number of clock cycles required by the multi-level architecture modules.

Structure	Number of 1-bit registers
Integer ME	$N^2\left[5\times w-1+log_2(N)\right]+N\times(2p-1)\times w\times 4$
Data Re-Use	$(2p-1)\times(N+2p-1)\times w$
Interpolation	$2\times(N+1)+4w+2$
Fractional ME	$\left\{\left[\left(2^{1/k}-1\right)\times 2+1\right]^2-1\right\}\times\left[5\times w+log_2\left(N^2\right)\right]$

(a) Number of registers.

Structure	Latency
Integer ME	$(N+2p-1)\times(p-1)+log_2(N)+1$
Data Re-Use	0
Interpolation	$(p-1)+1$
Fractional ME	$N^2+log_2\left\lceil\left[\left(2^{1/k}-1\right)\times 2+1\right]^2\right\rceil$

(b) Latency.

data re-use and FPA modules, such a configuration requires enormous amounts of hardware for large search ranges, as shown in table 3 and figure 9(b). Hence, for implementations based on devices with restricted amounts of hardware, one possible alternative configuration of the multi-level ME architecture would make use of only the IPA and FPA modules, since the data re-use module is the one that requires more hardware (see table 2a, where w denotes the number of bits used in the representation of the pixel value). In fact, from tables 1 and table 3, one can conclude that both the hardware requirements and the performance of such a configuration would be very similar to the one obtained for an implementation using only the IPA module. Nevertheless, its memory bandwidth constraints would make the use of such structure in high-quality video ME almost impractical, as it is shown in figure 9(a).

Figure 9(c) depicts the ratios between the required number of memory accesses and the number of registers required for the stand alone IPA structure and for the IPA+FPA structure. One observes that for SA ranges greater than twice the macroblock width $(p \geq 2N)$, the increase of the hardware requirements does not justify the use of the data re-use module. Moreover, figure 9(c) clearly shows that the optimal configuration for the multi-level ME circuit is the one with $N = p$.

In table 1 it is also presented the operating frequency obtained for each considered implementation. One observes that the maximum operating frequency of the overall multi-level architecture (IPA+DR+FPA) is almost the same as the one obtained for an architecture composed by solely the IPA module, with a slight increase of the latency, as shown in table 2b. This was already expected, due to the pipeline structure adopted to interconnect the several modules of the system. From this results, one can also derive the variation of the frame rate obtained for different image formats and several search area setups (ρ). This variation is depicted in figure 10 from which it is possible to conclude that the proposed multi-level ME processor using the data re-use and the FPA modules is able to estimate MVs up to a rate of 30 fps for high-quality video with the 4CIF image format.

Table 3. Percentage of CLB slices, LUTs and maximum operating frequency to implement only the fractional-pixel ME circuit in a VIRTEX XCV3200E-7 FPGA.

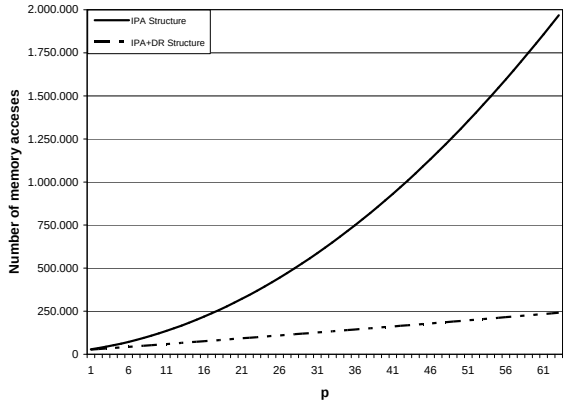
N	CLB Slices	LUTs	F (MHz)
8	3.8%	3.2%	141.4
16	5.2%	4.5%	138.8

5. Conclusions

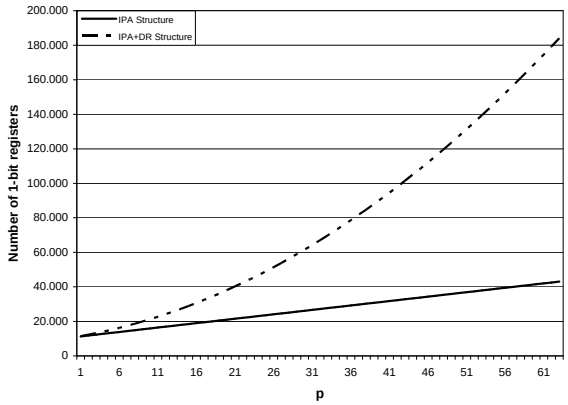
A modular and configurable architecture for FPA ME was presented. This architecture is composed by two cascaded processing blocks, matching an IPA FSBM motion estimator and a FPA block that carries out the estimation with sub-pixel accuracy using the value obtained from the first stage as an initial estimate of the final motion vector. Due to its pipeline processing scheme, this architecture is able to compute MVs with any given accuracy without compromising its throughput and at the cost of a slight increase of its latency. To minimize the required memory bandwidth, a dedicated data reuse block of the search area pixels was embedded in the base architecture. Hence, the performance of the architecture is optimized by avoiding redundant accesses to the frame memory. Experimental results show that the proposed multi-level ME processor using the data re-use and the FPA modules is able to estimate motion vectors with half-pixel accuracy up to a rate of 30 fps for high-quality video with nowadays existing memory chips. Moreover, the obtained results also show that the use of the FPA module in Field-Programmable Gate Array (FPGA) implementations of the proposed architecture causes an almost negligible increase in the global motion estimator hardware requirements.

Acknowledgements

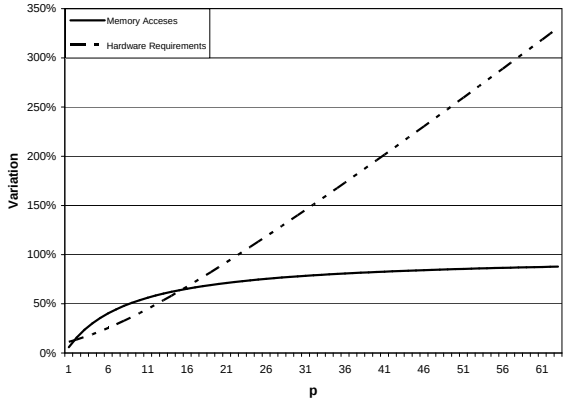
This work has been supported by the POSI program and the *Portuguese Foundation for Science and for Tech-*



(a) Memory accesses.



(b) Required 1-bit registers.



(c) Variation

Figure 9. Memory accesses and required registers for the proposed multi-level ME architecture ($N = 16$).

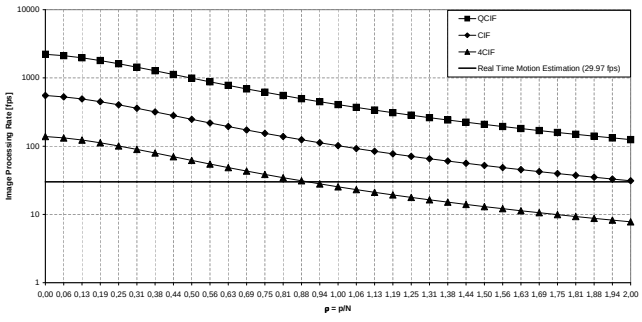


Figure 10. Number of frames processed per second for different image formats ($N = 16$, $p = 16$).

nology (FCT) under the research project *Configurable and Optimized Processing Structures for Motion Estimation* (COSME) POSI/CHS/40877/2001.

References

[1] Vasudev Bhaskaran and Konstantinos Konstantinides. *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer Academic Publishers, 2nd edition, June 1997.

[2] Tiago Dias. Dedicated processors for motion estimation in video sequences. Master's thesis, Instituto Superio Tcnico, Lisbon, September 2003.

[3] Tiago Dias, Nuno Roma, and Leonel Sousa. High-throughput parallel FSBM ME processor with an efficient data reuse scheme. Technical report, INESC-ID – Lisboa, *http://parsig.inesc-id.pt*, Portugal, November 2003.

[4] ISO. *ISO/IEC JTC1 CD 11172 - "Coding of moving pictures and associated audio for digital storage media up to 1.5 Mbit/s"*, 1992.

[5] ISO. *ISO/IEC JTC1 CD 13818 - "Generic coding of moving pictures and associated audio"*, 1994.

[6] ITU-T. *ITU-T Recommendation H.261 - "Video Codec for Audiovisual Services at $p \times 64$ kbits"*, March 1993.

[7] ITU-T. *ITU-T Recommendation H.263 - "Video Coding for Low Bitrate Communication"*, February 1998.

[8] ITU-T. *ITU-T Recommendation H.264, "Advanced Video Coding for Generic Audiovisual Services"*, May 2003.

[9] T. Komarek and P. Pirsch. Array architectures for block matching algorithms. *IEEE Transactions on Circuits and Systems*, 36(10):1301–1308, October 1989.

[10] MPEG. *MPEG-4 Video Verification Model - version 18.0 - ISO/MPEG N3908*, January 2001.

[11] Y. Ooi. Motion estimation system design. In Keshab K. Parhi and Takao Nishitani, editors, *Digital Signal Processing for Multimedia Systems*, chapter 12, pages 299–327. Marcel Dekker, Inc, 1999.

[12] Fernando Pereira and Touradj Ebrahimi, editors. *The MPEG-4 Book*. Prentice Hall PTR, 2002.

[13] N. Roma, T. Dias, and L. Sousa. Customisable core-based architectures for real-time motion estimation on fpgas. In *13th International Conference on Field Programmable Logic and Applications (FPL'2003)*, Lisboa - Portugal, September 2003.

[14] N. Roma and L. Sousa. Parameterizable hardware architectures for automatic synthesis of motion estimation processors. In *IEEE Workshop on Signal Processing Systems (SiPS)*, pages 428–439, Antwerpen - Belgium, September 2001.

[15] Nuno Roma and Leonel Sousa. Efficient and configurable full search block matching processors. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(12):1160–1167, December 2002.

[16] Jen-Chieh Tuan, Tian-Sheuan Chang, and Chein-Wei Jen. On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(1):61–72, January 2002.

[17] L. Vos and M. Stegherr. Parameterizable VLSI architectures for the full-search block-matching algorithm. *IEEE Transactions on Circuits and Systems*, 36(10):1309–1316, October 1989.