MultiFrame Fast Search Motion Estimation and VLSI Architecture

Dr.D.Jackuline Moni [1] K.Priyadarshini [2]

[1] Karunya University , Dept.Of ECE,Coimbatore
[2] Karunya University ECE, Coimbatore, India

Abstract- This paper presents a fast Motion Estimation algorithm concept with reduction in execution time, which provides low power consumption in the design of hardware architecture. A new VLSI architecture for Integer Motion Estimation based on the Full search algorithm is proposed.This architecture uses Sum Of Absolute Difference (SAD) operation,the commonly used metric to determine the best match of the blocks. Computations and comparisons are performed using Carry Save Ahead Adder (CSA) and Carry Look Ahead Adder(CLA) for SAD operation. The design has been implemented on the Xilinx Spartran which depicts the characteristics of memory used , and power consumption. Fast search algorithm reduces search area by determining motion vectors for multiframes instead of single frame. The Motion Estimation algorithm is the most computationally intensive part of the encoder which is simulated using MATLAB**.**The proposed algorithm gives us a fast processing time schedule with a fixed number of computations. Multiframe motion estimation speeds up the computation time and the simulation results shows that PSNR value is high compared with the single frame process and computation time is reduced with negligible performance degradation.

Index Terms- Motion Estimation,Fast search algorithm, SAD Hardware Architecture.

I. INTRODUCTION

Motion estimation and motion compensation are the predictive techniques for exploiting the temporal redundancy between successive frames of video sequence. For encoding , Block matching motion estimation takes a maximum part of the processing time. An MPEG video, is represented as a sequence of frames and the small differences existing between the frames are realized by the process of motion vector. This best motion vector is obtained by full-search block matching algorithm and various fast searching algorithms. Full search block matching algorithm provides the optimal solution by exhaustively comparing each (NXN)block of the current frame with all the candidate blocks of $(N+2p)^2$ search window. In video coding system, Motion estimation (ME) plays a key role to improve coding efficiency by reducing temporal redundancy within video sequence. ME takes more than 50% computational complexity in H.264/MPEG-4 AVC [1] which is the state-of-the-art video compression standard. Block matching algorithm, which is ME algorithm used in H.264/MPEG-4 AVC, is based on dividing a frame into MacroBlocks (MBs)(16x16 pixels), and searches for the best matching block with current MB within search area in a reference frame. ME process requires very high computational complexity, and it has been implemented as a dedicated hardware with reduced memory area and low power consumption. Several search techniques like Full Search[2] Fast Full Search[3] and Fast Search are present. Full search algorithm matches all possible candidates with a minimal distortion to find the displacement. ME refines the best candidate for each sub blocks using two stages Integer Motion Estimation and Fractional Motion Estimaton .[4] During the first stage 41 motion vectors are determined using various searches. In IME, to determine the Motion Vector(MV), the integer pixel searches the best matching integer position using performance cost metric Sum of Absolute Difference (SAD), Mean Square Error (MSE) and Mean Absolute Difference (MAD) which gives a numerical data about how similar two blocks are.

.

$$MAD=1/N^2 \sum_{i=0}^{N} \sum_{i=0}^{N} |I(k+i,l+j) - I(k+i+m,l+j+n) \quad (1)$$

MAD cost function's simplicity and ease of implementation in hardware tends to overemphasize small differences giving an inferior result to MSE.

$$MSE=1/N^2 \sum_{i=0}^{N} \sum_{i=0}^{N} |I(k+i,l+j) - I(k+i+m,l+j+n)^2 \quad (2)$$

$$SAD= \sum_{K=0}^{M-1} \sum_{l=0}^{N-1} |C(k,l)-R(k+l,l+j) \quad (3)$$

In equations (1)&(2) I(k+i,l+j)and I(K+i+m,l+j+n)
are the pixels of reference and current frames respectively. (j,n) represents the displacement of the candidate frame within the search window, In equation(3) C(k,l) & R(k+l,l+j)are the pixels of reference and current frame.

In algorithms [4] [5] center position of next searching step is not known until the minimum search step is found which leads critical issues in latency and pipelining cycles.[6] Full search motion estimation algorithms follow a regular search pattern and usually provides superior visual quality in terms of PSNR compared to other motion estimation algorithms.[7] PSNR measures the difference between two images and gives a result in a figure of decibels. Higher values of PSNR results in small differences between two images [8].

In this paper, motion vectors for a single reference frame and multiframes are determined for sample sequences and a comparative analysis is done based on PSNR value and computation time for the cost metrics MSE and MAD .The rest of the paper is organized as follows: Section 2 deals with the features of PSNR values of multi frame motion estimation. Section 3 deals with the architecture design of SAD using CSA and CLA. Section 4 figures out the simulation results with the comparison chart. Section 5 concludes the paper.

II. FRAME MOTION ESTIMATION

Fast block matching algorithms Three Step Search (TSS), New Three Step Search (NTSS), Four Step Search (FSS,) Diamond Search (DS), Adaptive Root Pattern Search (ARPS) reduces up the number of computations by decreasing the number of search points which leads to poor PSNR computations. Multiframe motion estimation is performed with reduced computation time and with minimum degradation in PSNR value of the motion compensated frame.

Table I represents the PSNR value and the Computation Time of Single Frame Processing for various video sequences using MSE

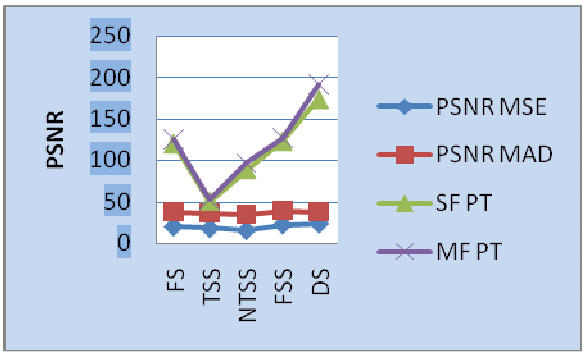
Video Sequence	Full Search(FS)		Three Step Search(TSS)		New Three Step Search(NTSS)		Four Step Search(FSS)		Diamond Search(DS)	
	PSNR	Computation Time	PSNR	Computation Time	PSNR	ComputationTime	PSNR	Computation Time	PSNR	Computation Time
Akiyo_QCIF(518X350) @15 fps [2x2]	34.18	87.71	31.72	37.22	26.90	70.02	33.46	101.89	32.94	67.68
Car (320x240)@ 50fps[2x2]	23.8	122.0	20.9	50.57	17.3	90.3	26.6	113.58	23.51	178.3
Vipmen (160X120)@ 100fps[2x2]	27.80	30.93	26.12	13.6	22.92	22.21	31.18	24.02	27.75	37.37

Table II represents the PSNR value and the Computation Time of the proposed MultiFrame Processing for various video sequences using MSE

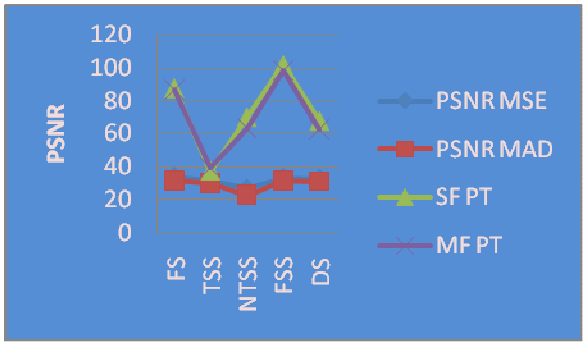
Video Sequence	Full Search(FS) (MultiFrames)		Three Step Search(TSS) (MultiFrames)		New Three Step Search(NTSS) (MultiFrames)		Four Step Search(FSS) (MultiFrames)		Diamond Search(DS) (MultiFrames)	
	PSNR	Computation Time	PSNR	ComputationTime	PSNR	Computation Time	PSNR	Computation Time	PSNR	Computation Time
Akiyo_QCIF(518X350) @15 fps [2x2]	31.38	86.6	29.65	38.96	22.44	63.08	31.40	98.46	30.93	62.36
Car (320x240)@ 50fps[2x2]	19.97	120.03	17.9	50.06	15.45	88.9	22.4	122.86	23.51	173.8
Vipmen (160X120)@ 100fps[2x2]	21.62	28.67	21.69	13.5	18.50	22	27.51	25.41	23.8	37.00

Table III represents the PSNR value and the Computation Time of the proposed MultiFrame Processing for various video sequences using MAD

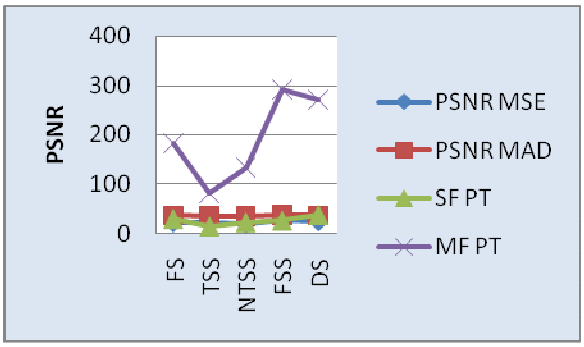
Video Sequence	Full Search(FS) (MultiFrames)		Three Step Search(TSS) (MultiFrames)		New Three Step Search(NTSS) (MultiFrames)		Four Step Search(FSS) (MultiFrames)		Diamond Search(DS) (MultiFrames)	
	PSNR	Computation Time	PSNR	ComputationTime	PSNR	Computation Time	PSNR	Computation Time	PSNR	Computation Time
Akiyo_QCIF(518X350) @15 fps [2x2]	51.20	88.95	48.8	39.46	44.44	69.65	51.16	103.38	50.3	169.4
Car (320x240)@ 50fps[2x2]	37.76	124.95	36.36	52.19	34.54	96.93	39.33	126.93	37.62	192.13
Vipmen (160X120)@ 100fps[2x2]	37.10	183.00	36.1	81.60	35.04	134.14	38.71	292.57	37.05	270.84



(a) Akiyo Video Sequence



(b) Car Video Sequence



(c) vipmen video sequence

Figure I(a)-(c) shows the PSNR value of MSE and MAD for the Akiyo video sequence. The Processing Time of Multiple Frame is minimum with slight degradation in the PSNR value. PSNR value of MAD gives best results when compared with MSE for Multiple Frames

III. VLSI ARCHITECTURE

The main characteristic of the proposed architecture is the large amount of Processing Element Units(PE's)that are used to calculate SAD (Sum Of Absolute Difference) metric. [8] The internal structures of the PE is composed by a large number of addition to calculate the SAD's. In this paper,SAD operation is very time consuming due to the complex nature of the absolute operation. For different search algorithms different address generation schemes are used. The resulting SAD's and motion vectors are stored in a small memory accessible through an external bus which acts as a communication bridge.[9]The Motion Estimation unit consists of Search Area (SA) memory, a processing array which contains 256 Processing Elements, Adder Tree, comparator and Decision unit. The data in the memory are stored in a ladder like manner to avoid delay. The 256 absolute differences are summed up by the adder tree and the outputs are the 41 block partitions. [10] Two types of adders CSA and CLA are processed and compared. The comparator updates the minimum SAD throughout the scanning process for the 41 block partition.

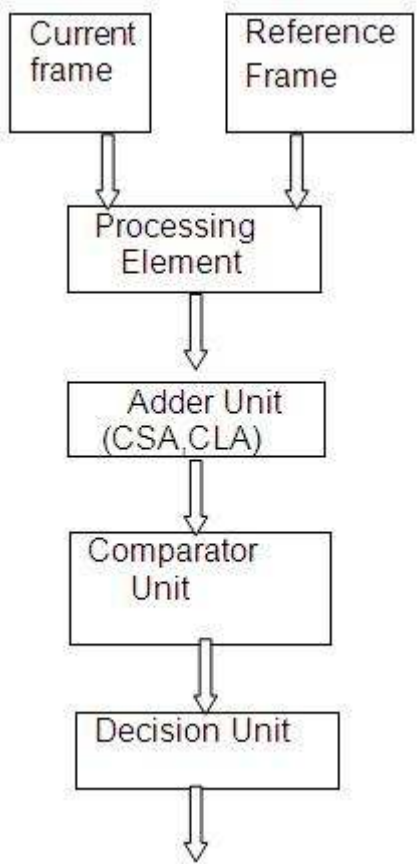


Figure 2: Generation of Motion Vectors

The proposed PE Units are proposed to calculate the SAD with reduced computation complexity and delay. The ME unit requires 256 clock cycles for each MB search. The input and output for the adder tress is 8 bits wide and the SAD output is 4 and 16 bits wide.These data are then input into the comparator, together with the current search location information.

Adder Unit:

The Carry Look Ahead [11] improves speed by reducing the amount of time required to determine carry bits. Carry Look Ahead logic uses the concepts of generating (G) and propagating (P) carries. The Carry-Look Ahead adder is broken up in two modules. Carry-Save Adders (CSA) is useful in situations when we need to add more than two numbers, since this design automatically avoids the delay in the carry–out bits. The proposed architecture is designed with CLA and CSA. Table IV results shows that replacement of CSA with CLA decreases the propagation delay and power consumption.

Table IV: Synthesis Report of Adder and SAD circuits

S.No	Parameters	Adder Unit		Motion Estimation Unit	
		CSA (8 bit)	Proposed CLA (16 bit)	SAD Unit (8 bit)	Comparator Unit (8 bit)
1.	Path Delay	21.831 ns	6.261 ns	6.236 ns	8.61 ns
2.	Memory Usage	185300 KB	156236 KB	160404 KB	160404 KB
3 .	Power Consumption	247 mw	227 mw	262 mw	167 mw

IV. CONCLUSION

In this paper an important coding tool of H.264 is the Motion Estimation matching algorithm. The motion estimation process from the current frame and the reference frame is simulated using Matlab software.The SAD architecture with the Adder units is implemented using Xilinx Spartan 3E FPGA. Five different algorithms for motion estimation are simulated and PSNR value for MSE and MAD values are determined. It is concluded that multiframe processing PSNR value of MAD gives best results when compared with MSE value with increase in processing time.

APPENDIX

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the

singular heading even if you have many acknowledgments.

REFERENCES

[1] J. V. Team, Draft ITU-T Rec. and Final Draft Int. Standard of Joint Video Spec. ITU-T Rec. H.264 and ISO/IEC 14496- 10 AVC, May 2003.

[2] I.Richardson, H.264 and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia. John Wiley&Sons, Chichester, 2003.

[3] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp.148–157, Apr. 1993.

[4] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast blockmatching motion estimation," IEEE Trans. Image Process., vol. 9, no. 2, pp. 287–290, Feb. 2000.

[5] M. Porto, L. Agostini, S. Bampi, A. Susin, "A High Throughput and Low Cost Diamond Search Architecture for HDTV Motion Estimation",in IEEE International Conference on Multimedia & Expo, 2008, pp.1033- 1036.

[6] M.-J. Chen, G.-L. Li, Y.-Y. Chiang and c.-T. Hsu, "Fast Multiframe Motion Estimation Algorithms by Motion Vector Composition for the MPEG-4/AVC/H.264 Standard," IEEE Trans. Multimedia, vol. 8, n. 3,Jun. 2006, pp. 478 - 487.

[7] S.-S. Lin, P.-C. Tseng, and L.-G. Chen, "Low-power parallel tree architecture for full search block-matching motion estimation," in *Proc. IEEE International Symposium on Circuits and Systems*, May 2004, pp. 313–316.

[8] J.-C. Tuan, T.-S. Chang, and C.-W.Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 61–72, Jan. 2002.

[9] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture," in *Proc IEEE Int. Symp. Circuits Syst.*, 2004, pp. 273–276.

[10] S. Vassiliadis, E. Hakkennes, S. Wong, and G. Pechanek. The Sum-Absolute-Difference Motion Estimation Accelerator. In *Proceedings of the 24th Euromicro Conference*, 2000.

[11] Abdellatif Bellaouar and Mohamed I.Elmasry, Low power Digital VLSI Design, Kluwer Academic Publishers, Norwell, MA, 1995.

AUTHORS

First Author – Dr.D.Jackuline Moni, Karunya University , Dept.Of ECE,Coimbatore, Email: jackuline_2000@yahoo.com

Second Author – K.Priyadarshini, Karunya University ECE, Email: priyanilesh@rediffmail.com