http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (6): 1073-1077, 2006
ISSN 1812-5638

A Novel Coarse-to-fine Search Motion Estimator

Reeba Korah and J. Raja Paul Perinbam
Department of Electronics and Communication Engineering, Anna University, Chennai, India

Abstract: This study presents a new block matching motion estimation and its VLSI architecture design. The proposed Candidate Block and Pixel Sub-sampling (CBPS) algorithm reduces the number of candidate blocks selected to 5x(2p+1) against $(2p+1)^2$ in case of FSBMA. A new 4-queen pattern for pixel sub-sampling is also introduced. Our basic aim is to reduce the overall computations, so as to make data flow more regular and suitable for hardware. The loss in PSNR is very negligible from frame to frame and at the worst case, comes to 0.23 dB. The speed up factor achieved is 13.6. Computational complexity is reduced to 5.25%. We propose an architecture that is composed of a single PE and an adder to efficiently compute Sub-sampled Sum of Absolute Difference (SSAD). Also a modified search criteria called Reduced Bit Sub-sampled Sum of Absolute Difference (RBSSAD) is introduced. Simulation results show that our design is much more area and power efficient than many full search architectures, while maintaining good video quality and processing capability.

Key words: Motion estimation, block matching, sub-sampling, RBSSAD

INTRODUCTION

Block matching algorithms have been adopted for motion estimation by all the existing international standards related to video coding, such as ISO MPEG series (ISO/IEC 11 172-2, 1993; 1996; ISO/IEC 14 496-2, 1999) and the ITU-T H.262 series (H.261, 1993; H.263, 1998; H.264, MPEG-4, Part10, 2003). Motion Estimation (ME) removes temporal redundancy between frames and thus provide coding systems with high compression ratio. Since a ME module is usually the most computationally intensive part in a typical video coder (60-80% of the entire system), efficient implementation of ME is must. There exists a number of block matching algorithms out of which full search and coarse-to-fine search are very popular. Full search based method (Yang *et al*., 1989) is computationally very expensive, but yields a fairly accurate result, whereas coarse-to-fine-search algorithms are faster at the expense of accuracy. Three-step search (Jung and Lee, 2004; Seth *et al*., 2004) the 2-D logarithmic search, the conjugate directional search and four-step search (Po and Ma, 1996), diamond search and hexagonal search are some of the examples for coarse to-fine search algorithms. Among these, the three step search block-matching algorithm is considered as one of the best algorithms. But in all these fast search algorithms, it is assumed that the error criteria monotonously decreases and reaches the minima as the search point (candidate block) moves towards the global minimum. In fact, there are chances that a local minima may be mistaken as a global minima.

In our algorithm, this problem is rectified by choosing the candidate blocks as shown in Fig. 1. During coarse search, error is calculated for 77 selected blocks and the block with minimum error is determined, which will be the nearest block to the global minima. In fine search, error is calculated for 8 blocks surrounding the chosen candidate block. Here, the block with the minimum error will be the global minimum point. Thus in this method, absolutely there are no chances of misinterpreting a local minima as global minima.

The real time processing of video signals requires tremendous computational capability that can only be achieved cost effectively by using VLSI (Huang *et al*., 2004; Yang *et al*., 1989). The usefulness of video algorithms strongly depends on the feasibility and effectiveness of its VLSI implementation. We propose an efficient VLSI architecture for CBPS algorithm. The core consists of a single processing element and an adder. According to the hardware oriented features of this algorithm, it is very easy to develop an application specific processor.

ALGORITHMS

Usually, the block matching process is performed only on the luminance frame. Each luminance frame is divided into blocks of size N×N and each block in the current frame is matched with candidate blocks of size N×N within the search area in the reference frame. The best matched block has the lowest SAD among all the selected candidate blocks. Instead of the original block,

Corresponding Author: Reeba Korah, Department of Electronics and Communication Engineering, Anna University, Chennai, India

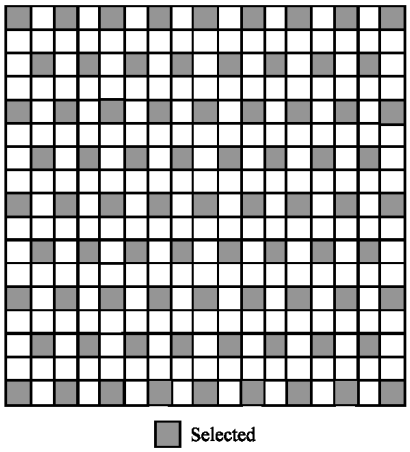

Fig. 1: Selection of candidate blocks

the displacement (motion vector) of the best matched block and the prediction residue will be transmitted to the decoder.

Full Search Block Matching Algorithm (FSBMA): FSBMA can be expressed with the following equations:

$$SAD(m,n) = \sum_{I=0}^{N-1}\sum_{j=0}^{N-1} R(i,j)-s(i+m,j+n)$$

$$MV = \{(u,v\ SAD(u,v) \leq SAD(m,n)\}$$

Where SAD(m,n) denotes the SAD at each search position (m,n). (m,n) is in the search range of {-p,p}. MV expresses the motion vector of current block with minimum SAD among $(2p+1)^2$ search positions.

Candidate Block and Pixel Subsampling (CBPS) algorithm: In FSBMA, $(2p+1)^2$ candidate blocks are to be searched to find the motion vector for a particular block. This is computationally very intensive process. In Candidate Block Subsampling (CBS), we have chosen candidate blocks in the following pattern.

Initially, with coarse search 5(2p+1)-8 blocks are searched and during fine search 8 more blocks are searched. In fast search algorithms, it is assumed that the error criteria monotonously decreases and reaches the minima as the search point (candidate block) moves towards the global minimum. But, actually, there are chances that a local minima may be mistaken as a global minima as shown in Fig. 2.

In the proposed method, this problem is rectified by choosing the candidate blocks as shown in Fig. 1. During coarse search, error is calculated for 77 selected blocks and determines the block with minimum error, which will be the nearest block to the global minima. Now, in fine search, error is calculated for 8 blocks

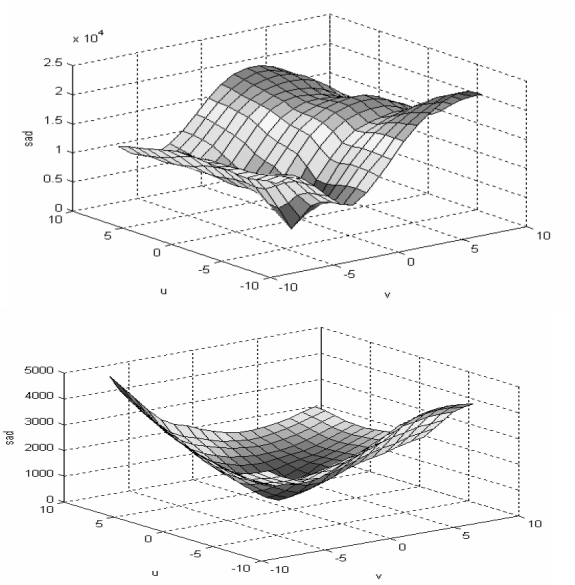

Fig. 2: Plot of SAD

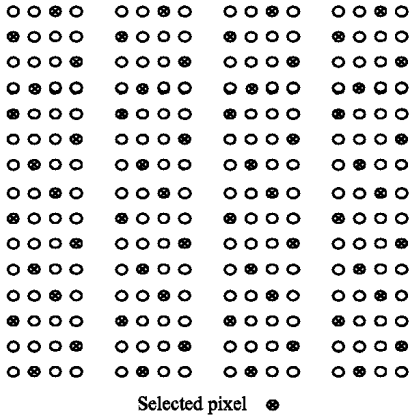

Fig. 3: 4-Queen pattern

surrounding the chosen candidate block. Here the block with minimum error will be the global minimum point. Thus, in this method, absolutely there are no chances of misinterpreting a local minima as global minima.

We can improve motion estimation by reducing the number of pixels from a block used for matching. The pixel decimation approaches can be easily combined with candidate block sub sampling approach. Our new 4-Queen pattern shown in Fig. 3 gives good performance.

CBPS Algorithm can be expressed with the following equations:

$$SSAD(m,n) = SSAD(m,n)+SM(i,j)*R(x*N+i,y*N+j)-S(x*N+i+u,y*N+j+v)$$

$$MV = \{(u,v)\,|\,SSAD(u,v) \leq SSAD(m,n)\}$$

Table 1: Comparison of PSNR (a) Garden sequence, (b) Telephone sequence

Frame No.	FSBMA	Proposed method
Garden sequence		
41	26.8125	26.5819
42	28.6343	28.4614
43	28.2480	28.1143
44	26.7367	26.5833
45	28.8699	28.6704
46	29.1844	28.9217
47	27.4993	27.2572
48	27.2974	27.1778
49	26.9754	26.8651
50	30.1596	30.1414
Telephone sequence		
131	28.8588	28.7414
132	32.8974	32.6701
133	30.9867	30.7276
134	31.1862	31.0247
135	30.3926	30.1068
136	32.3542	32.0322
137	32.3068	31.9918
138	33.4527	33.2789
139	32.5606	32.5129
140	31.8932	31.4826

Table 2: Comparison of PSNR and MSE for different RBSSAD for frame No. 135 of Telephone sequence

Error criterion	No. of bits truncated/pixel	MSE	PSNR
SSAD	0	17.9895	30.1413
RBSSAD7	1	17.9345	30.1963
RBSSAD6	2	18.0353	30.0955
RBSSAD5	3	18.3458	29.7850
RBSSAD4	4	19.0302	29.1006

Instead of SSAD, RBSSAD also is experimented where, in place of representing a pixel with 8 bits, the least significant bits are truncated so as to obtain RBSSAD7, RBSSAD6, RBSSAD5, i.e., truncating 1 LSB, 2 LSB and 3 LSB, respectively. The results are shown in Table 1 and 2.

ARCHITECTURE DESIGN

The proposed motion estimation architecture using our algorithm is shown in Fig. 4. The architecture consists of two parts, namely, data accessing and pipelined matching modules. We have chosen N = 16 and p = 8 to explain our architecture

Data accessing: This part consists of an address generator as shown in Fig. 5 and a control section. For our pixel sub-sampling technique, 64 pixels (16×4) are chosen from 16×16 pixels. i.e., one block.

Pixel subsampling index j (k) is calculated as follows. For each row (i) of selected macro block, four pixels are selected with initializing the

Column index j (k) = 1,3,0,2 for k = 0, 1,2, 3 where i = 0 to 15

$$j(k) = j(k) + 4 ; \quad k = i \bmod 4$$

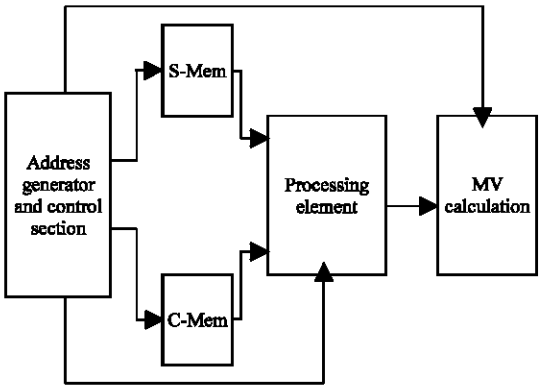

Fig. 4: Proposed CBPS architecture

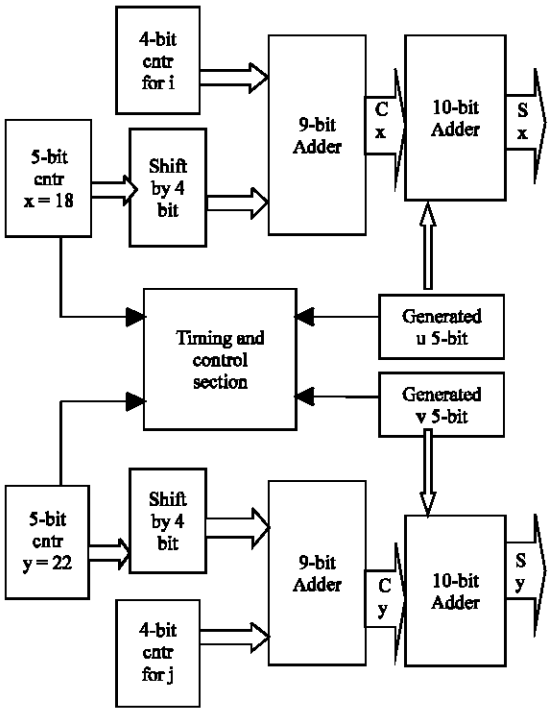

Fig. 5: Structure of address generator

For CIF frame (288×352), with a block size of 16×16, the total number of blocks = 18×22 = 396
Pixel co-ordinates of current frame are calculated as

$$C_x = N*x + i$$
$$C_y = N*y + j$$

Pixel co-ordinates of previous frame are calculated as

$$S_x = N*x + i + u$$
$$S_y = N*y + j + v$$

Where < u, v > = -p to p

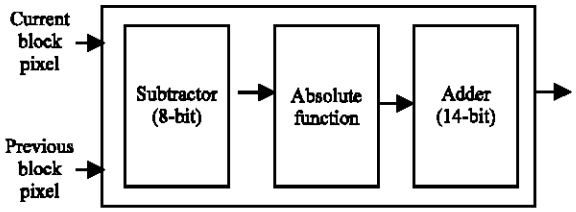

Fig. 6: Structure of processing element

Processing element: The novelty of our architecture is the use of a single processing element shown in Fig. 6. Thus the area requirement is reduced manifold. PE consists of 8-bit subtractor, absolute function and a 14-bit adder for adding absolute difference values. SSAD or RBSSAD calculated thus is given to the motion vector calculation block.

The advantages of using RBSSAD criteria are:

Savings in VLSI area: In RBSSAD, pixel comparison is a *k* bits subtractor which is implemented by a k bits adder in VLSI. The comparison requires O(k) bits hardware with a trivial adder. In this case, if the k is reduced to a maximum of 62.5%, the comparison hardware is also reduced to 62.5%.

Speedup in VLSI operation: The comparison is naturally a bit-serial operation. So, if k is reduced, the hardware runs faster.

In addition to the above advantages, the RBSSAD is suitable for current FPGA implementation, Many FPGAs have large numbers of configurable logic blocks (CLB). A CLB has small number of inputs and outputs. Owing to this limited I/O in CLB, the RBSSAD can be more effectively implemented than the SSAD.

RESULTS

The proposed CBPS algorithm and its architecture are developed mainly for video phone applications which demands estimation of very small motions. The algorithm works well for medium and high motion also. We have verified the functional credibility of the algorithm and architecture by VHDL simulation using 288×352 CIF frames. Performance of the proposed architecture has been compared with FSBMA and other popular architectures. The fascinating feature of our architecture is the use of a single processing element to maintain 30 f/s with an operating frequency of 65 MHz. With a single processing element, 5443 clock cycles are required per block. RTL schematics of the architecture are shown in Fig. 7a-c.

Synthesis report

Device utilization summary:

Selected Device : SPARTAN -II E (2s300epq208-6)

Number of Slices	: 226 out of 3072	7%
Number of Slice Flip Flop	: 249 out of 6144	4%
Number of 4 input LUTs	: 400 out of 6144	6%
Number of bonded IOBs	: 66 out of 146	45%
Number of GCLKs	: 1 out of 4	25%

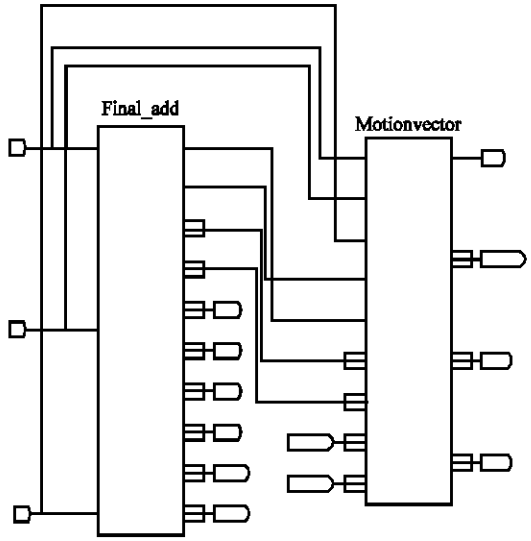

Fig. 7a: Schematic diagram for CBPS architecture

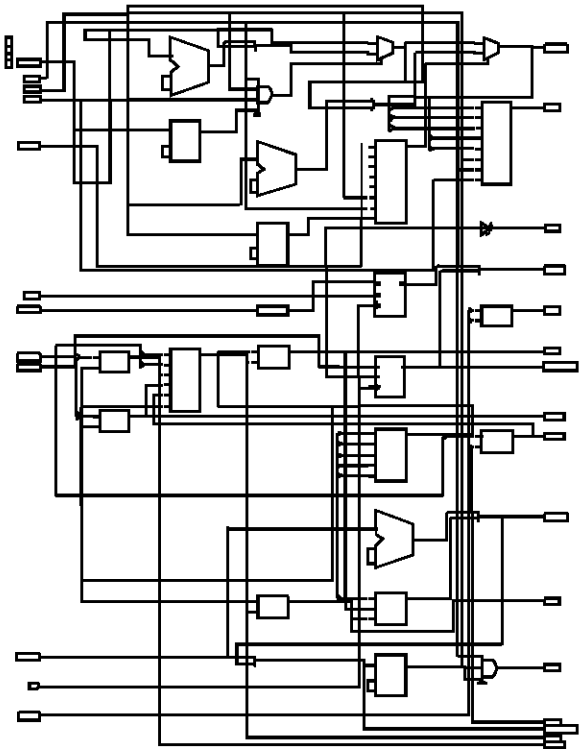

Fig. 7b: Schematic diagram for address generation and control

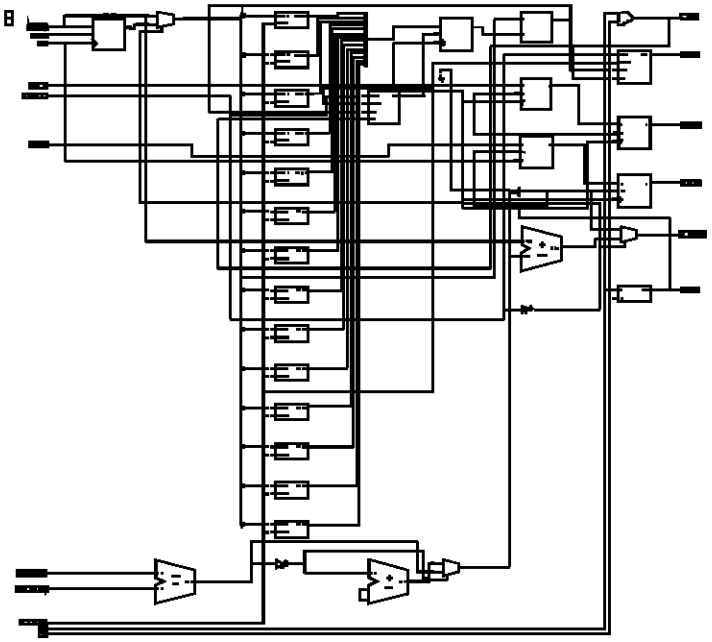

Fig. 7c: Schematic diagram for motion vector generator

DISCUSSION

Candidate block and pixel sub-sampling algorithm gives an appropriate blend of performance of FSBMA and other fast search algorithms. Due to the peculiar selection of candidate blocks, any chance of a wrong motion vector calculation is totally eliminated. The hardware complexity is further reduced by the new error criterion RBSSAD. The algorithm works well for all kinds of motion. VLSI architecture developed for this algorithm works with a single processing element in spite of maintaining the standard frame rate at nominal clock frequency. We are further in the process of enhancing the speed by incorporating pipelining and parallel processing.

REFERENCES

Advanced Video Coding, 2003. ITU-T Recommendation H.264, MPEG-4, Part 10.

Chen, M.J., L.G. Chen, T.D. Chiueh and Y.P. Lee, 1995. A new block-matching criterion for motion estimation and its implementation. IEEE Trans. Circ. Syst. Video Technol., 5: 231-236.

Huang, Y.W., S.Y. Chien, B.Y. Hsieh and L.G. Chen, 2004. Global elimination algorithm and architecture design for fast block matching motion estimation. IEEE Trans. Circ. Syst. Video Technol., 14: 898-907.

Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 M bits/s, 1993. Part II: Video, ISO/IEC, 11: 172-172.

Information Technology-Generic Coding of Moving Pictures and Associated Audio Information, 1996. Video. ISO/IEC 13 818-2, ITU-T Recommendation H.262.

Information Technology-Coding of Audio Visual Objects, 1999. Part II: Visual. ISO/IEC., 14: 492-496.

Jung, S.T. and S.S. Lee, 2004. A 4-way pipelined processing architecture for three-step search block-matching motion estimation. IEEE Trans. Consumer Electronics, 50: 674-681.

Po, L.M. and W.C. Ma, 1996. A novel four-step search algorithm for fast block motion estimation. IEEE Trans. Circ. Syst. Video Technol., 6: 313-317.

Seth, K., P. Rangarajan, S. Srinivasan, V. Kamakoti and V. Balakuteshwar, 2004. A parallel architectural implementation of the new three-step search algorithm for block motion estimation. Proc. 17th Intl. Conf. VLSI Design (VLSID '04).

Video Codec for Audio Visual Services at px64 Kbits/s, 1993. ITU-T Recommendation H.261.

Video Coding for Low Bit Rate Communication, 1998. ITU-T Recommendation, H.263.

Yang, K.M., M.T. Sun and L. Wu, 1989. A family of VLSI designs for the motion compensation block-matching algorithm. IEEE Trans. Circ. Syst., 36: 1317-1325.