

LOW-POWER ARRAY ARCHITECTURES FOR MOTION ESTIMATION

Leonel Sousa

Dept. of Electrical Engin.
Instituto Superior Técnico/INESC
R. Alves Redol, 9, Lisboa-Portugal

Nuno Roma

Dept. of Electrical Engin.
Instituto Superior Técnico/INESC
R. Alves Redol, 9, Lisboa-Portugal

Abstract - This paper proposes new efficient low-power systolic architectures for Full Search-Block Matching (FS-BM) motion estimation. These architectures allow to eliminate unnecessary computations, reducing the power consumption while preserving the optimal solution and the throughput. The new and traditional systolic architectures for motion estimation are compared in what concerns the required hardware and the power consumption.

INTRODUCTION

Block Matching (BM) motion estimation is used to exploit temporal redundancy in video coding [1]. Among the algorithms used for motion estimation by block matching, the FS-BM algorithm uses an exhaustive search to find the candidate block that is closest to the reference block.

FS-BM estimation processors typically adopt systolic array architectures [3] and are responsible for the major part of power consumption in video coding systems [4]. Reduction of power consumption is one of the major challenges for implementing portable multimedia systems. To reduce the computational complexity and the power consumption associated to BM processors, several fast but approximate search procedures have been proposed [2, 6]. These procedures restrict the search space but do not guarantee the optimal solution. Reference [5] presents a low power systolic array architecture for FS-BM motion estimation. It eliminates unnecessary computations by computing a conservative estimation of the distortion values before computing the exact value. The main drawback of this architecture is the considerable extra hardware required to estimate the distortion.

In this paper, low-power systolic array architectures for FS-BM are proposed. The reduction of power consumption is obtained by disabling the Processing Elements (PE) as soon as the distortion values become greater than the minimum distortion value already computed. The architectures were modelled and synthesised using VHDL tools. The results show the functional validity of the proposed architectures, confirm the low extra hardware required to implement the low-power procedure and predict a reduction of power consumption in the order of 50%.

LOW-POWER SYSTOLIC ARCHITECTURES

For a reference block composed of $n \times n$ pixels and a search area with $(n + p) \times (n + p)$ pixels (for simplicity we consider that p is an odd number), Eqs. 1 and 2 are used by the FS-BM algorithm to compute the mean absolute error $D(l, c)$ and to identify the motion vector (u, v) .

$$D(l, c) = \sum_{i=0}^{n-1} D_i(l, c) ; D_i(l, c) = \sum_{j=0}^{n-1} |x_t(i, j) - x_{t-1}(l+i, c+j)| \quad (1)$$

$$(u, v) = (l, c) |D_min \quad \lfloor -p/2 \rfloor \leq l, c \leq \lfloor p/2 \rfloor . \quad (2)$$

In these equations, $x_t(i, j)$ and $x_{t-1}(i, j)$ represent the pixels in the reference and candidate blocks, respectively. The same final results for the error and the motion vector are obtained if the computation of new Absolute Differences (AD) is stopped when the value of $D(l, c)$ becomes greater or equal than the temporary value of D_min (D_min_t) for the Area Already Searched (AAS)—in Fig. 1, the AAS covers the range from $(\lfloor -p/2 \rfloor, \lfloor -p/2 \rfloor)$ to $(0, -1)$.

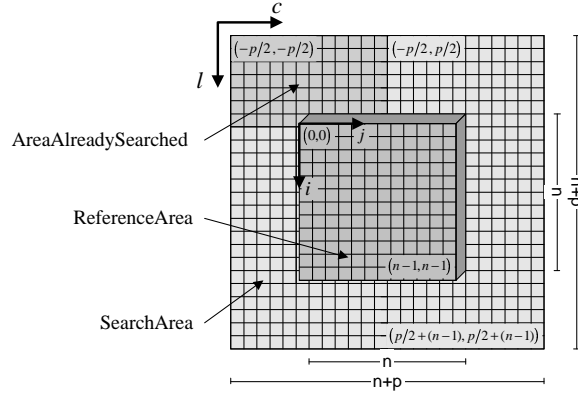


Figure 1: Search area and reference block spaces.

The values of $D_i(l, c)$ are computed and accumulated for every i ($D^i(l, c)$). When $D^i(l, c)$ becomes greater or equal than D_min_t for the ASS, the sum of absolute differences for rows $i + 1$ to $n - 1$ are not computed (Eq. 3). The value of D_min_t is updated during the processing.

$$\begin{aligned} D^i(l, c) &= D^{i-1}(l, c) + D_i(l, c) ; 0 \leq i \leq n - 1 \\ D_i(l, c) &= \begin{cases} \sum_{j=0}^{n-1} |x_t(i, j) - x_{t-1}(l+i, c+j)|, & \text{if } D^{i-1}(l, c) < D_min_t \\ D_{i-1}(l, c), & \text{otherwise} \end{cases} \\ D(l, c) &= D^{n-1}(l, c) \\ D_min_t &= \min(D(l, c)) \{ (l, c) \in AAS \} \end{aligned} \quad (3)$$

The power consumption of the architectures is reduced by blocking new values of x to enter into the PEs, preventing the circuits to switch when $D_i(l, c) = D_{i-1}(l, c)$ in Eq. 3. For reducing power consumption with such pipeline architectures, it is necessary to compute $D^{i-1}(l, c)$ and D_{min_t} on time to block the computation of $D_i(l, c)$. This problem can be solved by spacing out in time the computation of $D_i(l, c)$ and $D_{i+1}(l, c)$ for a given (l, c) , *i.e.* by computing in sequence partial distortion values for different candidate blocks.

```

D_min_t := ∞ ; BLOCKING(, , 0) := FALSE
for l = ⌊-p/2⌋ to ⌊p/2⌋
  for i = 0 to n - 1
    for c = ⌊-p/2⌋ to ⌊p/2⌋
      for j = 0 to n - 1
        if BLOCKING(l, c, i) = FALSE then
          D(l, c, i, j + 1) := D(l, c, i, j) + |x_i(i, j) - x_{t-1}(l + i, c + j)|
        end {j}
        D(l, c, i, n) := D(l, c, i, n) + D(l, c, i - 1, n)
        if D(l, c, i, n) ≥ D_min_t then BLOCKING(l, c, i + 1) := TRUE
      end {c}
    end {i}
  for c = ⌊-p/2⌋ to ⌊p/2⌋
    if D(l, c, n - 1, n) < D_min_t then
      D_min_t := D(l, c, n - 1, n) ; (u, v) := (l, c)
    end {c}
  end {l}

```

Figure 2: Single assignment code to derive low-power linear architecture.

The FS-BM algorithm can be described by several different dependence graphs (DGs), corresponding to different organizations of the computations in the four dimensional index space (l, c, i, j) [3, 7]. The single assignment code in Fig. 2 presents a suitable FS-BM algorithm representation for deriving low-power linear systolic architectures. The absolute differences corresponding to a single row of the reference block are accumulated for a line of candidate blocks in sequence (c loop). The $(p + 1)$ elements of the *BLOCKING* bit vector identify when the computation for any one of those candidate blocks should be disabled. The $(p + 1)$ elements of this vector are updated in every iterations of the i loop, and the value of D_{min_t} is updated after processing a line of candidate blocks.

The low-power linear architecture proposed in Fig. 3 is derived from the single assignment code presented in Fig. 2. Candidate blocks are placed in raster format at the input of the first PE. For each row i of the reference block, the sums of absolute differences corresponding to all $(p + 1)$ candidate blocks in a line are computed in consecutive clock cycles: $2 \times n$ clock cycles after $x_{t-1}(l + i, \lfloor -p/2 \rfloor)$ appears on the input, the value of $D_i(l, \lfloor -p/2 \rfloor)$ is provided by the last PE to one of the final adder inputs. This adder sums $D_i(l, \lfloor -p/2 \rfloor)$ with $D^{i-1}(l, \lfloor -p/2 \rfloor)$, provided by the output shift register. The value of the sum is then compared with the minimum distortion value found for the AAS and the comparison result is stored in the blocking register. At the same time, the

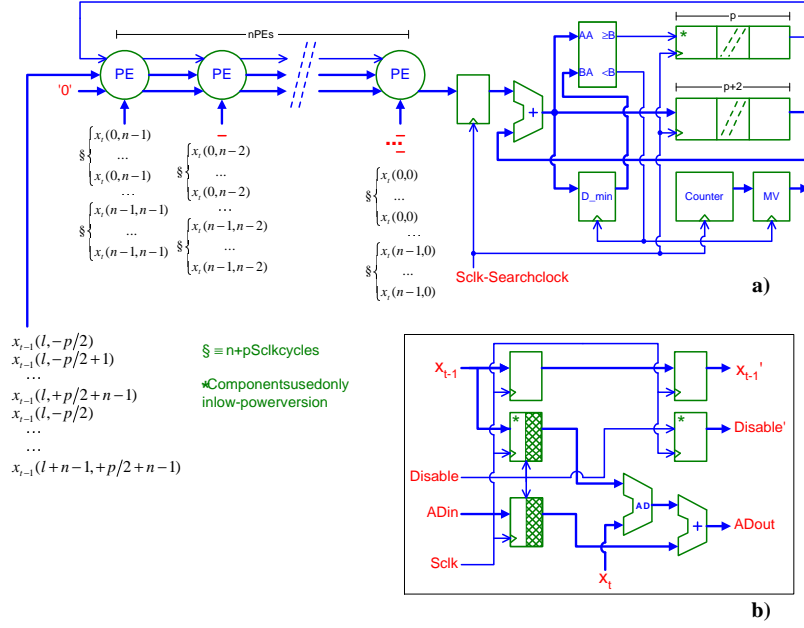


Figure 3: Low-power architecture : a) linear array; b) processing element.

updated value of D^i is stored in the output shift register. This processing is repeated in the next p clock cycles for the remaining p candidate blocks in line l . Every time a pixel of the first column of a candidate block appears in the array input, the blocking shift register provides a disable signal to eliminate the unnecessary computations. Blocking registers are introduced in each PE to prevent the internal AD and adder circuits from switching when the disable signal is asserted, as depicted in Fig. 3b. The disable signal is pipelined through the array, to match the pipelining of the distortion computation. Few additional registers (marked with '*') are required to implement the part of the algorithm designed to reduce the power consumption. This architecture fully implements the processing presented in Eq. 3, with no restrictions on p and n values.

Processors with linear array architectures require a high working frequency. For the proposed architecture, about $clk_B = (n + p) \times n \times (p + 1)$ clock cycles are necessary to process a reference block and $clk_I = clk_B \times M$ clock cycles are needed to process an entire image with M blocks. The minimum working frequency for a frame rate F is $f_{min} = clk_B \times M \times F$. Therefore, for CIF ($M_{CIF} = 18 \times 22$) and QCIF ($M_{QCIF} = 9 \times 11$) images and ($n = 16$, $p = 31$) the minimum working frequency is $f_{CIF} = 190.6MHz$ and $f_{QCIF} = 47.7MHz$, for a frame rate of 20 fps.

The minimum working frequency value can be decreased by using multiple linear arrays, in order to process in parallel the distortion for different blocks. For example, if two linear arrays are used, the values of the working

component	#LP (#non LP)
Inverter	968
2 inp. OR	572
3 inp. OR	174
2 inp. AND	1165
3 inp. AND	38
2 inp. NAND	2296
2 inp. XOR	24
D-Flip-flops	1420 (1245)
TOTAL	6657 (6482)

(a) Number of components required

video clip	% of Power saved			aver.
	quantization step			
	4	8	16	
Carphone	54.1	51.7	48.5	51.4
Claire	52.9	50.5	48.8	50.7
Foreman	56.3	53.6	50.1	53.3
Susie	51.1	48.1	44.7	47.9
Trevor	55.0	52.6	49.3	52.3
Salesman	51.8	48.5	45.4	48.5
average	53.4	50.8	47.7	50.6

(b) Percentage of power saved with the low-power architecture

Table 1: Simulation Results (LP–Low-Power).

frequency referred in the previous paragraph are reduced to half. The control of these two linear arrays is quite independent, assuming that the frame buffer supports two simultaneously accesses to different positions. Another way of decreasing the minimum working frequency is to design low-power 2-D array architectures based on Eq. 3. However, it is more difficult to solve the dependency problem in 2-D structures, because two loops have to be processed in parallel. Consequently, the power consumption reduction is lower [8].

EXPERIMENTAL RESULTS

As it was mentioned before, the proposed architecture was modelled and synthesised using VHDL synthesis tools [10]. Simulation results show the functional validity of the architecture.

In Table 1(a) it is summarized the results of the synthesis for the proposed architectures, in what concerns the total number of logic gates and type-D flip-flops. The total number of gates for the low-power processor is approximately the same as the non low-power one, while the number of flip-flops is increased by 175 to cope with the extra eight bit register and one extra flip-flop in each PE, and the p positions FIFO memory: $(8 + 1) * n + p = 175$.

A simulator has been specifically developed to estimate power consumption of architectures for motion estimation, using a simple approach similar to the referred in [5]. The unit of energy consumption (\mathcal{S}) is defined as the amount of energy consumed in the computation of a sum. An absolute difference unit consumes $2\mathcal{S}$ and a comparator consumes \mathcal{S} .

The simulation program was integrated and tested on a software video CODEC for the H.263 standard [9]. Search areas of 47×47 pixels were considered for macroblocks with 16×16 pixels and for integer motion vectors in the range from $(-16, -16)$ to $(+15, +15)$.

Table 1(b) presents the average reduction of power consumption achieved

with the proposed architecture, regarding to the non low-power version, for several video sequences. The results were obtained by coding 40 consecutive frames (QCIF format) in *INTER* mode with three different quantization steps. It can be stated that the power consumption of the proposed low-power architecture is only about 50% of the power consumption of a traditional one.

CONCLUSIONS

New low-power linear array architectures were proposed in this paper. They require a simple control scheme and low extra hardware, regarding to the well-known array architectures. Simulation results show that the proposed array architecture saves about 50% of the power required by the traditional architectures.

References

- [1] V. Bhaskaran and K. Konstantinides, "Image and Video Compression Standards, Algorithms and Architectures", *Kluwer Academic Publishers*, Boston, 1995.
- [2] T. Koga, *et al.*, "Motion Compensated Interframe Coding for Video Conferencing", Proc. Nat. Telec. Conf., Nov. 1981, pp. G 5.3.1-G 5.3.5.
- [3] P. Pirsch, N. Demassieux and W. Gehrke, "VLSI Architectures for Video Compression—A Survey", *Proc. of the IEEE*, vol. 83, no. 2, February 1995, pp. 220-246.
- [4] K. Parhi, "Low-Power Multimedia DSP Systems", Proc. of 1997 Int. Conf. on VLSI and CAD (ICVC), Seoul, Korea, Oct. 1997, pp. 10-17.
- [5] V. Do and K. Yun, "A Low-Power VLSI Architecture for Full-Search Block-Matching Motion Estimation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 4, August 1998, pp. 393-398.
- [6] B. Natarajan, B. Vasudev and K. Konstantinides, "Low-Complexity Algorithm and Architecture for Block-Based Motion Estimation Via One-Bit Transforms", Proc. *ICASSP*, May 1996, pp. 3244-3247.
- [7] S. Y. Kung, "VLSI Array Processors", *Prentice Hall*, New Jersey, 1988.
- [8] L. Sousa, "Applying Conditional Processing to Design Low-Power Array Processors for Motion Estimation", *to appear* in the Proc. of Int. Conf. on Image Processing, Kobe, Japan, October 1999.
- [9] L. Sousa, *et al.*, "On the Development of a video CODEC for Low Bivariate Communication in General Purpose Computers", Proc. of IASTED AI'99, Innsbruck, Austria, Feb. 1999.
- [10] "Design Analyzer Reference Manual", *Synopsys Inc.*, 1997.