International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 02, February 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

FLEXIBLE MOTION ESTIMATION PROCESSOR WITH LOW LATENCY FOR REAL TIME VIDEO COMPRESSION TECHNIQUES

G.Sandhiya[1], K.Dhivya[2], V.Kannagi[3], P.Arthi Devarani[4], G.Parvathy[5]

[1,2,3,4,5] *Department of Electronics and Communication,*
R.M.K College of Engineering and Technology, Chennai (India)

ABSTRACT

An innovative adaptive architecture that is capable of estimating variable motion vectors in accordance to the motion between reference frame and current frame has been developed. The amount of motion occurring in the frames is not equal throughout the video. A reasonable amount of compression can be obtained by choosing a different algorithm to estimate motion vector. Selection of an algorithm from the three algorithms stated in the hybrid architecture is made based on the deviation between the reference frame and the current frame. A new algorithm called cross diamond search algorithm is used that requires less search points compared to the diamond search algorithm and three step search algorithm. This algorithm helps in bandwidth conservation by dipping temporal redundancy and also reduces power consumption by decreasing computational complexity. The performance of the algorithm is compared with other algorithms in terms of number of search points. In the existing system, the frames were stored in external memory that leads to increased delay while accessing the frames. Furthermore the area of the device also increases leading to more power dissipation. In the proposed system, BRAM within the chip is used to store the frames thereby resulting in reduced delay, area and power dissipation.

Index Terms—Hybrid Architecture, Motion Vector, Cross Diamond Search Algorithm, BRAM

I INTRODUCTION

Transmission of video has been increasing tremendously in the last decade with the evolution of digital technology. But transmission of these videos through a medium consumes more bandwidth. For example, a video is the sequence of 2 dimensional image or frames. A video of 1 second consist of 30 frames and each frame is 256* 256 pixels size on an average. The memory requirement of each pixel is 16 bits. Therefore memory requirement of each frame is around 1 MB bits. Hence a bandwidth of 1 MB is expended to transmit a video of 1 second. With the above calculation, if a video of 3 hours duration is to be transmitted it would consume a bandwidth of around 10 GB. It is evident that when a video is transmitted without compression, it consumes huge bandwidth. The bandwidth available is very limited and it becomes mandatory to compress the video before transmission. Many video

ISSN (O) 2454-9665
ISSN (P) 2454-0663

compression techniques have been developed in the last decade that involves extracting the motion vector from the frames and then transmitting these vectors along with the reference frame. At the receiver, the video is reconstructed with this information.

Motion estimation is the process of obtaining motion vectors by comparing the reference frame and the current frame. Figure 1.1 shows the block diagram for motion estimation. The motion estimation section will generate motion vectors for the current frame by comparing with the reference frame. The model will contain the motion vector of the current frame. This motion vector is used for motion compensation and the compensated frame is then encoded and transmitted.

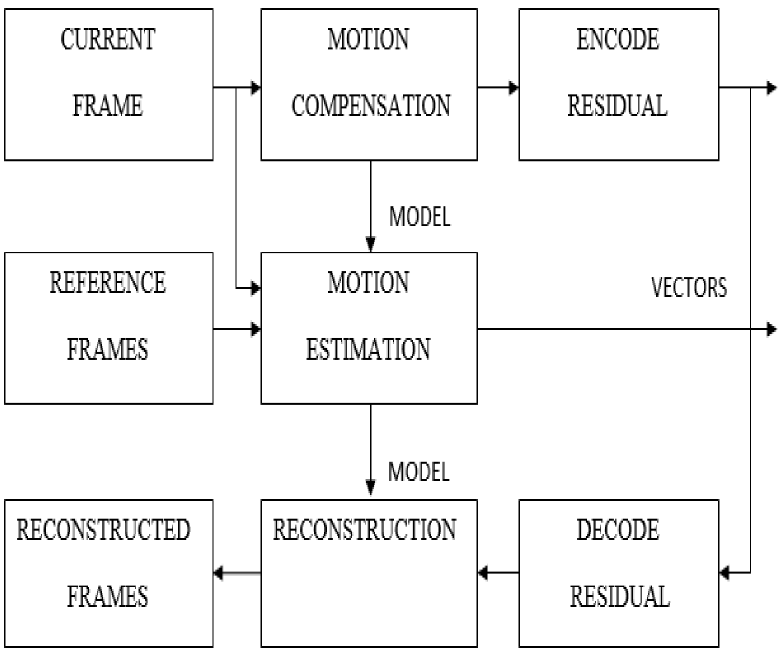


Figure 1.1 Motion estimation block diagram

II EXISTING SYSTEM

In the existing system the motion vectors are generated with the help of a single motion estimation algorithm for all the frames in the video. The various motion estimation algorithm are three step search, diamond search, cross diamond search algorithm etc.. This motion estimation algorithm has its unique properties. In our considered existing system, the video is converted into frames. Then edge is detected for all the frames discretely. The edge detection will give the outline of the objects in a frame. The motion vectors are obtained by comparing the displacement occurred in these edge detected frames. The motion estimation algorithm which is used for generating motion vectors are three step search algorithm (TSS), diamond search algorithm (DS) and cross diamond search algorithm (CDS). The three step search algorithm is used if significant motion is occurred. The search cycle is completed in a three step. So it is speeder than the other mentioned algorithm. The diamond search algorithm has more search points than cross diamond search algorithm. Diamond search is used when small motion is occurred over a large distance. Cross diamond search is also used when gentle and small motion occurred over a large distance at the same time has less search points. These algorithm are so far used separately. In order to combine the benefits of the three algorithm we construct a hybrid architecture which can combine these algorithm according to the requirement. So the compression achieved is also more.

In the existing system, the frames are stored in the external memory. The delay increases while accessing the frames from the external memory. The area of the device is also more due to the addition of external memory. It leads to more power dissipation.

III PROPOSED SYSTEM

In the proposed system, we develop a hybrid architecture which utilizes the aforementioned three algorithm in a single structure adaptively. Any one of these algorithm is chosen based on the standard deviation between the reference frame and the current frame. If the standard deviation is more, it means the motion is significant. So the three step search algorithm is used to estimate motion vectors. In the other case, the motion is small attributed to the large distance. So diamond search or cross diamond search is used according to the search point consideration.

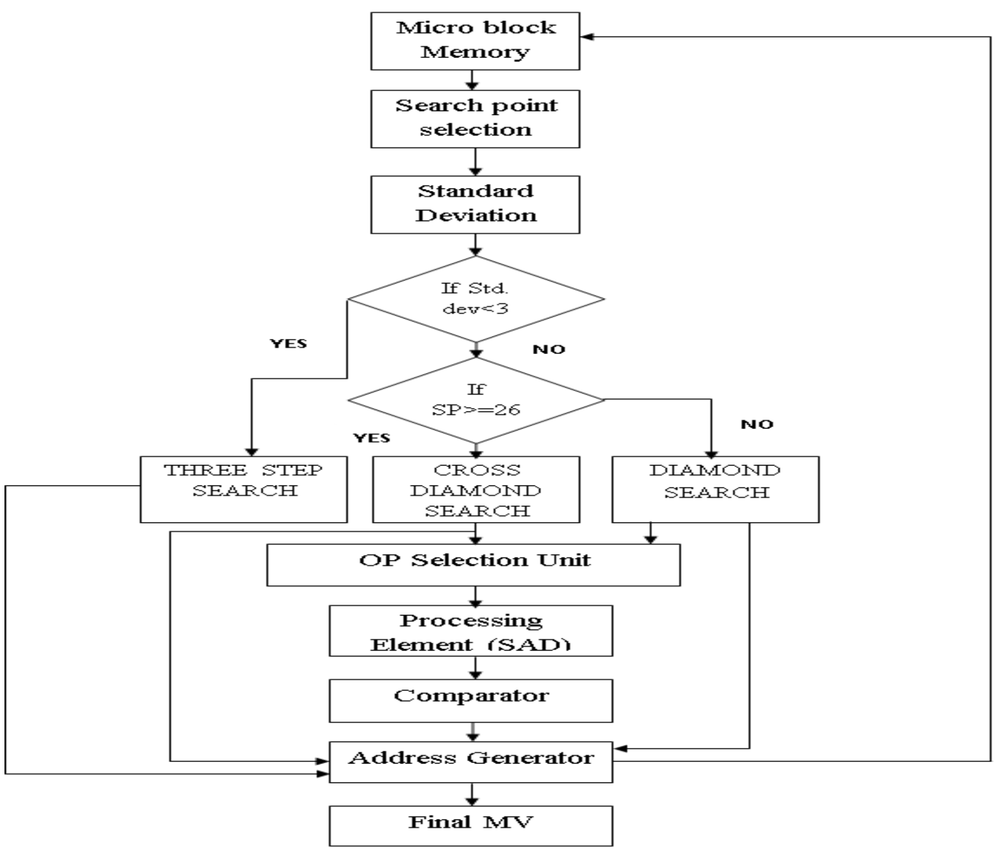


Figure 3.1: Block diagram of hybrid architecture

The block are compared from the micro block memory and standard deviation among them is calculated. According to the standard deviation any one of these three algorithm is chosen and executed to obtain motion vectors. The algorithm generate the motion vector by calculating the sum of absolute difference (SAD) and chooses the minimum point which will act as the center point in the search with the help of a comparator. The new search pattern are loaded with the help of an address generator from the block memory.

In order to reduce the delay in proposed system, the BRAM (Block RAM) within the chip is reconfigured to store the frames. The area of the device is also reduced due to the absence of external memory. So power dissipation is less.

IV FAST SEARCH ALGORITHM USED

In order to reduce the complexity involved in full search algorithm, various fast search algorithm such as TSS, DS, CDS are developed. The difference among them is on the search pattern used. The speed and accuracy of the algorithm also depends upon the pattern of the algorithm.

4.1 Three Step Search Algorithm (TSS)

TSS is used if the motion occurred is significant and active. It is simple, robust and faster than both DS and CDS algorithm. The following step explains the TSS algorithm:

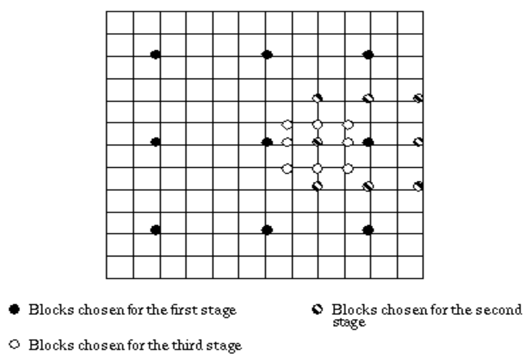


Figure 4.1.1: Three step search algorithm

Step 1: first the step size between the blocks are chosen. Eight blocks are selected for comparison.

Step 2: from the above blocks, a minimum block distortion (BDM) is obtained and it becomes the center of new search cycle. Step size is reduced to half the value. The new BDM is used as center of the new search cycle with half step size.

Step 3: the step is repeated until the reduced until the step size is 1.

International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 02, February 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

4.2 Diamond Search Algorithm

The diamond search algorithm provides an enhanced performance than the TSS algorithm. The DS algorithm uses a diamond shape pattern with 9 search points out of which 4 points located at the corners and another 4 points located at the center of the edges in diamond shape. The diamond search algorithm is used for low bit rate and to find the small motion. It has two search pattern. They are Large Diamond Search Pattern (LDSP) and a Small Diamond Search Pattern (SDSP).

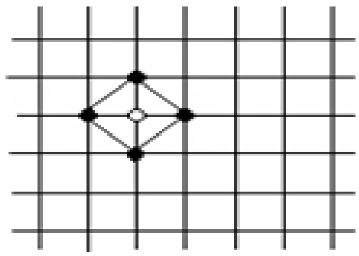
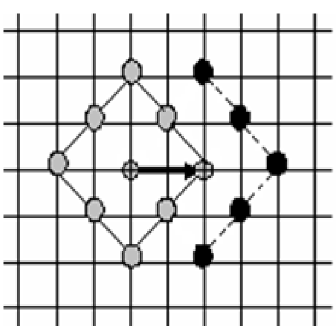
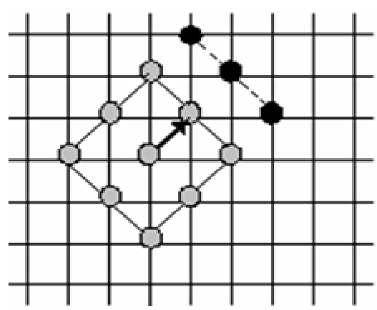


Figure 4.2.1 SDSP

First LDSP having nine search points is executed. A Minimum Block Distortion (MBD) is chosen among the search points and it act as the new center for the next LDSP. If the MBD is at the edge or corner of the LDSP, then some of the search points in new and old LDSP are overlapping. So only the non-overlapping search points can be considered. It reduces the number of search points in the next LDSP. If MBD is at edge the only three search are evaluated. If MBD is at corner five search points are evaluated. If the MBD is at the center of LDSP then SDSP is applied with new MBD as center. The MBD at different positions are explained.





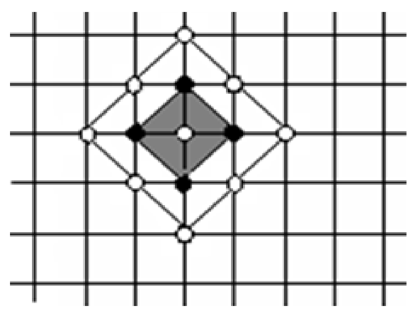


Fig 4.2.2: MBD point is at corner **Fig 4.2.3: MBD point is at edge** **Fig 4.2.4: MBD point is at center**

The MBD points thus obtained will give the motion vector.

4.3 Cross Diamond Search

Cross diamond search algorithm uses less search point than three step search and diamond search. Cross diamond search is shown below in figure

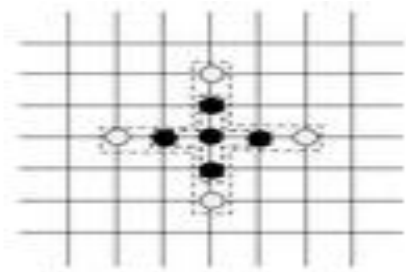


Figure 4.3.1: Cross Diamond Search

4.3.1 Search Patterns

The search pattern is divided in 2 different shapes: small cross-shaped pattern (SCSP) and the large cross-shaped pattern (LCSP).

Step 1 (Starting – Small Cross Shape Pattern SCSP): with origin as center, a SCSP is formed. A minimum block distortion BDM is found from the 5 search points of SCSP. If the BDM occurs at the center of SCSP then the search stops otherwise go to step 2.

Step 2: (SCSP): with BDM as center, a new SCSP is formed. From this a new BDM is found out. If the new BDM is at the center of new SCSP then the search stops otherwise go to step 3.

Step 3: the three remaining search points of LCSP are checked to guide the possible correct direction for the next step. And then go to step 4.

Step 4: A new Large Diamond Search Pattern LDSP is formed from the BDM found in previous LDSP step as center. If the new BDM is at the center, then go to step 5. Otherwise repeat this step.

Step 5: with minimum BDM point from before step as center a SDSP is formed. The new BDM from this SDSP will give the motion vector.

V BLOCK MOTION MATCHING

Motion estimation is the computationally intensive procedure in video coding. It has an effect on the speed of the video coding. If the motion estimation search area is large then the motion estimation becomes even more complex. So to reduce the complexity, the frames is divided into small equal blocks. These blocks are compared and block distortion is found by computing the sum of absolute difference (SAD). Minimum point is chosen from these SAD value which will give the motion vector. The SAD is estimated by the below formula:

$$\text{SAD}(V_i) = \sum_{x=0}^{M}\sum_{y=0}^{N} | S_l(x,y) - S_{l-1}(x+dx, y+dy) |$$

Where

M*N = block size

$S_l(x, y)$ = pixel value of frame l at point x, y.

$[V_i = (dx, dy)]$ = displacement vector.

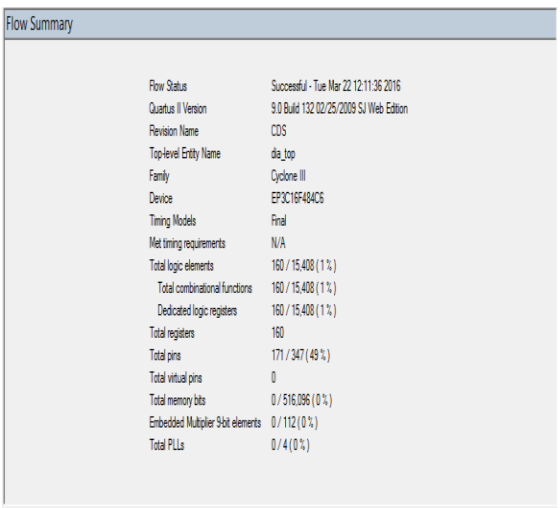
VI RESULT

6.1 SPEED (Fmax summary)

Delay in a system mainly depends upon the accessing time taken in the device. Here the delay can be found out using the Fmax summary. In existing system, frames are stored in external memory. So delay increases while accessing the frames from external memory. Hence speed is reduced. In proposed system, frames are stored in BRAM within the chip without using external memory. So delay is reduced and speed is increased.

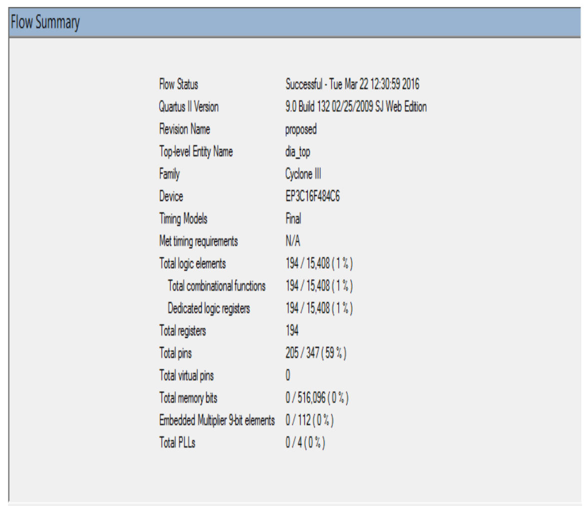
6.2 Area

Area of a device depends upon the total number of logical elements in the system. The proposed system has less number of logical elements than the existing system. So area of the device is more in existing than proposed one. The area of a device has significant effect on power dissipation in device.



Flow Summary

Flow Status	Successful - Tue Mar 22 12:11:36 2016
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	CDS
Top-level Entity Name	dia_top
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	160 / 15,408 (1 %)
Total combinational functions	160 / 15,408 (1 %)
Dedicated logic registers	160 / 15,408 (1 %)
Total registers	160
Total pins	171 / 347 (49 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	0 / 4 (0 %)



Flow Summary

Flow Status	Successful - Tue Mar 22 12:30:59 2016
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	proposed
Top-level Entity Name	dia_top
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	194 / 15,408 (1 %)
Total combinational functions	194 / 15,408 (1 %)
Dedicated logic registers	194 / 15,408 (1 %)
Total registers	194
Total pins	205 / 347 (59 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 6.2.1 Area of CDS algorithm **Figure 6.2.2 Area of adaptive architecture**

6.3 Power Dissipation

Many parameters affect the device power consumption and they are choice of process technology and supply voltage. Power consumption also depends upon the area of the device. A larger size device consumes more power than a smaller size device in the same family because of its larger number of transistor present. In existing system because of external memory usage, the area is more. So the power dissipation is more. In proposed system, BRAM is used rather than external memory. So the area is reduced and power dissipation is lowered.

ISSN (O) 2454-9665
ISSN (P) 2454-0663

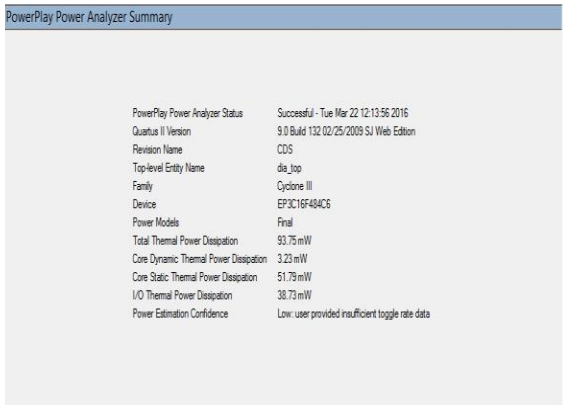


Figure 6.3.1 Power dissipation of CDS algorithm

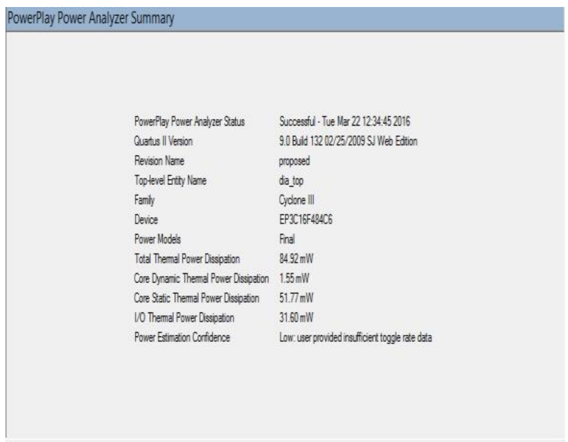


Figure 6.3.2 Power dissipation of adaptive architecture

OVER ALL COMPARISION

PERFORMANCE	EXISTING SYSTEM			PROPOSED SYSTEM
	CDS	TSS	DS	HYBRID ALGORITHM
AREA (NO OF LOGICAL ELEMENTS)	160	288	288	194
POWER DISSIPATION	93.75 mw	96.77 mW	104 mW	84.92 mW
DELAY	114.13 MHZ	261.73 MHZ	32.39 MHZ	114.25 MHZ

VII CONCLUSION

The hybrid architecture which is capable of estimating motion vectors is developed. The three step search (TSS), diamond search (DS) and cross diamond search algorithm are executed separately to find motion vectors. Their delay, area and power dissipation is calculated. The hybrid architecture delay, area and power dissipation is also calculated and compared with the above algorithms. It is very clear that usage of BRAM decreases the delay in the hybrid architecture. So the speed is high in hybrid structure than the individual algorithm. Even the compression ratio is increased since the cross search algorithm uses less search points.

A de-blocking filter is a video filter applied to blocks in decoded video to improve visual quality and prediction performance. They smoothen the sharp edges which occurs while block coding is done. The filter objective is to enhance the appearance of decoded frame. Decoded frames consist of corner artifacts that reduce the quality of an image. To avoid these we can design an Adaptive bilateral loop filter.

REFERENCES

1) Ce zhu, Xiao lin, Lap pui chau, Keng pang lim, Hock ann ang, Choo yin ong, “A novel hexagonal based search algorithm for fast block motion estimation, centre for signal processing,school of electrical & electronic engineering.
2) Chetan S.Dhamande, Prashant.A.Bhalge,(june 2013) “Overview of motion estimation in video compression,”International journals of scientific & engineering research, vol. 4.
3) Chi wai lam Lai man po and Chum ho cheung, “A new cross-diamond search algorithm for fast block matching motion estimation” Department of Electronic Engineering City University of Hong Kong, Hong Kong SAR.
4) Deumji Woo, Chae Eun Rhee, and Hyuk-Jae Lee,(February 2014) “A Cache-Aware Motion Estimation Organization for a Hardware-based H.264 Encoder,”IEEE Trans. Consumer Electron.,vol. 60, pp. 83-91.
5) Grzegorz Pastuszak and Mariusz Jakubowski,(May 2013) “Adaptive computationally scalable motion estimation for the hardware H.264/AVC encoder,”IEEE Trans. Consumer Electron.,vol.23.
6) Ishfaq Ahmad, Senior Member, IEEE, Weiguo Zheng, Member, IEEE, Jiancong Luo, Member, IEEE, and Ming Liou, Life Fellow, IEEE(march 2006) “A Fast Adaptive Motion Estimation Algorithm” IEEE transactions on circuits and systems for video technology, vol. 16, no. 3.
7) Jong nam kim Tae sun choi, “A fast three step search algorithm with minimum checking points using unimodal error surface assumption” IEEE Transaction vol.44 pp. 638-648.
8) K.Peter, “Algorithms,complexity analysis and VLSI architectures for MPEG-4 motion estimation”, Kluwer Academic Publishers, Boston, 1999.
9) K. B. Kim, Y. Jeon, and M.-C. Hong,(Aug 2008) “Variable Step Search Fast Motion Estimation for H. 264/AVC Video Coder,” IEEE Trans. Consumer Electron., vol. 54, no. 3, pp. 1281-1286.
10) L.C.Manikandan, Dr.R.K.Selvakumar, (July 2014) “A study on block matching algorithm for motion estimation in video coding,” International journals of scientific & engineering research, vol. 5.

11) Li Zhang and W. Gao,(May 2007) "Reusable Architecture and Complexity-Controllable Algorithm for the Integer/Fractional Motion Estimation of H.264," IEEE Trans. Consumer Electron., vol. 53, no. 2, pp. 749-756.
12) "Phase correlation motion estimation" report available at http://visilab.unime.it/~ianni/slides_CV/liang_report-phase-correlation-motion-prediction.pdf
13) T. Huang, Y. Hsu and R. Tsai "Interframe coding with general two-dimensional motion compensation", IEEE International Conference on ICASSP '82 (Acoustics, Speech and Signal Processing), vol. 7, pp. 464 - 466, May 1982.
14) T. C. Chen, S. Y. Chien, Y. W. Huang, C. H. Tsai, C. Y. Chen, T. W. Chen, and L.-G. Chen,(June 2006) "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder," IEEE Trans. Circuit Syst. Video Technol., vol. 16, no. 6, pp. 673-688.
15) Yi luo, "Fast Adaptive Block Based Motion Estimation for Video Compression", Department of Electrical Engineering and Computer Science and the Russ College of Engineering.
16) Y. K. Lin, C. C. Lin, T. Y. Kuo, and T. S. Chang,(July 2008) "A Hardware-Efficient H.264/AVC Motion-Estimation Design for High-Definition Video," IEEE Trans. Circuits Syst. I, vol. 55, no. 6, pp. 1526-1535.
17) Zhou Wei, Duan zhen-min, and Zhou xin, "Efficient fast motion estimation algorithm for H.264 based on polynomial model," IMACS Multi conference on computational engineering in System Applications, pp. 1654-1657, Oct. 4-6, 2006.