VLSI DESIGN FOR LOW-POWER DATA-ADAPTIVE MOTION ESTIMATION

Luca Fanucci[♦], *Sergio Saponara*[◊]

[♦] CSMDR, National Research Council, Via Diotisalvi 2, I-56122 Pisa, Italy
Tel: +39 050 568 668, Fax: +39 050 568 522, e-mail: luca.fanucci@iet.unipi.it

[◊] Department of Information Engineering, University of Pisa, Via Diotisalvi 2, I-56122 Pisa, Italy
Tel: +39 050 568 557, Fax: +39 050 568 522, e-mail: sergio.saponara@iet.unipi.it

ABSTRACT

A data-adaptive motion estimation algorithm and its low-power VLSI implementation are presented in this paper. Basically, the proposed technique exploits the input data variations to dynamically reconfigure the search window size of an exhaustive block-matching search. As proved by computer simulations, the same high performance of the conventional full-search approach is achieved for a remarkable reduction of the circuit power consumption.

1. INTRODUCTION

Motion Estimation (ME) is a key issue in any video compression system since it exploits the temporal correlation between adjacent frames in a video sequence to reduce the data interframe redundancy [1,2]. ME has also become an important operation in the field of pre- and post-processing video algorithms as image key feature extraction, adaptive coding and filtering, scene change detection, frame up-conversion [3,4].

A straightforward technique for performing ME is that of full-search block-matching (FS) [1,2]: the current frame of a video sequence is divided into not overlapped *N*x*N* blocks (reference block) and, for each of them, a block in the previous frame (candidate block) is exhaustively searched for the best matching within a search window with maximum horizontal and vertical displacements of *p*. The matching algorithm consists in computing the Sum of Absolute Differences (SAD) between the blocks. If *a(i,j)* and *b(i,j)* are the pixels of the relevant reference and candidate blocks and *m* and *n* are the coordinates of the Motion Vector (MV) (i.e. the position of the candidate block within the search window), the SAD is defined as (with $-p \le m,n \le p-1$):

$$SAD\,(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \left| a(i,j) - b(i+m,\, j+n) \right| \qquad (1).$$

This distortion is computed for all the $4p^2$ possible positions of the candidate blocks within the search window. The block corresponding to the minimum distortion (SAD_m) is used for prediction; its MV is given by $MV = (m,n)\big|_{SAD_m}$ with $SAD_m = \min_{(m,n)} [SAD(m,n)]$ (2).

This exhaustive approach achieves optimal performance in terms of PSNR (Peak Signal-to-Noise Ratio) for a given compression factor but at the expenses of high computational burden and data bandwidth with a quadratic dependence on the search window size. For example, for a 30 Hz CIF format with typical values *p* = 16, *N*=16, over 3×10^9 absolute difference operations per second are required. The high computational load and memory access requirements lead to high power consumption. Therefore the conventional FS approach is not suitable for battery-supplied applications where the low-power constraint is mandatory: 3^G mobile phones, CMOS camera, personal digital assistants, wireless surveillance terminals and tele-assistive technologies for elderly/disabled people. Reducing power consumption is also a key point in highly integrated system-on-a-chip to avoid heat removal problems which require the use of expensive packaging and cooling mechanisms.

To address the above issues, in this paper, a VLSI design for low power data-adaptive ME is presented. The proposed technique exploits the input data variations to dynamically reconfigure the search window size of an exhaustive block-matching search. This way it features the same high coding efficiency of the FS approach but, avoiding unnecessary computation, it reaches a remarkable power consumption reduction.

After this introduction Section 2 gives an overview of ME techniques for low-power applications. Section 3 describes the new data-adaptive algorithm called enhanced window-follower. Section 4 details the relevant VLSI architecture. Section 5 presents the implementation results in a sub-micron CMOS technology. Finally, some conclusions are drawn in Section 6.

2. PREVIOUS WORKS

Several ME algorithms have been proposed in literature where the complexity of the search is traded-off with optimality of the match: Three Step Search, Four Step Search, 2D-Log Search, Cross Search [1]. They reduce the number of candidate blocks investigated for each reference block without exploiting the statistics of the input data. Instead of determining the global minimum over the search range usually a local minima is obtained.

On the contrary the class of predictive algorithms [5,6] exploits the spatial and temporal correlation of the video motion field. The MV of a given block can be predicted from a set of initial MV candidates, selected from its spatio-temporal neighbours, according to a certain law. To further reduce the estimation error a refinement process is performed using the predicted MV as the starting point. For low bit-rate applications (tens of Kbits/s) it is possible to achieve the same high coding efficiency of the FS for a computational load reduction greater than one order of magnitude. However, the higher is the bit-rate of the considered application the worse is the algorithm performance.

To overcome the limits of the above fast ME techniques, in the next Sections we present an exhaustive block-matching search which exploits the input data variations to avoid unnecessary computation thus reducing circuit power consumption.

3. DATA-ADAPTIVE MOTION ESTIMATION

3.1 Window Follower Algorithm

The motion activity of a scene directly relates to the values of the relevant MV fields. In case of high motion activity a large search window size permits to get a MV which results in small estimation error. When the amount of motion in the scene is small then searching over a large search area is not necessary. Based on this approach Minocha and Shanbahg proposed in [7] a Window Follower (WF) ME technique in which the search window size for all the blocks of a frame depends on the maximum displacement in the MV field of the previous frame. The size of the window follows the MV field. For the generic ith frame the algorithm is:

Step 1. From the complete list of MV of the previous (i-1)th frame compute the maximum MV displacement (hereafter called S).

Step 2. Compute the MV for all the reference blocks of the ith frame by adopting an exhaustive search with a window displacement p=S+1. For the first frame p=$p_{\max}$.

For sequences with gradual motion changes this approach results in an average 60% power reduction with respect to the FS one for a similar coding efficiency [7]. The WF algorithm fails in case of sudden motion changes or frames with objects characterised by different motion activities. Indeed the increment of the search window displacement in the Step 2 is not enough to follow rapid motion changes. Moreover the algorithm acts likewise for all the blocks in a frame.

3.2 Enhanced Window Follower Algorithm

We overcome the limits of the WF approach by i) adopting the SAD_m values as measure of the efficiency of the ME; ii) exploiting the spatial correlation in the motion field. As a matter of fact, in case of interframe coding, a high value of SAD_m relevant to a processed block determines a high prediction error. This is why we envisage to compare the SAD_m related to a certain block with proper thresholds in order to select the optimal search window size for the successive blocks. For the generic ith frame the Enhanced WF (EWF) algorithm is:

Step 1. From the complete list of MV of the previous (i-1)th frame compute the maximum MV displacement (hereafter called S).

Step 2. Compute the MV for all the reference blocks of the ith frame by adopting an exhaustive search with a window displacement p_j, for the generic jth block, sized according to the following rules:

i) $SAD_{mj-1} \geq T_1 \rightarrow F = 1$ and $p_j = p_{\max.}$

ii) $T_2 \leq SAD_{mj-1} < T_1$ and $F = 1 \rightarrow p_j = 1+\max(S, s_{j-1})$

$T_2 \leq SAD_{mj-1} < T_1$ and $F = 0 \rightarrow p_j = 1+S$

iii) $SAD_{mj-1} < T_2$ and $F = 1 \rightarrow p_j = \max(S, s_{j-1})$

$SAD_{mj-1} < T_2$ and $F = 0 \rightarrow p_j = S$

For the first frame p=$p_{\max}$.

SAD_{mj-1} and s_{j-1} represent the SAD_m and the maximum MV displacement for the (j-1)th block. T_1=4096 and T_2=2048 are two thresholds used to evaluate the efficiency of the motion prediction. Their values have been derived from computer simulations of typical video sequences with different grades of dynamism. F is a flag which is set to 0 at the beginning of each frame computation. During the ME processing if F=0 the search size decisions depend only on the MV field of the previous (i-1)th frame through the variable S. If F=1 the search size decisions depend also on the MV field of the current ith frame through the variable s_{j-1}.

3.3 Performance Results

A quality analysis, based on several test conditions, demonstrates the efficiency of the proposed ME technique with respect to the FS approach. The evaluation of the EWF algorithm has been made using the TMN implementation of the H.263+ coder [8]. The only piece of code modified from the original software was the ME. Fig. 1 shows the values of the SAD_m, averaged on each frame, for the FS, WF and EWF algorithms considering a CIF test pattern composed of 30 frames Akiyo plus 30 frames Foreman. A zoom of the frames around the scene change is also depicted. Fig. 2 shows the values of the search window size p, averaged on each frame, for the same test case.

The EWF approach behaves better than the WF. In case of gradual motion changes the use of the threshold T_2 avoids unnecessary search window increment (Fig. 2). When a sudden motion change or a scene change occurs, it is detected by the threshold T_1 and the search size is set to its maximum value (Figs. 1,2). This way EWF allows a quick recover of the best MV which minimises the prediction error. Furthermore, in this case the flag F is set to 1 and so the search size decisions will depend on the MV fields of both the previous frame and the current frame through the variables S and s_{j-1}.

Fig. 3 summarises PSNR results for FS and EWF obtained with a 256 Kbits/s channel and a mixed test pattern composed of four pieces of different CIF sequences (Akiyo, Container, Coastguard, Foreman). The two PSNR curves substantially coincide, also after the scene changes. Similar figures have been obtained comparing the EWF and the FS for other test cases.

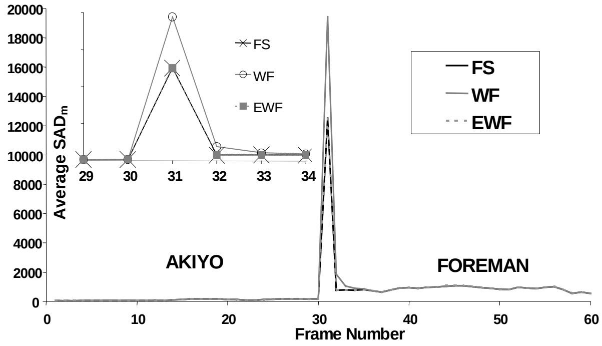

Fig 1. Average SAD$_m$

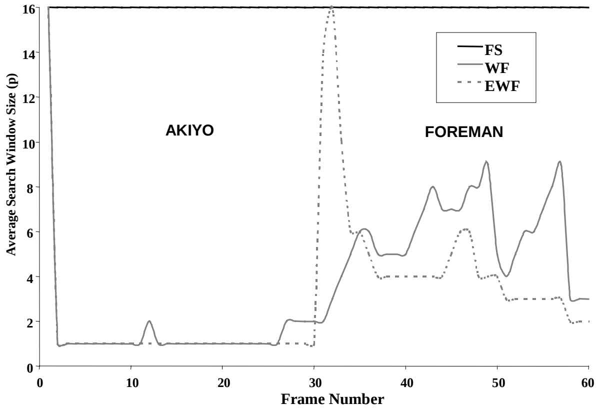

Fig 2. Search window size

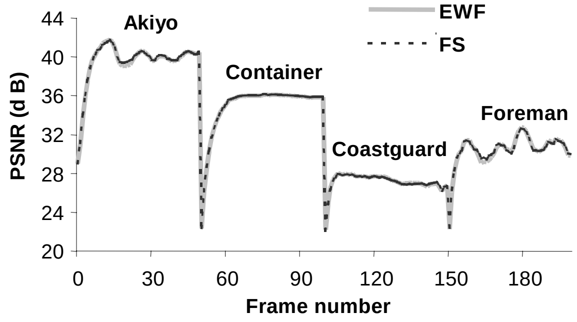

Fig 3. PSNR results at constant bit rate

4. VLSI ARCHITECTURE DESIGN

The proposed data-adaptive ME algorithm has been implemented by the VLSI architecture whose block diagram and circuit details are sketched in Figs. 4, 5. The architecture is composed by two main units:

i) A search engine which implements an exhaustive block-matching search with programmable window size.

ii) A window size controller which dynamically adjusts (signal p_ctrl in Figs. 4,5) the search area size according to the MV and SAD statistics. It implements the control rules proposed in Section 3.2.

All the relevant control signals are provided by the Control Unit of Fig. 4 which is a finite state machine.

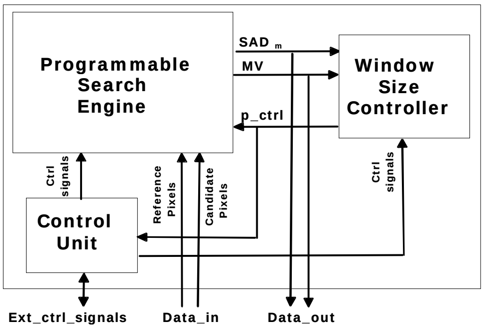

Fig 4. VLSI architecture overview

4.1 Programmable Search Engine

The programmable search engine is based on an intellectual property (IP) VLSI macrocell proposed by the authors in [2]. The cell is fully parametric in terms of block size (N) and maximum window size ($p_{\max}$). For a sake of clarity Fig. 5 details the VLSI architecture for the example case N=4 and $p_{\max}$=4. Based on a hardware multiplexing strategy it elaborates the SAD and the MV for a NxN block starting from the corresponding four N/2xN/2 sub-blocks by means of a systolic array of processing elements (PE) sized for N/2. Each PE implements at pixel level an absolute difference operation. The N/2 partial sums of each row of PE are added in the Adder Tree block as shown in Fig. 5. The proper timing of the candidate block pixels in the PE array is obtained by (N/2-1)(2p-2) shift registers (SR) cascaded to the PE columns along the y line. The two MDD (Minimum Distortion Detection) blocks of the MV processor, starting from the SAD(m,n) sequentially provided by the Adder Tree, calculate the SAD$_m$ and the (m, n) coordinates of the relevant MV for both N/2xN/2 and NxN blocks. This processing is made concurrently thus allowing the implementation of the advanced prediction (AP) mode foreseen by main coding standards. An important feature of this IP is the possibility to dynamically program the value of p within the interval [1, $p_{\max}$]. This is achieved by controlling the used length of each column of the SR matrix by means of a barrier of N/2-1 multiplexer with $p_{\max}$ vias. For example, by selecting the hth via of the multiplexer, the equivalent length of each SR chain results to be 2h-2, which corresponds to an equivalent value of p=h.

4.2 Switching Activity Reduction

The designed IP is characterized by a continuos input data flow and block distortion calculation. It is able to process each block NxN in $4{\cdot}(2p+N/2-1)^2$ clock cycles being $(2p+N/2-1)^2$ the number of pixels relevant to a search displacement p for a block N/2xN/2. Starting from the above expression and given the video format (frame rate R and frame pixel number D) and the architecture configuration (N, p), the clock frequency required for the real time processing is provided by the formula:

$$4{\cdot}R{\cdot}D/N^2 \cdot (2p+N/2-1)^2 \qquad (3),$$

which depends on p^2. Reducing the value of p by a factor of K, according to the EWF algorithm, it is possible to elaborate the same video format with a clock frequency reduced by a factor of roughly K^2. The frequency reduction directly yields a cut of the circuit switching power. As proved in Section 3.3 this does not imply any significant degradation of the image quality. A direct silicon implementation of this approach is not suitable since it would require the on-chip generation of p_{max} different clock signals (at least 16 for cases of practical interest). This problem can be overcome by adopting a single clock signal with a frequency sized for the maximum displacement $p=p_{max}$. In this case the above switching power reduction can be achieved by exploiting a clock-gating approach. Given $p \leq p_{max}$ the window displacement selected for a reference block, the whole search engine works for only $4 \cdot (2p+N/2-1)^2$ clock cycles while its clock is gated for the remaining cycles. The clock-gating signal (En_clk in Fig. 5) is provided by the Control Unit according to the selected p_ctrl value.

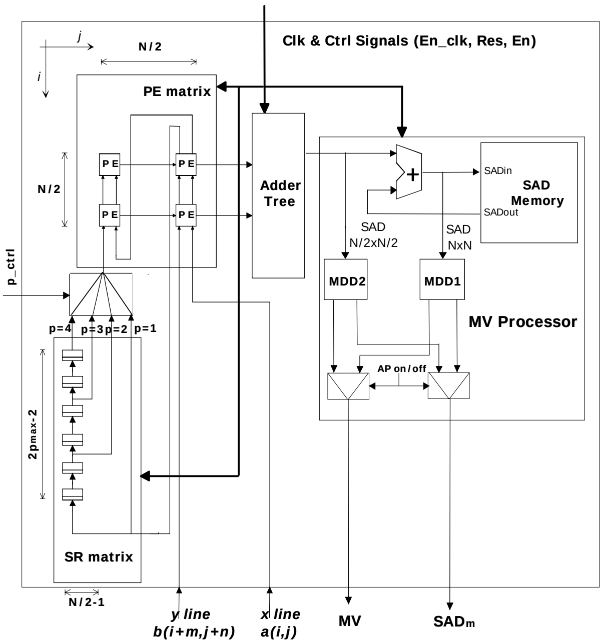

Fig 5. Search Engine (example case N=4, p_{max}=4)

5. VLSI RESULTS

The proposed VLSI architecture has been implemented in a 0.25 µm, 2.5 V, 6 metal levels CMOS standard-cell technology for the case N=16 and p_{max}=16. It permits the concurrent processing of blocks sized 16×16 and 8×8. The resulting ASIC is characterised by a complexity of about 35 Kgates plus a dual-port RAM of 12 Kbits. Following the approach proposed in Section 4.2 the clock frequency has been sized for the case $p=p_{max}$ according to eq. (3). For the real time processing of 30 Hz (R=30) QCIF (D=176x144 pixels), CIF (D=352x288 pixels) and 4CIF (D=704x576 pixels) video formats it amounts to respectively 18.07 MHz, 72.28 MHz and 289.12 MHz. The ASIC has been characterised in terms of power consumption by gate-level simulations. The results for QCIF and CIF video sequences with different grades of dynamism are presented in Table 1. Both the performance of conventional FS (without clock gating and p programmability) and EWF techniques are presented. The latter achieves a circuit power saving up to 90 % for a maximum power consumption below 12 mW and 47 mW for the QCIF and CIF cases.

	QCIF		CIF	
	EWF	**FS**	**EWF**	**FS**
Akiyo	3.85	43.72	15.35	174.94
Foreman	11.71	44.52	46.99	179.45
Coastguard	7.97	48.38	32.15	197.55

Table 1. Power results (mW) for different test sequences

6. CONCLUSION

In the paper the algorithm and VLSI architecture design for low power data-adaptive motion estimation has been presented. An exhaustive block-matching search exploiting input data variations is adopted to avoid unnecessary computation and thus reducing circuit power consumption. A quality analysis, based on several test conditions, confirms the efficiency of the proposed algorithm with respect to the FS one. The experimental results demonstrate the effectiveness of our approach even in case of sudden motion changes. The algorithm has been implemented by a systolic VLSI architecture which automatically adjusts the search area size according to MV and SAD statistics of the input video signals. By using a proper clock gating strategy, the reduced computational load directly yields a circuit power saving up to 90 %. The maximum power consumption is below 12 mW and 47 mW for the QCIF and CIF cases. The proposed technique is thus suitable for both battery-supplied terminals and highly integrated system-on-chip video applications.

REFERENCES

[1] P. Kuhn, Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation, Kluwer Academic Publishers, 1999

[2] L. Fanucci, S. Saponara, L. Bertini, A parametric VLSI architecture for video motion estimation, Integration The VLSI Journal, vol.31, n.1, 2001, pp. 79-100

[3] A. Lan et al., Scene-context-dependent reference-frame placement for MPEG video coding, IEEE Trans. on Circ. and Syst. for Video Technology, vol.9, n.3, 1999, pp.478-489

[4] N. Vasconcelos and F. Dufaux, Pre and post filtering for low bit rate video coding, Proc. IEEE International Conference on Image Processing, 1997, pp. 291-294

[5] F. Kossentini, Y. Lee, M. Smith, R. Ward, Predictive RD optimized motion estimation for very low bit-rate video coding, IEEE JSAC, vol. 15, n. 9, 1997, pp. 1752–1763

[6] L. Fanucci et al., Power optimization for H.263/MPEG-4 VLSI video coding, Proc. IEEE/WSES SSIP'01, pp. 2521-2527

[7] J. Minocha and N. Shanbhag, A low power data-adaptive motion estimation algorithm, Proc. IEEE Workshop on Multimedia Signal Processing, 1999, pp. 685-690

[8] Telecom Standardization Sector of ITU, Video Codec Test Model, Near Term, Version 10, TMN10 ITU-T, 1998