I.J. Image, Graphics and Signal Processing, 2012, 2, 37-43
Published Online March 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2012.02.06

Fuzzy Logic Based Three Step Search Algorithm for Motion Vector Estimation.

Suvojit Acharjee
Department of Electronics and Telecommunication Engineering, Jadavpur University
Jadavpur, Kolkata, India
E-mail: acharjeesuvo@gmail.com

Sheli Sinha Chaudhuri
Department of Electronics and Telecommunication Engineering, Jadavpur University
Jadavpur, Kolkata, India
E-mail: shelism@rediffmail.com

Abstract— Motion compensation process is the most computationally expensive operation in the entire video compression process. Fast motion estimation technique plays a very important role in video compression standard. In Block Matching Algorithm Full Search Algorithm produces the best result for motion vector estimation. But Full Search algorithm is a time consuming and computationally expensive process. The Challenge is to reduce the computational complexity of Full Search algorithm without losing too much quality at the output.
In this paper we propose to implement the fuzzy logic based Three Step Search algorithm. This Algorithm performs better than the Three Step Search(TSS), New Three Step Search(NTSS), Four Step Search(FSS) algorithm.

Index Terms— **Three Step Search, Fuzzy logic based Three Step Search, Motion vector Estimation with Fuzzy, Fuzzy Block Matching Algorithm**

I. INTRODUCTION

We live in the age of multimedia and internet. Hence the issues regarding downloading video streams and storing them in various media have become very important, as they need a lot of memory. The ISO Moving Picture Experts Group (MPEG) video coding standards is for compressed video storage in physical media like hard disk, and International Telecommunications Union (ITU) [13] addresses the real time point to point or multipoint communication over a network. In both the cases the entire compression and decompression process is largely the same. Figure 1 shows the block diagram for video compression process. The most computationally expensive part in the compression process is the Motion Estimation. Motion Estimation examines the movement of objects in sequence to try to obtain the vectors representing the estimated motion. Encoder side estimates the motion of the current frame with respect to previous frame. A motion compensated image of the current frame is then created. Motion vector is then transmitted to decoder. Decoder reverses the whole process and creates a full frame. This way motion compensation uses the knowledge of object motion to achieve data compression. There are two main techniques for motion vector estimation.

a) *Pel-Recursive Algorithm*
b) *Block Matching Algorithm*

Pel-Recursive Algorithm (PRA) [11] introduced by Netravali and Robbins, defines motion vector for every pixel which comprises the location of the pixel of the current frame in the previous frame. In BMA it is assumed that every pixel within a macro block has same motion activity and produce one motion vector for each macro block. The main idea behind block matching is to divide the current frame into number of macro blocks of fixed size and create a motion vector which comprises the location of the macro block of the current frame in the previous frame. Usually the macro block is taken as a sequence of 16 pixels and search area is up to 7 pixels on all fours

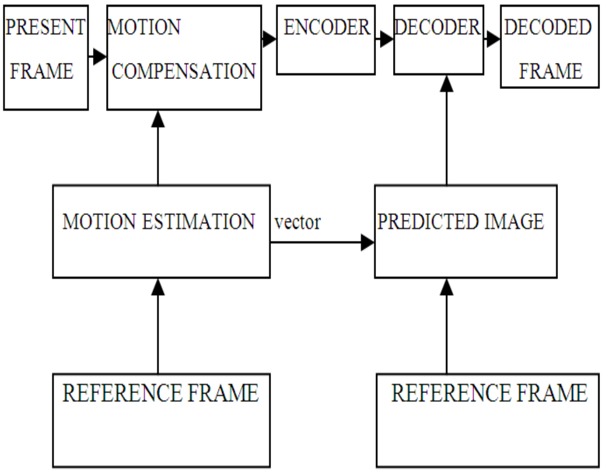

Figure 1. Block Diagram for Video Compression

sides of the corresponding macro block in previous frame. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block. There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) [9] given by equation (1).Another cost function is Mean Squared Error (MSE) [9] given by equation (2).

$$MAD = 1/N^2 \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} | cufra(i, j) - refra(i, j) | \quad (1)$$

$$MSE = 1/N^2 \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (cufra(i, j) - refra(i, j))^2 \quad (2)$$

N= Size of the macro block.

cufra(i, j)= Intensity value for the current frame.

refra(i,j)= Intensity value for the reference frame

Peak Signal to Noise Ratio(PSNR) characterizes the motion compensated image that is created by using motion vectors and macro blocks from the reference image. PSNR can be calculated using Equation (3).

$$PSNR = 10 \log \left[\frac{(PtoP)^2}{MSE} \right] \quad (3)$$

PtoP = Peak to peak value of original data.

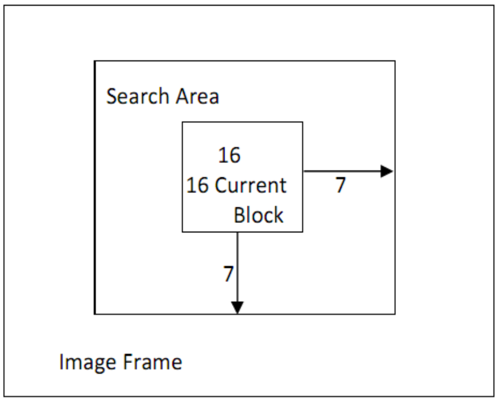

Figure 2. Block Matching of a 16x16 macro Block within a search area of 7 pixels

Block matching algorithm for motion estimation is accepted in all the video coding standards proposed till date. It is easy to implement in hardware and real time motion estimation and prediction is possible. Exhaustive search (Full Search) is the most time and computationally expensive algorithm. Later the other algorithms implemented to reduce the computation cost are Three Step Search (TSS), Four Step Search (FSS), New Three Step Search (NTSS), Diamond Search (DS), and Adaptive Rood Pattern Search (ARPS) etc. In this paper we propose to implement Fuzzy logic based Three Step Search Algorithm.

Section I is a brief introduction about the motion vector estimation and video compression. Section II is a short review on the different research works about motion compensation. Section III describes the Three Step Search algorithm. Section IV presents our proposed algorithm. In Section V the performance of our proposed algorithm is compared with different block matching algorithms.

II. LITERATURE REVIEW

Many algorithms have been developed to reduce the computational complexity of motion vector estimation. Pel-Recursive Algorithm , introduced by Netravali and Robbins[11], defines motion vector for every pixel by the use of steepest descent method. Bargman [14] proposed to use Newton Raphson method instead of using steepest descent method. Walker and Rao [10] introduced a variable step size for steepest descent method.

For block matching algorithms, Full search or Exhaustive search provides best result. Later different algorithms were developed to reduce the computation cost of Full search algorithms. Earlier square search pattern of different length was adopted. Algorithms like Three Step Search [2], New Three Step Search [3], Four Step Search [4], Block Based Gradient Descent Search [5], Simple and Efficient Search [12] etc have adopted the square search pattern. The square search pattern can be classified in '+' and 'x' pattern. Two dimensional logarithmic search [1] has adopted the '+' search pattern where cross search algorithm [17] has adopted 'x' search pattern.

Then diamond type search pattern minimizes the process time. Different algorithms adopted the diamond search pattern as unrestricted center biased diamond search algorithm [8], New Diamond Search Algorithm [6] etc. Adaptive Rood Pattern Search [16] uses the motion vector of the macro block to its immediate left to predict the motion vector of the current macro block. This algorithm uses the diamond type search pattern. But ARPS performs better than other algorithms. Recent studies show that other patterns like hexagon [7], octagon [15] provides better result than diamond pattern. Other approaches to reduce the computation cost of Full search are Fast full search algorithm [19] and Successive elimination algorithm [18] etc.

III. Three Step Search Algorithm

This is one of the earliest attempts for Fast Block Matching Algorithm and dates back to 1980s. Koga [2] introduced this algorithm in 1981. Three Step Search algorithm first adopted the square search pattern. Different Steps of the algorithm are:

1. Set the search location at center and Set the Step Size S=4
2. Search the center and eight locations +/-S around the center. So total search point is nine locations.
3. Using one of the cost function MAD or MSE, cost is calculated for nine locations.
4. From these nine locations searched so far it picks the one giving least cost and makes it the new search origin.
5. It then sets the new step size S = S/2, and repeats similar search for two more iterations until S = 1.

At that point it finds the location with the minimum cost function and the macro block at that location is the best match. The calculated motion vector is then saved for transmission. So total points searched for Three Step Search is 25.

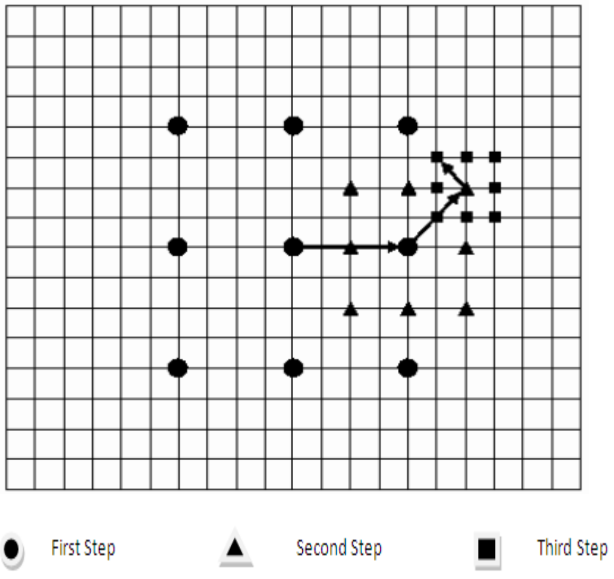

Figure 3. Three Step Search Procedure

This way the Three Step Search reduced the search computation by nine times. For search parameter p=7, computation cost for ES is 225 macro blocks whereas computation cost for TSS is 25 macro blocks. So the computation cost is reduced by a factor of 9 in the case of Three Step Search.

IV. Proposed Algorithm

The Three Step Search algorithm searches all the four side of a macro block. But sometimes the search at all the four side of a macro block is unwanted. The change in intensity from the darker region to the lighter region or from the lighter region to the darker region is called the EDGE region of an image. The macro block positioned on one side of edge region need not to be searched at the other side of the edge for best match. As an example if a macro block is at the lighter side of the edge then search at the darker side of the edge is unwanted. So in this algorithm a fuzzy membership value according to intensity is introduced for every macro block. Now searching the macro block of the reference frame for the best match only can continue if the fuzzy membership value of that block is in the allowable range of the current macro block of the present frame. The search location and all other steps are similar with the conventional three step search. The proposed algorithm is like almost three step search and can be described like this:

1. Calculate fuzzy membership value for every macro block of the reference frame.
2. Calculate fuzzy membership value for every macro block of the current frame.
3. Set the search location at center and Set the Step Size S=4
4. Whether the fuzzy membership value of the macro block of the previous frame is in allowable range with the fuzzy membership value of the macro block of the current frame.
5. If yes then calculate the cost function for that macro block else skip the calculation.
6. The same process described in step 4 and 5 for center location is repeated for all eight locations +/- S around the center.
7. If calculation is skipped for all the nine locations then we keep the search origin same.
8. Else from these nine locations searched so far it picks the one giving least cost and makes it the new search origin.
9. According to the three step algorithm new step size is S=S/2 and repeats the similar search for two more iterations until S=1.

At the end of the search we find the location with the minimum cost function and the macro block at that location is the best match. The calculated motion vector is then saved for transmission. In this proposed algorithm unwanted calculations can be skipped which make the whole process faster. The worst case computational requirement is 25 block matches.

The Fuzzy Membership value calculated for every macro block can be anything that defines the intensity characteristics (lighter or darker) of that macro block. As an example the normalized mean of the intensity values of that macro block can be used as a fuzzy membership value for that macro block. The mean can be calculated using equation (4)

$$Mean = 1/(255 * N^2)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} frame(i,j) \qquad (4)$$

N= Size of the Macro Block
frame(i,j)= intensity value at i,j.

As 255 is the maximum intensity for an image. So this value is need to normalize the value of mean between o to 1.

This process has an advantage that the quality of reconstructed image and speed of the calculation can be controlled by controlling the allowable range. To increase the calculation speed the range should be small and for better quality the range should be large.

In the Figure 4 the current macro block is in the lighter region. Upper side of the macro block is the darker region. So search at that region for best match is unwanted as it is outside the allowable range of fuzzy membership value of the macro block of the current frame. So the blue colored macro blocks in the figure shows the skipped search areas by the proposed algorithm which needs to be searched for Three Step Search process. According to the figure for this macro block computational requirement is 19 block matches.

V. SIMULATION

The proposed algorithm is simulated using the table tennis sequence and Foreman Sequence. First 78 frames of table tennis sequence and first 50 frames of foreman sequence are taken for simulation. The performance of the proposed algorithm is then compared with the performance of different block matching algorithm as Three Step Search (TSS), New Three Step Search (NTSS) and Four Step Search

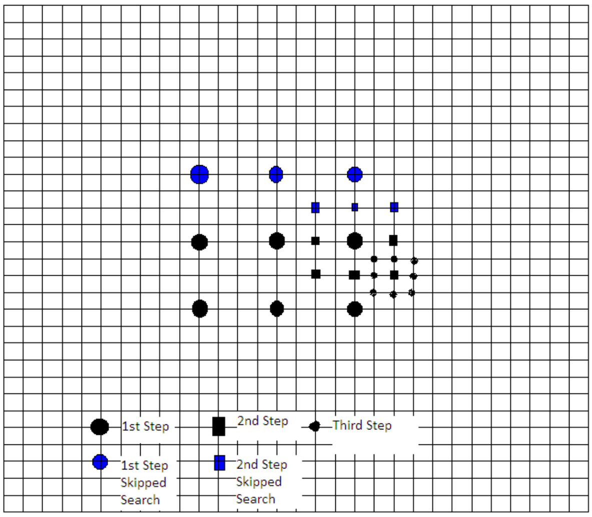

Figure 4 Fuzzy Logic Based Three Step Search Procedure.

(FSS) algorithm. In the above simulation procedure MAD used as cost function, macro block size is 16x16, search parameter p=7. In the simulation process first frame and after every 6th frame one original frame is transmitted. Figure 5,6,7 shows the 16th no of frame in original, reconstructed image after TSS procedure and reconstructed image after Proposed algorithm procedure. Figure 8 shows the time required to simulate first 78 frame of the table tennis sequence. Here our proposed algorithm, the Fuzzy Logic Based Three Step Search algorithm takes less time than other algorithms. In Figure 9,10 the performance of the proposed algorithm compared with different Fast Block matching algorithms for Table tennis Sequence. From the figures it can be concluded that proposed algorithm producing a very good result with almost same picture quality. Proposed Algorithm also simulated using first 50 frames of the foreman sequence.

Figure 5. Original table tennis Sequence 16th frame

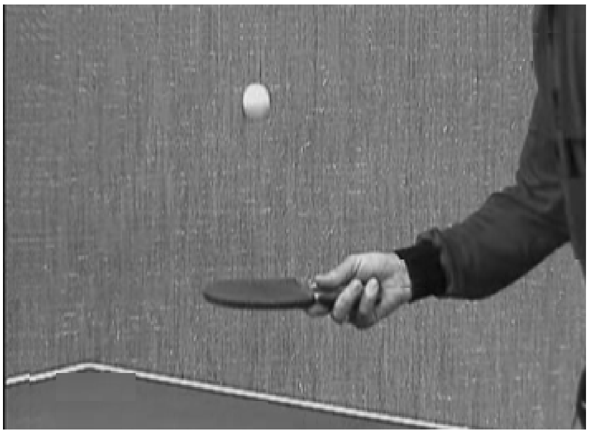

Figure 6. After TSS procedure reconstructed Table tennis sequence 16th frame

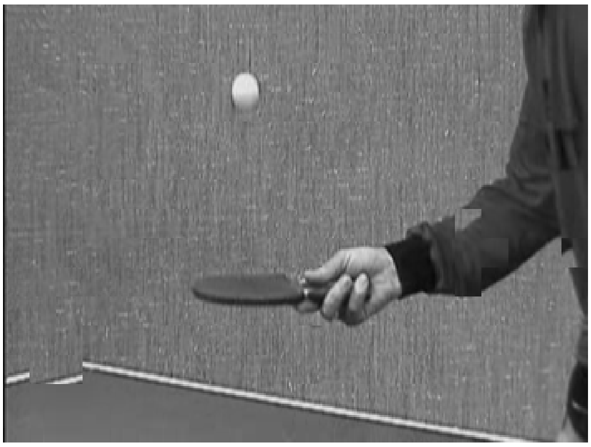

Figure 7. After Fuzzy Logic Based TSS procedure reconstructed table tennis sequence 16th frame

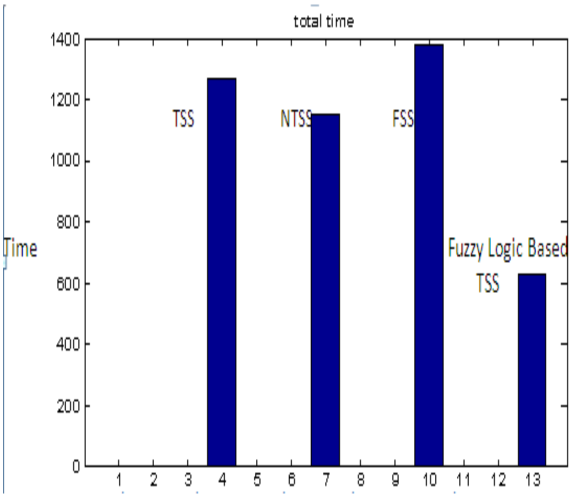

Figure 8 Total Time in Seconds taken by different algorithm for table tennis Sequence

In Figure 11, 12, 13 the performance of the proposed algorithm compared with the other Fast Block Matching Algortihm like TSS,NTSS,FSS etc.From the figures it can be observed that with almost the same picture quality fuzzy logic based three step search algotirhm produces better result than other algorithms. It should be noted that the first frame is always sent in full, and so are some other frames that might occur at some regular interval (like every 6th frame). The standards do not specify this and this might change with every video being sent based on the dynamics of the video. In this example after every 6th Frame one frame is sent in original. So no calculation takes place after every 6th frame. So after every 6th frame PSNR and computation becomes zero.

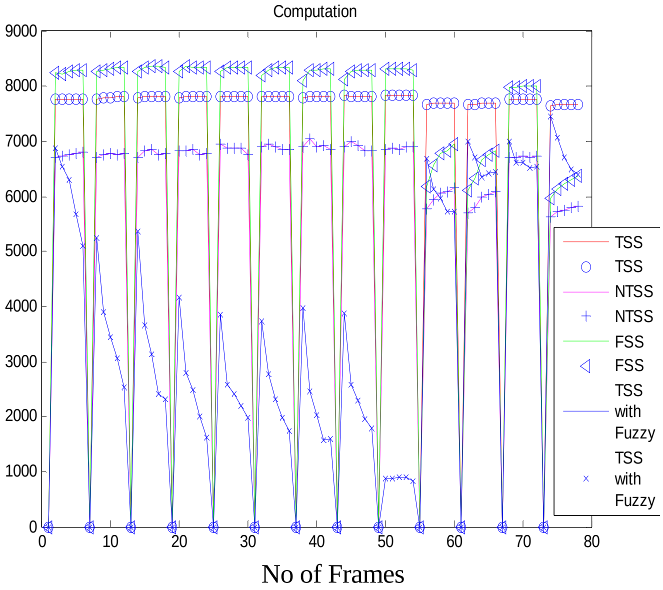

Figure 9. Total no of computation required by Different Algorithms for table tennis sequence

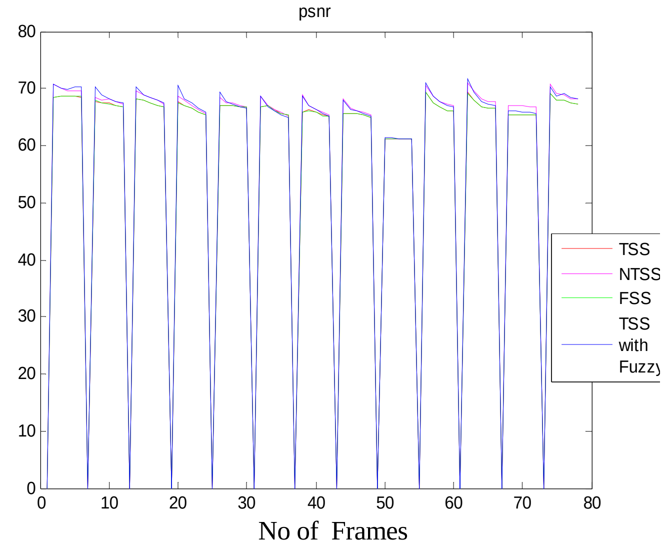

Figure 10. PSNR performance by different algorithms for table tennis sequence

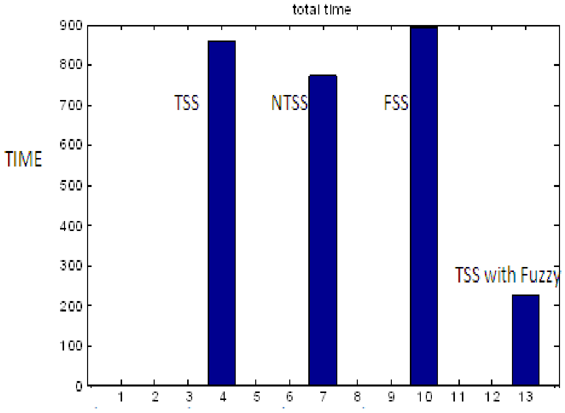

Figure 11. Total Time required by different algorithms for Foreman Sequence

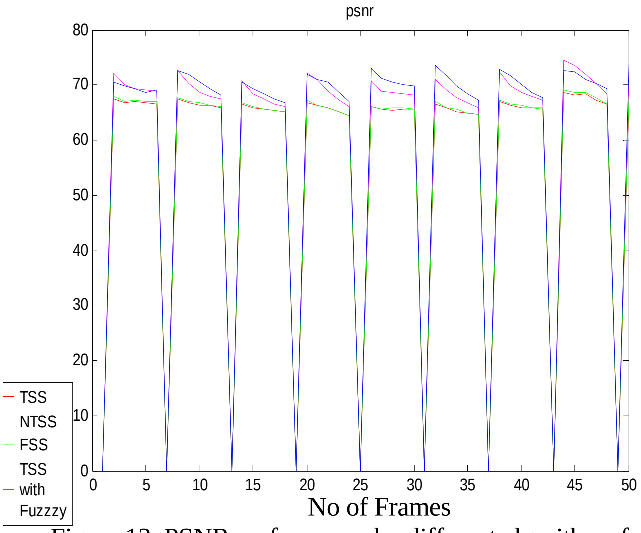

Figure 12. PSNR performance by different algorithms for Foreman sequence.

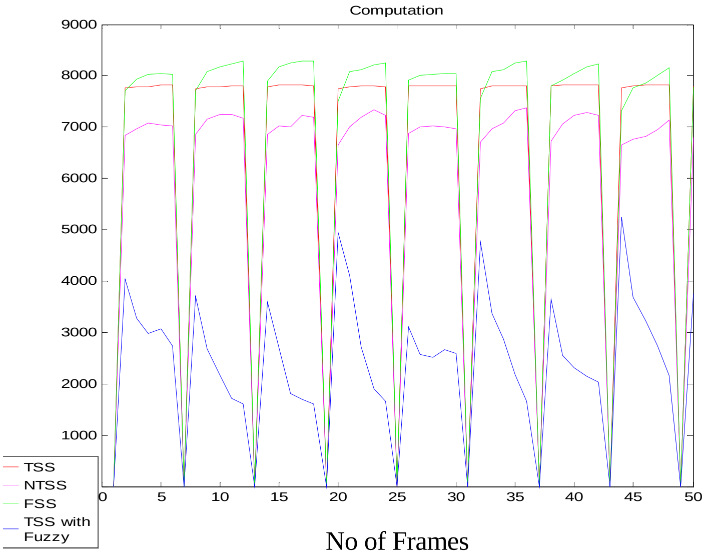

Figure 13. Total Computation required by different algorithms for Foreman sequence

VI. CONCLUSION

The Fuzzy Logic Based Three Step Search algorithm in this paper reduced computation time especially in the edge region of image. As the computation time is reduced, the total time to complete the process is also reduced but the PSNR almost stays same. This concept can be used with the other block matching algorithm also. This process has an advantage to control the quality of the image and the speed of the process as required, by controlling the allowable range.

REFERENCES

[1] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COMM-29, pp.1799–1808, Dec. 1981.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, andT. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. National Telecommunication. Conf., New Orleans, LA, Nov.29–Dec. 3 1981, pp. G5.3.1–.3.5.

[3] R. Li, B. Zeng, and M. L. Liou, "A new three step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438–442, Aug. 1994.

[4] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313–317, June 1996.

[5] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits System Video Technol., vol. 6, pp. 419–423, Aug. 1996

[6] Shan Zhu, and Kai-Kuang Ma, " A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", *IEEE Trans. Image Processing*, vol 9, no. 2, pp. 287-290, February 2000.

[7] Ce Zhu, Xiao Lin, and Lap-Lui Chau,"Hexagon based Search Pattern for Fast Block Motion Estimation " , IEEE Trans. On circuits and systems for video technology, Vol.12, No.5,May 2002

[8] J.Y. Tham, S. Ranganath, M. Ranganath, A.A.Kassim, "A novel unrestricted centerbiased diamond search algorithm for block motion estimation", IEEE Trans. Circuits Systems VideoTechnol. 8 (August 1998) 369–377.

[9] Y.Q.Shi and H Sun, "Image and Video Compression for Multimedia Engineering "

[10] Walker D.R., Rao.K.R. "Improved Pel-Recursive Motion Compensation", IEEE trans on Communications, vol com 32,no. 10, Oct. 1984, 1128-1134.

[11] A.N.Netravali and J.D.Robbins, "Motion compensated television coding-part1", Bell syst. Tech.J., Vol.58,pp 631-670,Mar.1979

[12] Jianhua Lu amd Ming L.Liou " A Simple and Efficient search algorithm for Block Matching Motion Estimation", IEEE trans circuits And Systems for Video Technology, Vol 7,no,2,pp. 429-433 april 1997

[13] A Barjatya, “Block Matching Algorithms For Motion Estimation,” DIP 6620 Spring 2004 Final Project Paper

[14] H.C.Bergmann, “Displacement estimation based on the correlation of image”,IEEE proceedings of International conference on Image processing,York,England,july 1982,pp. 215-219.

[15] L.P.Chau and C.Zhu, “A fast octagon-based search algorithm for motion estimation”, Elsevier Science Signal processing 83,2003,pp. 671-675.

[16] Yao Nie and Kai Kuang Ma, “Adaptive Rood Pattern Search for Fast Block Matching Motion Estimation”, IEEE transaction on Image Processing,Vol 11,no. 12,1442-1448,December 2002.

[17] M. Ghanbari, “ The Cross Search Algorithm for motion Estimation”, IEEE Transaction on Communication vol. 38, pp. 950-953, july 1990.

[18] W. Li and E. Salari, “Successive elimination Algorithm for motion Estimation”, IEEE transaction on Image Processing ,Vol 4,no 1,pp. 105-107,January 1995.

[19] Y.C. Lin and S.C. Tai, “Fast Full Search Block matching algorithm for motion compensated video compression”, IEEE transactions on Communications, Vol 45, no. 5 pp. 527-531, May 1997

Suvojit Acharjee was born in Murshidabad, West Bengal, India on March 16th , 1988. Suvojit is now a M-tech student in Jadavpur University. Suvojit is awarded the degree of Bachelor of Technology in recognition of his successful completetion of of the 4 year programmed Electronics and Communication course from Future Institute of Engineering and Management, Kolkata under the university West bengal University of Technology, Kolkata , India on June , 2009.

His current research interests include Robotics, Soft Computing, Image Processing, Video Compression and algorithms.

Dr. Sheli Sinha Chaudhri is an associate professor of jadavpur University. Her research interests include soft computing, Image processing , Multimedia.