Low power distance measurement unit for real-time hardware motion estimators

Tiago Dias[2,3], Nuno Roma[1,3], and Leonel Sousa[1,3]

[1] IST / [2] ISEL / [3] INESC-ID
Rua Alves Redol, 9
1000-029 Lisboa, PORTUGAL
{Tiago.Dias, Nuno.Roma, Leonel.Sousa}@inesc-id.pt

Abstract. Real-time video encoding often demands hardware motion estimators, even when fast search algorithms are adopted. With the widespread usage of portable handheld devices that support digital video coding, low power consideration becomes a central limiting constraint. Consequently, adaptive search algorithms and special hardware architectures have been recently proposed to perform motion estimation in portable and autonomous devices. This paper proposes a new efficient carry-free arithmetic unit to compute the minimum distance in block matching motion estimation. The operation of the proposed unit is independent of the adopted search algorithm and of the used prediction error metric, simultaneously speeding up motion estimation and significantly reducing the power consumption. Moreover, its low latency is particularly advantageous when partial distance techniques are applied to further reduce the power consumption. Experimental results show that the proposed unit allows to reduce the computation time in about 40% and it consumes 50% less power than commonly adopted architectures.

1 Introduction

Motion Estimation (ME) is a central operation in video encoding, in order to exploit temporal redundancy in sequences of images. However, it is also by far the most computationally costly part of a video CoDec [1]. Among the several algorithms that have been proposed, the Block-Matching Motion Estimation (BME) is the most adopted method. Using this strategy the distance between each block of the reference frame and a set of candidate prediction blocks, defined within reduced areas of the previous frame (search areas), is computed to obtain the corresponding motion vector (MV). Nevertheless, even though the complexity of this method depends on the adopted distance metric and on the selected search algorithm, only hardware solutions are usually able to fulfill the real-time requirement.

In the last few years, it has been widely claimed that the first generation of search algorithms often reveals to be inadequate for portable devices, powered by batteries. Examples of these algorithms are the optimal Full-Search Block-Matching (FSBM) and other non-optimal but faster search algorithms, such as

the Three-Step-Search (3SS) [2], the Four-Step-Search (4SS) [3] and the Diamond Search (DS) [4]. Therefore, more efficient search algorithms have been developed by taking advantage of temporal and spacial correlations of the MVs in order to adapt and optimize the search pattern, thus avoiding unnecessary computations and memory accesses. Examples of these recent fast search techniques are the Motion Vector Field Adaptive Search Technique (MVFAST) and the Enhanced Predictive Zonal Search (EPZS) [5, 6].

At the same time, several specific architectures have also been proposed to implement the different search algorithms [7, 8]. Very recently, an Application Specific Instruction Set Processor (ASIP) was proposed for adaptive video motion estimation [9]. The common feature of all these hardware architectures is that they all use, at their processing core, an arithmetic unit specially designed to compute the distance metric between the current block and the considered prediction blocks, as well as to compare these distances in order to find the best match.

The investigation of efficient hardware units for estimating MVs has been mainly focused either on the implementation of the overall algorithms [7] or on the computation of the distance metrics (such as the Sum of Absolute Differences (SAD) [10] or the Mean Square Error (MSE)). However, it has not yet been proposed any efficient structure that avoids the usage of carry-propagate type adders to compute and compare the obtained values of the distance measure, in order to obtain the minimum distance value and thus the best match. This issue presents several drawbacks, since carry-propagate type adders are rather slow units with a high cost. Furthermore, the application of the usual alternative accelerating techniques, such as the application of pipelining, can often be disadvantageous for fast BME, since the data dependency in fast and adaptive algorithms implies sequentiality in the processing.

In this paper it is proposed a very efficient hardware unit that avoids the usage of carry-propagate units to compute and compare any considered distance metric. This unit exhibits null latency, reduced cost and power consumption, speeding up the evaluation of the BME condition by using carry-free arithmetic. Due to its null latency, the proposed unit can also be directly used to apply some of the power-saving techniques that have been proposed in the last few years, which evaluate the partial measures and disable a sub-set of the architecture modules in order to further reduce the power consumption [11].

2 Block-Matching Motion-Estimation Algorithm

In the widely adopted BME algorithm, for each macroblock of $N \times N$ pixels of the current image (z) it must be found the best matching prediction block in a reference frame ($\hat{z}$), according to the considered search algorithm and the used matching criterion. The MSE and the SAD are the most frequently used matching metrics. For a given distance metric (D), eq. 1 is computed for each

candidate MV $\boldsymbol{v}$ and the set of pixels γ in a macroblock:

$$D = \sum_{(i,j)\in\gamma} f(z_{i,j} - \hat{z}_{v_i+i,v_j+j}) \ ; \ f : |\ | \ (\text{SAD}) \ , \ [\ ]^2 \ (\text{MSE}) \left(z_{i,j} - \hat{z}_{v_i+i,v_j+j}\right)^2 \tag{1}$$

The BME algorithms then determine the MV corresponding to the "best match" ($\boldsymbol{V}$) for a given search region (ζ) and a specific metric:

$$\boldsymbol{V}(\zeta,\gamma) = \arg \overset{\boldsymbol{v}\in\zeta}{\min} \{D(\boldsymbol{v},\gamma)\}. \tag{2}$$

By considering a single iteration, i.e., a single candidate block t, this condition can be stated as:

$$\boldsymbol{V}(\zeta^t,\gamma) = \begin{cases} \boldsymbol{V}(\zeta^{t-1},\gamma) \ \text{if } D(\boldsymbol{v}^t,\gamma) > D(\boldsymbol{v}^{t-1},\gamma) \\ \boldsymbol{v}^t(\gamma) \qquad \text{otherwise.} \end{cases} \tag{3}$$

The optimal solution of this algorithm, given by the FSBM approach, is provided by searching over all possible candidate macroblocks in ζ and by using a metric that considers all pixels (γ) of the macroblock. To achieve fast BME, not only can the search range be restricted to a subset of vectors, but also can the matching criteria be simplified by only computing partial metrics [12]. Partial-distance search techniques [13] are particular cases of these fast schemes.

3 Proposed Architecture

The traditional hardware architecture of a generic block matching motion estimator is presented in fig. 1. Such structure is usually composed by:

- an arbitrary *metric computation unit*, where either the SAD, MSE or any other distance metric is computed using the input pixels of both the current and previous frames;
- an *accumulation unit*, that accumulates, one at a time, the computed partial values for the considered metric;
- a *best-match detection unit*, which compares the value of the considered metric for the prediction macroblock under processing with the value obtained for the best matching macroblock that was found up to the current instant.

As it is presented in this block diagram, carry-save adder structures are often used in the implementation of the *accumulation unit*, in order to avoid the introduction of carry-propagation delays in the accumulation critical path, thus maximizing the operating frequency. In fig. 2 it is presented the hardware structure that is traditionally adopted to update the MV in each iteration of the BME algorithm. Such structure is usually composed by a fast carry-propagate adder (e.g.: carry-look-ahead structure), to add the sum (S^t) and the carry (C^t) bit-vectors obtained from the *accumulation unit*, and a fast carry-propagate subtracter, to compare the current distance metric with the best one previously obtained. This subtracter issues the "Greater or Equal" (GE) signal, that will
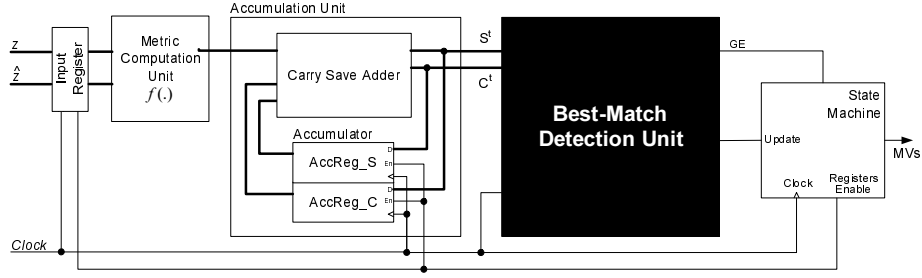
Fig. 1. Traditional hardware architecture of a generic block matching motion estimator

be used by the state machine to update the Best-Metric Register, corresponding to the current MV.

Nevertheless, this solution still reveals to be somewhat inefficient, since it does not prevent the need to use a carry-propagate adder structure in the best-match detector in order to compare the current metric value with the best one previously obtained. As an example, if power-saving techniques that abort the computation of the distance measure as soon as the partial value of the distance metric is greater that the one corresponding to the best matching macroblock [11] are applied, the carry-propagate structures present at the best-match detector will introduce a significant delay in the overall data-path, thus greatly conditioning the operation frequency.

To derive more efficient hardware structures to compute eqs. 1 and 2 in real-time, the following two specific properties of BME can be applied:

Property 1 - the distance D is a natural number, non-fractional and always positive;

Property 2 - the ultimate goal of BME is to compute the MV $\boldsymbol{V}$ (eq. 2) corresponding to the minimum distance measure (D), and not the value of D itself.

Property 1 can be used to simplify the comparison in each iteration t of eq. 3 provided that the strategy proposed in [10] to compare two m-bit numbers is also applied. By adopting a simplified notation, with $D^t \equiv D(\boldsymbol{v}^t, \gamma)$, it can be shown that:

$$D^t - D^{t-1} > 0 \ \Leftrightarrow \ 2^m - 1 - D^{t-1} + D^t > 2^m - 1 \ \Leftrightarrow \ \overline{D^{t-1}} + D^t \geq 2^m \quad (4)$$

To perform this comparison, it can be easily shown that only the carry out of the most significant bit $(m-1)$ must be computed, which corresponds to a much faster and simpler logic circuit than any carry-propagate subtracter required by the traditional architecture (see fig. 2).

On the other hand, *Property 2* provides the possibility to choose a representation for D that is more suitable to evaluate the condition stated in eq. 2. In fact, if the carry-save representation is also used to compute eq. 4, a single
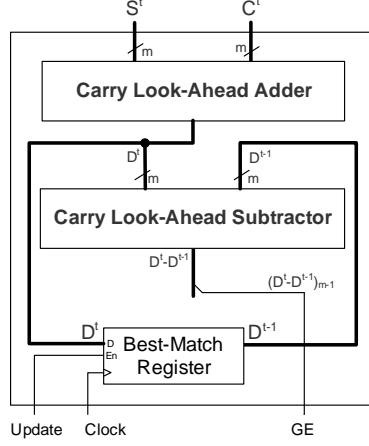
Fig. 2. Traditional hardware structure of the *best-match detection unit*

and unique representation can be applied to compute both eqs. 1 and 2, thus avoiding the inevitable penalty associated with carry-propagate operations.

Consequently, by storing the pair of carry-save $(m-1)$ bit vectors $\{C^{t-1}, S^{t-1}\}$, representing D^{t-1}, eq. 4 can be rewritten using the corresponding bit vectors computed at iteration t:

$$D^t - D^{t-1} > 0 \tag{5}$$
$$\Leftrightarrow \{C^t, S^t\} > \{C^{t-1}, S^{t-1}\} \tag{6}$$
$$\Leftrightarrow 2^m - 1 - (S^{t-1}) + 2^m - 1 - (2 \times C^{t-1}) + S^t + 2 \times C^t \geq 2^{m+1} - 1 \tag{7}$$

By replacing each multiplication by a left shift $(<< 1)$ and each subtraction by the corresponding one's complement value, eq. 7 can be written as:

$$1\#\overline{S^{t-1}}_{m-2:0} + \left(\overline{C^{t-1} << 1}\right)_{m-1:1} \#1 + (C^t << 1)_{m-1:1}\#1 + S^t \geq 2^{m+1} \tag{8}$$

where $\#$ denotes the concatenation operation and $(Z)_{y:x}$ the extraction of bits x to y of a binary number Z.

Hence, if the result of the left hand-side of inequality 8 is represented in the carry-save format $(\{C, S\})$, it can be shown that only bitwise logic operations and carry-free arithmetic are required for its evaluation. Once again, to perform the comparison only the carry-out of the most significant bit m must be computed, instead of the full sum value $(S + 2 \times C)$.

In fig. 3 it is depicted the proposed hardware structure for the *best-match detection unit*, directly obtained from the previous equations. With this new architecture, the carry-propagate type adder and subtracter of fig. 2 are replaced by two carry-save adders and a simple carry-out detector. This detector computes the carry-out signal that will be used to obtain the "Greater or Equal" (GE) signal, supplied to the main state machine of the motion estimator. It can be
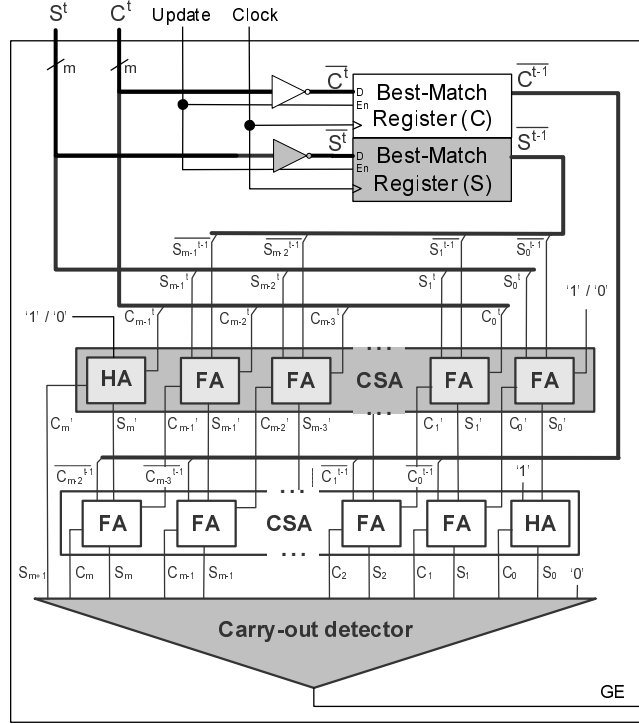
Fig. 3. Proposed carry-free hardware structure for the *best-match detection unit*

implemented using only the carry-generate descending branch of a carry-look-ahead adder structure, which provides a fast computation of the carry-out bit of two operands addition without actually computing the addition result.

It is also worth noting that the proposed architecture can still be applied with increased advantages to compare the partial computed distance metrics with a predefined threshold (T). This feature is particularly useful in the implementation of some of the last generation partial-distance search algorithms and techniques with strict power restrictions [13], which have been recently proposed to perform ME in portable and handheld devices:

$$\overline{T}_{m-1:0} + (C^t << 1)_{m-1:0} + S^t \geq 2^m \Rightarrow \boldsymbol{V}(\zeta^t, \gamma) = \boldsymbol{V}(\zeta^{t-1}, \gamma) \qquad (9)$$

The amount of hardware that is required for this approach is approximately one half of the original one, since only the components corresponding to the shaded blocks in fig. 3 are used.

4   Performance Evaluation

The performance evaluation of the proposed arithmetic unit for computing the minimum distance in BME was assessed using experimental results obtained for

Table 1. Implementation results obtained using the StdCell library based on a 0.18μm CMOS process

(a) Traditional architecture

Structure	F [$\boldsymbol{MHz}$]	Area [$\boldsymbol{\mu m^2}$]	Power [$\boldsymbol{mW}$]
Datapath	294.99	12269	12.75
Best-Match Detection Unit	363.64	7025	14.47

(b) Proposed architecture

Structure	F [$\boldsymbol{MHz}$]	Area [$\boldsymbol{\mu m^2}$]	Power [$\boldsymbol{mW}$]
Datapath	416.67	14213	13.78
Best-Match Detection Unit	497.51	8968	14.62

ASIC implementations of the considered structure. To do so, both the traditional and the proposed architectures of such unit were specified in the IEEE-VHDL hardware description language using fully structural and parameterizable descriptions of their structure and circuits based only on three simple 2-input logic gates: AND, OR and EX-OR. The fast carry-propagate adder and subtracter of the traditional architecture, see fig. 2, were implemented using carry-look-ahead adders with binary tree architectures [14].

The implementation in ASIC was performed using Synopsys synthesis tools and a high performance StdCell library from Virtual Silicon Technology Inc., which is based on a 0.18μm CMOS process from UMC [15]. The two architectures were synthesized for the typical operating conditions, $V_{dd} = 1.8V$ and $T = 25\ ^oC$, and using the "suggested 20k" wire model. Moreover, some constraints were also imposed to the synthesis tool in order to achieve a minimum area solution.

The obtained experimental results for the two implementations demonstrate the superiority of the proposed architecture over the traditional structure. From table 1 it can be seen that the proposed architecture allows for a reduction of about 40% in the computation time with a modest increase of the implementation area, about 16%. Nevertheless, despite such increase the two circuits present very similar power consumption requirements. In fact, when operating at its maximum operating frequency (416.67 MHz) the proposed architecture consumes only more 8% than the traditional circuit, also operating at its maximum operating frequency (294.99 MHz). However, such high operating frequencies are not required for low power digital video coding applications, such as mobile and portable handheld devices that use non-optimal search algorithms [2–6]. For example, to encode a CIF frame in real-time using the 4SS [3] algorithm the ME circuit does not need to operate at a frequency higher than 150 MHz. Consequently, by considering the proposed and the traditional architectures operating at the same frequency, the maximum allowed operating frequency of the traditional one, it can be shown that the proposed structure allows a significant reduction in the power consumption requirements, as depicted in table 2. The power consumption of the *best-match detection unit* is reduced by more than 60% and the reduction for the complete datapath is about 50%.

Table 2. Comparative evaluation of the power consumption of the traditional and of the proposed architectures

Structure	Traditional Architecture	Proposed Architecture	
Datapath	$12.75mW$	$6.83mW$	54%
Best-Match Detection Unit	$14.47mW$	$5.38mW$	37%

The proposed architecture is able to provide even greater power saving rates when used as a control mechanism to avoid needless calculations in the computation of the best match for a macroblock [13]. Such computations can be avoided by disabling all the logic and arithmetic units used in the computation of the adopted matching metric as soon as the partial value of the distance metric for the candidate block under processing exceeds the one already computed for the current macroblock. On average, this technique allows to avoid up to 50% of the required computations [11]. Consequently, a ME circuit using such technique and the proposed architecture for the *best-match detection unit* is able to reduce in up to 75% its power consumption requirements.

5   Conclusions

By analyzing the particular computational characteristics of block-matching motion-estimation, a complete carry-free arithmetic formulation was derived and an improved architecture was proposed for computing the minimum distance in block-matching motion-estimation algorithms. Experimental results show the practical importance of this novel structure in the design of efficient real-time motion estimators, regardless of the search algorithms and the matching metrics adopted by such circuits. By using the proposed arithmetic unit, it is not only possible to reduce the computation time in about 40%, with only a modest penalty in the cost of about 16%, but also to maintain similar power consumption requirements. Moreover, for low power digital video coding applications, such as mobile and portable handheld devices, where fast and adaptive search algorithms are used, the proposed architecture provides a very significant reduction of about 50% in the power consumption of the motion estimators. Since the proposed arithmetic unit exhibits no latency, it can also be directly used to avoid unnecessary calculations in the computation of the minimum distance value, by using partial distance values to disable specific hardware blocks of the motion estimators. In such cases, the proposed architecture is expected to reduce even further, to a minimum of 25%, the power consumption requirements of motion estimators for real-time video encoding.

References

1. Richardson, I.: H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. Wiley (2003)

2. Li, R., Zeng, B., Liou, M.L.: A new three-step search algorithm for block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology **4**(4) (1994) 438–442

3. Po, L.M., Ma, W.C.: A novel four-step search algorithm for fast block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology **6**(3) (1996) 313–317

4. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast block-matching motion estimation. IEEE Transactions on Image Processing **9**(2) (2000) 287–290

5. Tourapis, A., Au, O., Liou, M.: Predictive motion vector field adaptive search technique (PMVFAST) - enhancing block based motion estimation. In: Proceedings of SPIE - Visual Communications and Image Processing (VCIP), San Jose, CA (2001) 883–892

6. Tourapis, A.: Enhanced predictive zonal search for single and multiple frame motion estimation. In: Proceedings of SPIE - Visual Communications and Image Processing (VCIP), San Jose, CA (2002) 1069–1079

7. Roma, N., Sousa, L.: Efficient and configurable full search block matching processors. IEEE Transactions on Circuits and Systems for Video Technology **12**(12) (2002) 1160–1167

8. Saponara, S., Fanucci, L.: Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems. IEE Proceedings Computers and Digital Techniques **151**(1) (2004) 51– 59

9. Momcilovic, S., Dias, T., Roma, N., Sousa, L.: Application specific instruction set processor for adaptive video motion estimation. In: 9th Euromicro Conference on Digital System Design: Architectures, Methods and Tools - DSD'2006. (2006)

10. Vassiliadis, S., Hakkennes, E., Wong, S., Pechanek, G.: The sum-absolute-difference motion estimation accelerator. In: Proc. of the 24th Euromicro Conference. (1998) 559–566

11. Sousa, L., Roma, N.: Low-power array architectures for motion estimation. In: Workshop on Multimedia Signal Processing - MMSP'99, IEEE (1999) 679–684

12. Lengwehasatit, K., Ortega, A.: Probabilistic partial-distance fast matching algorithms for motion estimation. IEEE Trans. on Circuits for Video Techn. **11**(2) (2001) 139–152

13. Sousa, L.: A general method for eliminating redundant computations in video coding. IEE Electronics Letters **36**(4) (2000) 306–307

14. Roma, N., Dias, T., Sousa, L.: Fast adder architectures: Modeling and experimental evaluation. In: Proc. of the Conference on Design of Circuits and Integrated Systems (DCIS'03). (2003) 367–372

15. Virtual Silicon Technology Inc., Tech. Rep. v2.4: eSi-Route/11^{TM} High Performance Standard Cell Library (UMC 0.18μm). (2001)