COMPARATIVE STUDY OF BLOCK MATCHING ALGORITHMS FOR MOTION ESTIMATION

[1]HUSSAIN AHMED CHOUDHURY, [2]MONJUL SAIKIA

NERIST, Arunachal Pradesh, India
Email:-hachoudhury10@gmail.com, monjuls@gmail.com

Abstract-In video compression, we remove the temporal redundancyand spatial redundancy by using Hybrid Video Codec. In encoder part of Hybrid Codec, we find the motion vector(MV) of the candidate block of current frame in reference frame using block based motion estimation technique. Motion estimation (ME) is to be done in the encoder side to find the best MV so that it can be applied on stored frames by motion compensated block to generate the predicted video in video compression. Various fast search techniques are employed to find the best matching point i.e. best motion vector(MV) in reference frame. In this paper we have reviewed in brief already implemented various block based motion estimation techniques namely full search (FS),two Dimensional Logarithmic search(2DLS), three step search(TSS), diamond search algorithm (DSA), Cross Search(CS),Binary Search(BS), Hexagonal Search Pattern(HXSP), Adaptive Rood Pattern Search(ARPS) etc. We also try to review their performance based on the number of searching points and PSNR.

Keywords: Video compression; Motion estimation; Block matching; MAD; MSE; FS;TSS;CS;DST etc.

I. INTRODUCTION

Video communication has found to have a wide range of application in modern era. The application domains may be of Conversational video such as video telephony, video conferencing through wired or wireless medium and Streaming Video such as Video on demand, HDTV etc. Digital video communication requires a high bandwidth and storage space. A video is a continuous stream of images taken by camera and it has temporal aspect of the signal that not only varying in space but also in time. As we know spatial redundancy is removed by transform domain coding and temporal redundancy can be exploited using predictive coding so we need the combination of both for video coding known as Hybrid video Codec.In the encoder the input video is subtracted from predicted video to get the predicted error of the residual signal. The residual signal is then encoded using a DCT(Discrete Cosine Transform), quantizer and entropy coder. Encoder has inbuilt decoder part to reconstruct the error frame which won't be same due to quantization at encoder side. At the encoder the predicted video isgenerated using the past video frames. When the frames/images are coming, we store the frames in memory and then with thestored frames and motion vectors given by the motion estimator the motion predictor generates the predicted video.

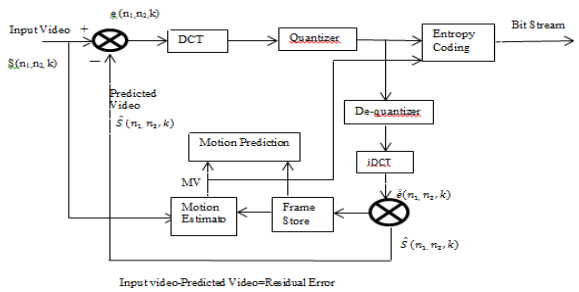

Fig.1: Hybrid Codec Encoder side

The motion estimator block gives motion vector of each block to predict how much displacement every block of the image has undergone in reference frame by using motion estimation techniques. In decoder we combine the frame store and predicted video block into a single block and with the MV it predicts the predicted frame. When we store the incoming frames of a video then only both the motion estimator or estimation block and motion compensation block will work.

II. INTRA AND INTER CODED FRAMES

The question now arises that how can we predict and encode the first frame as it has no past frame for reference? Then the concept of Intra-frame coding has come. Then we have to consider the first frame of each video as a still image which will have only spatial or intra-frame redundancy. The frames that contains all of its own information and results less error when encoded is known as Intra coded frame or I-frame. All subsequent frames have both temporal and spatial redundancy i.e. they use inter code redundancy for compression. These are called inter coded frames. In intra coded more bits are needed because temporal redundancy is not considered.In this paper section III explains the details ME, Matching criteria, section IV explains the various techniques, section V gives the relevant techniques, section VI explains the performance of techniques and VII discusses the conclusion.

III. MOTION ESTIMATION AND BLOCK MATCHING

In video between successive frames there is great deal of similarity. To find the displacement vector for the candidate block of current frame in reference frame (past frame or future frame) one has to perform

matching between two consecutive frames. Comparing pixel to pixel intensity between frames we cannot be sure that this pixel corresponds that pixel in two frames because two different pixels may have same value. So instead of matching pixel to pixel value, a region or block based matching is done over past frames. The matching is done by searching the position corresponding to the minimum value of matching criteria which gives the motion vector. This whole process of finding the best match is known as *Motion Estimation.* Since it is done per block basis it is called *Block based Motion Estimation Technique (BMA).* For BMA we subdivide the image into *NxN* non overlapping blocks.

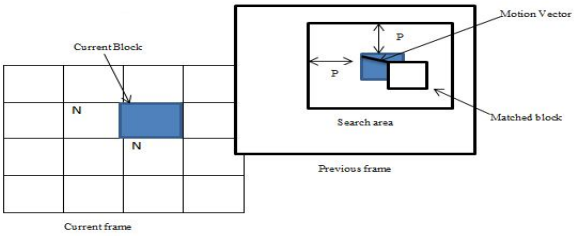

Fig.2: Basic idea of block based Motion estimation

Forward and backward prediction: (FP &BP):
In encoding frame k, we can use the past frame i.e. $(k-1)^{th}$ frame as a reference to predict what frame number k is going to be. As we are predicting future it is known as *Forward prediction* and for finding out MV and for doing FP that we are going back in time so it is called *backward motion estimation*. This is one type of unidirectional prediction. In the reverse case if we use future frame for reference to encode current frame we are predicting the past going ahead in time so the process is called *backward prediction* or *forward motion estimation technique.* The combination of both i.e. in some cases both past frame and future frame are used as a reference. This type of prediction is called *bidirectional prediction* and the frame used is known as Bi-directional predicted frame called B-frame.

Matching Criteria:
In order to find out the co-relation between consecutive frames k and k+1 and to find out the best MV in reference frames, the following criteria are considered. The position which corresponds to the minimum value of the following search criteria will give the best motion vector(MV). Matching is usually performed over a macro block of different size as discuss bellow. Among three different approach MDA is most commonly used as its computation cost is low.

Mean Square Error (MSE):
The BMA tries to find out the position having MV (d_1,d_2) with minimum MSE which is calculated by the following equation

$$MSE(d_1,d_2) = \frac{1}{N_1 x N_2} \sum_{(n_1,n_2)\in B} \sum [s(n_1,n_2,k) - s(n_1 + d_1, n_2 + d_2, k+1)]^2$$

In above equation N_1xN_2 is the macro block size, $S(n_1,n_2,k)$ is the co-ordinate position of k^{th} frame along n_1 and n_2 direction and B is an N_1xN_2 block of pixels.

*Mean Absolute Difference (MAD):*The MAD for the position $(d_1,d_2)$is given by

$$MAD(d_1,d_2) = \frac{1}{N_1 x N_2} \sum_{(n_1,n_2)\in B} \sum |s(n_1,n_2,k) - s(n_1 + d_1, n_2 + d_2, k+1)|$$

Sum of absolute difference (sad): Sum of absolute difference (SAD) is defined as:

$$SAD(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} |X_{m,n} - X^R_{m+i,n+j}|$$

IV. SOME TECHNIQUES

1.Complete Search or Full Search(FS)*:* As the name implies it searches all possible positions for best match by calculating Minimum MAD in the reference frame for the candidate block in current frame([21],[14]-[16]). As it searches all the positions, it at least it guarantees to find the accurate match i.e. optimal motion vector in reference frame irrespective of its high complexity. If we can search in all direction taking the candidate block as center and with w pixels then the total search point will be $(2*7+1)^2$=225. It shows that for each candidate block Complete or Full search will search 225 locations in reference frame for minimum cost which will give the required motion vector. The computational complexity is of order N^2 for a block size of N*N.

2. Two-Dimensional logarithmic Search (2DLS: Jain and Jain introduced this technique for fast motion estimation ([16],[17]). As search space two dimensional and step size reduces logarithmically so it is named so. Initially we assume a search range say p=8 and we consider the position of the candidate block as the center of the search window and search space becomes $(2*8+1)^2$ =289 points. The search takes place as follows:

i. Initiallysearch for minimum cost is made at 4 points at the end of plus ('+') sign with step size equal to half of search range i.e. p/2=4.We calculate minimum cost at 5 points including the center but not anywhere else.
ii. If we find the minimum cost other than center position we will shift the center to the point of minimum and carry out search as step 1 by taking same step size of previous step. This time we need to calculate only at 3 points out of 5 because at 2 points we have already calculated at last step.
iii. If the minimum is happens to be at center of earlier step, we reduce the step size by 2 i.e. new step size will become 2. Now calculate 4 points along '+' sign at a distance of 2 around center.

If the minimum is not at the center follow step 2 else reduce step size to 1.Search ends when step size becomes 1 because we cannot make p as fraction. Then we search all the neighboring 8 points at the end of both '+' and cross sign('x'). Whichever is minimum point, is the best matching block and it's co-ordinate will give the motion vector with respect to center. If we take the value of w=±7 then best case will search 13 points for best match and in worst case it requires 17 points to be searched.

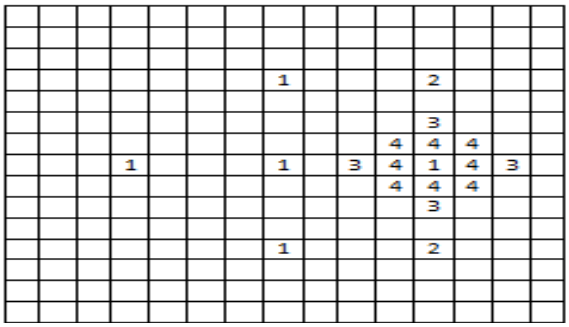

Fig. 3: 2D Logarithmic Search

3.Cross Search (CS): It is just a variation of 2DLS where search is made at the end of a Cross sign (X)([2],[17]). It introduces a predefined threshold value for finding the motion vector.

i. We compare the candidate block position at (0,0) with that of the center position of previous frame at the same position as shown below by calculating Sum of Absolute Difference(SAD) at those points: $|SAD_k(0,0)-SAD_{(k-1)}(0,0)|<T$ where T is some predefined threshold value. If the difference of SAD is <T then the candidate block is stationary and search is not required.

ii. Else we start searching for the best match at 5 points including center around the center along the end points of a 'X' with step size equal to half of the search range.

iii. If the minimum position is other than the center, we carry on search at points by taking the minimum point as new center with the step size of previous step. No need to calculate cost at that position at which cost is already calculated.

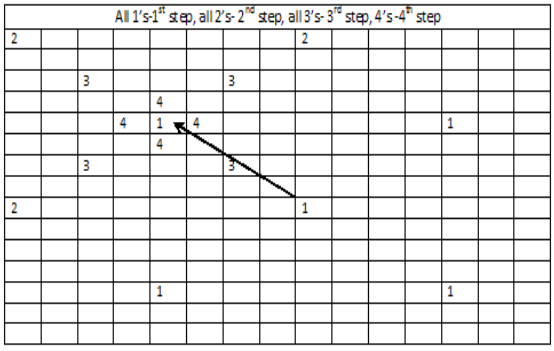

Fig.4: Cross search

iv. Else we reduce the step size by 2 and carry on step 2. Search stops when step size becomes 1. When step size becomes 1, we need not to search all 9 neighbor points rather we calculate only at the end of a '+' or 'X' depending upon the position of the minimum.

The minimum cost position will give us the best match and motion vectors. In best case it requires 13 positions to be searched for a maximum motion displacement of ±7.

4. Three step search (TSS): TSS can be called as a combination of both 2DLS and Cross search where search is done at 9 points both at the ends of a '+' and 'X' sign([15], [17]-[19]). We take the candidate block position as the center of the search window and if we take search window as w=7 then we calculate the initial step size like below:

L = floor(log10(w+1)/log10(2)) , stepMax p = 2^(L-1);

With the formula we get the initial step size of p=4 and our search space becomes $(2*4+1)^2$=9 X9. The TSS proceeds as below:

i. It searches all 9 positions i.e.(-4,-4),(0,-4),(4,-4), (-4,0),(4,0), (-4,4),(0,4) &(4,4) including (0,0)for least cost with step size 4 and shift the center to the new point of minimum.

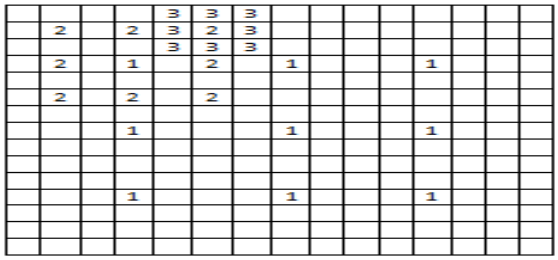

Fig.5: TSS algorithm

ii. Now the step size is reduced to half i.e. p/2=2. Eight new neighbor points are checked for minimum cost at a distance of 2 from the center in all direction and shift the center to the minimum point.

iii. Again the step size is reduced by 2 to make it 1. Now we check all 9 points including center (8 neighbors at a distance of 1 for minimum cost. The position with minimum cost will give us the motion vector and also the position of best match.

In general TSS is another form of 2DLS with better performance. If we assume our initial p=16 then it will become 5 step search(e.g. [21]). For the window size of ±7 we need to check 25 points for the minimum cost. As we increase the value of p, computation will increase.

5.Diamond Search Algorithm (DSA): It searches cost (minimum block distortion, MBD) at the corners of a diamond which may be having 9 checking points called Large Diamond Search Pattern and having 5 checking points called Small Diamond search Pattern([6], [15], [18]).The no. of computations of

DSA may be 9+3*n+4 where n is the no. of iterations([19], [21]). The steps of DSA are:

i. It starts with checking 9 points including 8points of Large Diamond with candidate block at the center. If the minimum cost is found at the center we will follow step 3 else step 2.
ii. The point with minimum cost is taken as the new center to check points of another Large diamond pattern. We do not calculate again at those points where we did at last step. If the minimum cost is located at center, follow step 3 else repeat step 2 until minimum cost lies at center.
iii. We reduce the step size means size of diamond from Large to Small which results in lesser no. of checking points i.e. 5. The position of point which will have minimum cost, MBD, will be the final position of matching block and it will give the best motion vector.

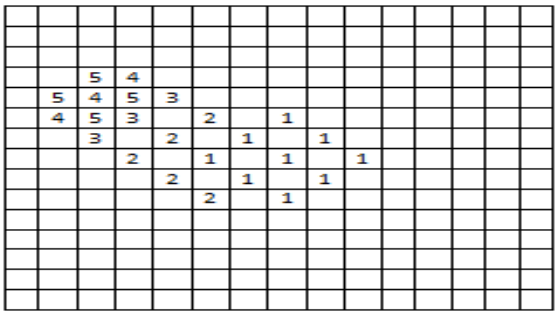

Fig.6: Diamond search Algorithm

6. Hexagonal Search Pattern (HXSP): In this method we start searching for minimum cost at 6 corner points of a regular hexagon and the center of that hexagon([18], [19]). In the next step depending upon a condition we confined our search at 5 points(4 corner points of smaller diamond and center point) otherwise we carry on with hexagonal pattern.

i. We start our search by taking the candidate block at the center of the search window and calculate the minimum cost at 7 points i.e. 6 corner points of regular hexagon and at center point. If the minimum cost point lies at the center of the hexagon, proceed to Step 3; otherwise, proceed to Step 2.

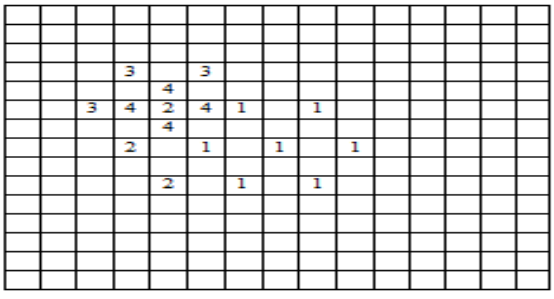

Fig.7: Hexagonal Search Pattern

ii. We take the minimum corner point as our new center for constructing the next hexagon pattern. 3 new neighbor search points are checked, and the cost point is again compared with those points also where we calculated earlier. If the minimum point is at the center of current hexagon, then go to Step 3; otherwise, repeat step 2 until minimum cost point lies at center of search window.
iii. We convert the search from 9 points to fewer 5 points(4 points of a small size diamond and the center point).So we search only at 4 points around center and whichever point is found to have minimum cost will give us final required motion vector.

The number of search points in this pattern in worst case is very large which can be expressed by: NP=7+3*i+4 where NP=No. of search points, i=no. of iterations. In best case it reduces the number of search points.

7.Binary Search Motion Estimation(BS)

This is another new technique for fast motion estimation used by MPEG tools but its performance is not so good because it excludes many search points depending upon the region basis [16].The steps are:

i. We calculate the minimum cost at the grid of 9 points i.e. the 4 corner points of search window and 4 pels at the borders.
ii. Then divide the whole search window into different regions depending on these points.

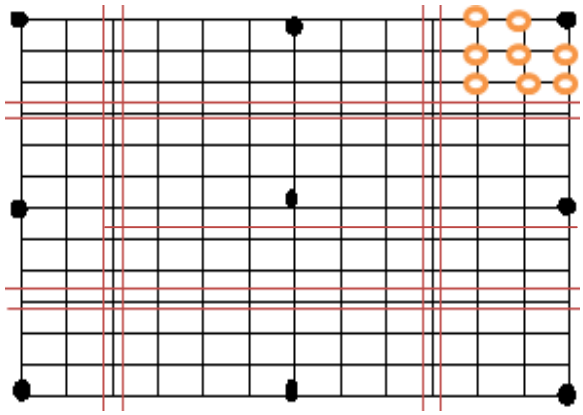

Fig.8: Binary Search(points with which regions are made are denoted by ● and Opoints in region where full search is made)

iii. The region where the minimum point lies, full search is made for the best match.

Though the number of comparisons is lesser but still it requires 33 computations in worst case.

V. OTHER RELEVANT WORKS

In comparison to the other fast block motion techniques, Hexagonal search pattern gives us the better result. This is a modification of DS algorithm. As all the techniques could not be covered in this paper, some other available fast block motion techniques are New Three Step search(NTSS) which is modified version of TSS, four Step Search(4SS), the hybrid of 2DLS & TSS which is known as Orthogonal Search Pattern, Adaptive Rood Pattern Search(ARPS), Cross Diamond Search (CDS), One at

a Time Search(OTS). Out of all these ARPS is found out to be having least search points.

VI. PERFORMANCE OF TECHNIQUES:

During the review we have taken few already implemented techniques and applied them on 'SampleVideo.avi' by taking the search window range of ±7 i.e. search space becomes 15*15=225 and macro block of size 8x8. We got the following simulation result. From the result it is clear that ARPS needs least number of search points and TSS has the 2nd highest.

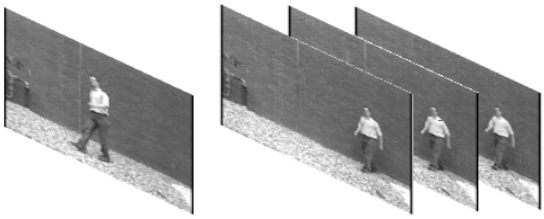

Fig.9: Video used for experiment "SampleVideo.avi"

Table 1: Comparison of Computations and PSNR for different BMA

BMA	Avg. no. of Computations	Avg. PSNR
ARPS	5.5363	33.2457
DS	12.9826.	33.4479
NTSS	17.0407	33.5919
SESTSS	17.3663	33.0461
SS4	16.6104	33.5145
TSS	24.0096	33.2305

In table1 average number of computations needed and average PSNR (between original and predicted) is shown for different BMA (Block Matching Algorithm).

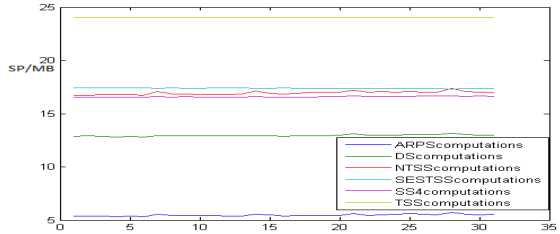

Fig.10: comparison of SP/MB(search point/macro block) for different techniques where SP=search Point, MB=macroblock.

In fig.10 the graph shows number of search point required per macro block to predict its motion vector. TSS (Three Step Search) requires highest (i.e. 24 point aprox.) where as ARPS (Adaptive Rood Pattern Search) needs least number of points to be searched (i.e. 5 point aprox.).

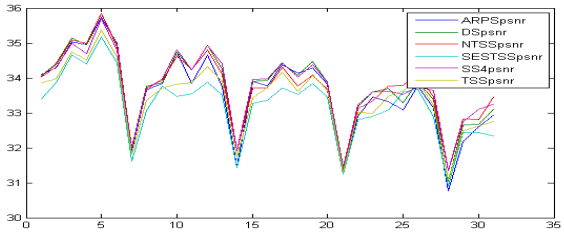

Fig.11: PSNR comparison of different search techniques for 'SamplVideo.avi'

CONCLUSION:

We have reviewed seven different block matching techniques in brief in this paper. It is found that though FS has high computational complexity it shows highest PSNR(Peak Signal to noise ratio) and guarantees the optimal solution. But still to make faster search many fast block based motion estimation techniques are developed though they gives sub-optimal solutions only. During the review, we applied few existing techniques on 'SampleVideo.avi', it is seen that ARPS greatly reduces the number of searching points in comparison to others.

REFERENCES

[1] Puri, H. M. Hang and D. L. Schilling, "An efficient block matching algorithm for motion compensated coding," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc., pp. 1063-1066, 1987.

[2] M. Ghanbari, "The cross search algorithm for motion estimation," IEEE Trans. Commun., Vol. COM-38, pp. 950-953, Jul. 1990.

[3] R. Li, B. Zeng and M. L. Liou, "A new three step search algorithm for block motion estimation," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 4, No. 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. on Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 313-317, Jun. 1996.

[5] Lurng-Kuo Liu and Ephraim Feig, "A block based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6, No. 4, pp. 419-422, Aug. 1996.

[6] S.Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," IEEE Trans. Image Processing, Vol. 9, No. 2, pp. 287-290, Feb. 2000.

[7] P.Cicconi and H. Nicolas, "Efficient region-based motion estimation and symmetry oriented segmentation for image sequence coding," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 4, No. 3, pp. 357-364, Jun. 1994.

[8] M. Ghanbari, *Video Coding, An Introduction to Standard Codecs*, London: The Institute of Electrical Engineers, 1999. Ch.2, 5, 6, 7 & 8

[9] Jianhua Lu, and Ming L. Liou, "A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation", *IEEE Trans. Circuits And Systems For Video Technology*, vol 7, no. 2, pp. 429-433, April 1997.

[10] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", *IEEE Trans. Image Processing*, vol 11, no. 12, pp. 1442-1448, December 2002.

[11] Chun-Ho Cheung, and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", *IEEE Trans. Circuits And Systems For Video Technology*, vol 12., no. 12, pp. 1168-1177, December 2002.

[12] C. W. Lam, L. M. Po and C. H. Cheung, "A New Cross-Diamond Search Algorithm for Fast Block Matching Motion Estimation", Proceeding of 2003 IEEE International Conference on Neural Networks and Signal Processing, pp. 1262-1265, Dec. 2003, Nanjing, China.

[13] HuiGu and Yan-Shan Li, "The research of block motion estimation algorithm in video compression", IEEE transactions on circuits and systems for video technology, 2004.

[14] MS. A.P. Chauhan, R. R. Parmar, S. K. Parmar, S.G. Chauhan, " Comparative analysis on the performance of block matching motion estimation algorithm", Journal of Information, Knowledge and research in Computer engineering Volume 2, Issue 2, ISSN 0975-6760, pp. 366-370, Novembor-2012.

[15] D.V. Manjunatha, Dr. Sainarayanan, "Comparison and Implementation of fast block matching motion estimation algorithms for video Compression", International journal of Engineering Science and Technology(IJEST) Vol. 3, ISSN 0975-5462, pp. 7608-7613, October-2011.

[16] P.C.Shenolikar, S.P.Narote,"Different Approaches for Motion Estimation", International conference on "Control, Automation, Communication and Energy Conservation, International Conference on, 4th -6th June 2009

[17] S.Usama, M. Montaser, O. Ahmed," A complexity and quality Evaluation of Block Based Motion Estimation Algorithms", ActaPolytechnica Vol. 45 No.1,2005.

[18] S.I.A. Pandian, G.J.Bala, Backy A George," A study on Block Matching Algorithms for motion Estimation", International journal on Computer Science and Engineering(IJCSE) Volume 3, No.1,ISSN: 0975-3397, Jan 2011.

[19] R.A.Manap, R.S. Sarban Singh," Performance Analysis of hexagon-diamond search Algorithm for motion estimation using Matlab", Journal of telecommunication, electronic and computer engineering Volume 2 No.2, ISSN: 2180-1843, July-December-2010.

[20] Ce Zhu, X. Lin, LappuiChau, Lai Man Po, "Enhanced Hexagonal Search for fast Block Motion estimation", IEEE Trans. On circuits and systems for video technology Volume 14, No.10, October-2004.

[21] ArohBarjatya, "Block Matching Algorithms For Motion Estimation", DIP 6620 Spring 2004 FinalProject Paper.

[22] Deepak Tauraga, Mohamed Alkanhal, "Search algorithms for Block Matching Estimation", MidtermProject, spring 1998.

[23] Ahmadi, M. M. Azadfar "Implementation of fast motion estimation algorithms & comparison with full search method in H.264," IJCSNS International Journal of Computer Science & Network Security, vol, 8, No.3, pp 139-143, March 2008.

[24] N.B. Yusop,"implementation of four step search(4SS) algorithm for motion estimation in MATLAB", final year Degree report, April-2007.

[25] Murali E. Krishnan, E. Gangadharan and Nirmal P. Kumar,"H.264 Motion Estimation and Applications"*Anand Institute of Higher Technology, Anna University, India*

★★★