"Efficient Motion Estimation using Block Transform Scaling Technique"

Ruchi Jain
Department of Information Technology
MIT, Pune, India
rj.ruchijain23@gmail.com

Trupti Baraskar
Department of Information Technology
MIT, Pune, India
trupti.baraskar@mitpune.edu.in

***Abstract*— Motion Estimation (ME) is the most computationally expensive and resource hungry operation in the entire compression process. ME is the process of generation of Motion Vector (MV) and also determines how the motion compensated prediction frame is created from previous frame. But, ME's computational complexity poses great challenge for real time implementations. Among all ME algorithms, Block matching algorithm is commonly used. It is simple and efficient. The main purpose of Block matching algorithm is to determine the displacements of each block of pixel between two successive frames. Not all types of videos are suitable for one type of Block matching algorithm such as Gradient search which is more suitable for slow videos. Full search is mainly used for non-real time videos and 4SS is more suitable for fast videos. In this paper, we have removed this dependency to particular algorithm by making the blocks of frames of a video uniform by using Block Transform Scaling and also improves the PSNR and Estimation Time.**

Keywords- Motion Estimation, PSNR, MAD, Block Matching algorithm, DCT, MSE

I. INTRODUCTION

Video compression becomes necessary for an efficient data storage as well as for transmission of internet video. Compression is useful because it reduce the consumption of resources such as data space or transmission capacity [1]. ME is defined as finding the MV, which is nothing but the displacement of the coordinate of the best similar block in previous frame for the block in current frame. The design objective of video compression is to minimize the average number of bits used to represent a video sequence maintaining sufficient video quality. In a video sequences, there exists a high level of redundancy between consecutive frames which means changes are minimal from one frame to another. ME is the technique which reduces these temporal redundancies in the video sequence. The temporal redundancy reduction is to encode first a reference frame and for the consecutive frames encode only the difference between the reference frame and the current frame [2]. Among the different motion estimation algorithms Block Matching (BM) algorithm is the most common method of motion estimation for video Coding standards [3] [11].

A. Intra-frame & Inter-frame compression

A video can be viewed as a sequences of images. In image coded terminology, an image is known as a frame. The approach to compressing a frame or a sequence of frames can be viewed through two windows:

- Intra-frame compression: In this technique each frame is considered as a non correlated segment of an image sequence and reduces only the spatial (pixel) redundancies present in an image.
- Inter-frame compression: In this technique each frame is consider as part of an image sequence and employs temporal predications. This also increases the efficiency of data compression [4].

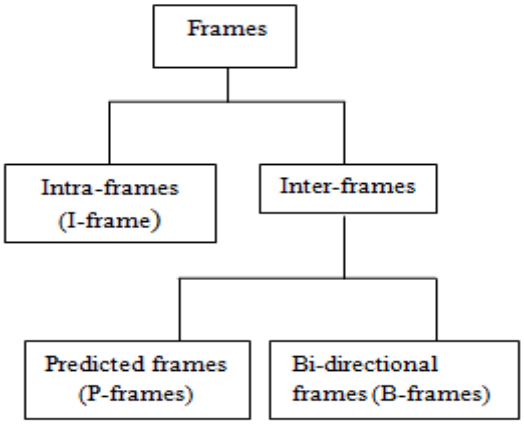

Figure 1. Classification of frame

According to the predications, Inter-frame is further divided into P-frames and B-frames. In encoding frame k, we 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 doing FP and for finding out MV we are going back in time so it is called *Backward Motion Estimation*. This is one type of unidirectional prediction. If combination of both i.e. in some cases both past frame and future frame are used as a reference. Than this type of prediction is called *Bidirectional Prediction* and the frame used is known as Bi-directional predicted frame called B-frame [5][11].

B. Motion Estimation Algorithms

Motion estimation algorithms has seen as the highest activity and have attracted much attention in research and industry, because of these reasons:

1. It is the most computation demanding algorithm of a video encoder (about 60-80% of the total computation time) which limits the performance of the encoder in terms of encoding speed.
2. The motion estimation algorithm has a high impact on visual performance for a given bit rate of an encoder.
3. Finally, the method to extract MV from the video is not standardized, thus it is being open to competition [6] [11].

C. *Classification*

Motion estimation algorithms can be categorized into time-domain and frequency-domain algorithms [6][11].

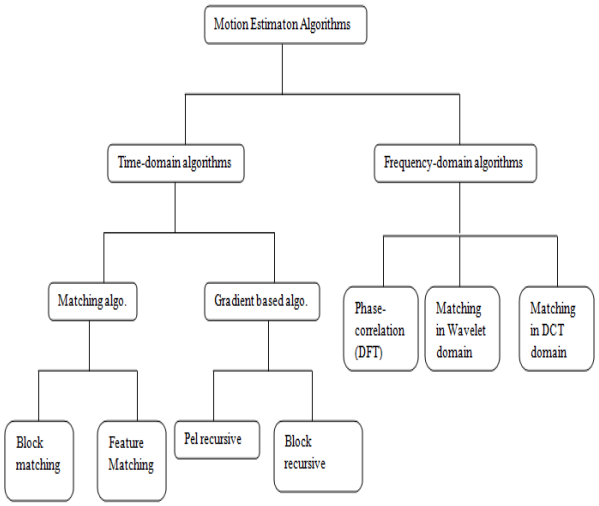

Figure 2. Classification of motion estimation algorithm

II. RELATED WORK

A. *Block Matching Algorithm*

To find the MV for the candidate block of current frame in reference frame (past frame or future frame) one has to perform matching between two consecutive frames. The matching is done on the frames by finding the position corresponding to the minimum value of matching criteria which in result gives the MV. 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 [5][11].

Depending on which search strategy is chosen, the search range can include all possible displacements (for the Full Search approach) or the only selected displacement (for the Fast Search approaches) within a reference block which is larger than the candidate block. Different search strategies may lead to different block matching motion estimation approaches which may result in different performance. Basically, there are three types of search strategies:

- Coarse to fine approach: In the first step, a large Step Size (SS) is taken at the centre of original block for finding best match. In next step, SS is reduced and search carry out around the best match of previous step. This strategy is most commonly used for suboptimal fast search approaches.
- Aggressive approach: In the first step, a large SS is taken at the centre of original block for finding best match. In next step, when the MAD/MSE value of the current stage is larger than of previous stage then SS is reduced.
- Hierarchical approach: In this approach, search space is divided into regions and for each regions a center point is selected. The best match is chosen among all the center points and a full search is done over entire best match center point region. This strategy is the basis for the Hierarchical Search [11].

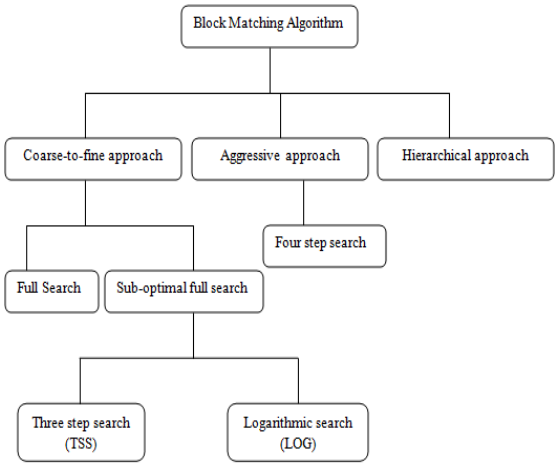

Figure 3. Classification of block matching algorithm

1. *Full-Search*

Full-Search (FS) algorithm evaluates all positions in the window search of (2W + 1)x(2W +1) size. The smallest distortion calculated between the reference block and each block in the window search is used to determine the best matching. The FS is by nature a brute force algorithm and involves a high computational cost. However, it is simple and guarantees a high accuracy in finding the best match [9] [11].

2. *Two-Dimensional Logarithmic Search*

Two-Dimensional Logarithmic Search (2DLS) technique introduced by Jain and Jain for fast motion estimation. As in this step size reduces logarithmically and search space is two dimensional so it is named so. Firstly, we assume a search range and by taking SS equal to half of the search range will search for minimum at 4 points at the end of plus(„+“) sign including centre. If we found minimum other than centre will shift the centre to that point which is minimum and carry out search again with SS of previous step. If minimum is centre itself only then we reduce the SS by 2 and carry out search again until SS become 1. Search ends itself when SS becomes 1. Then we search all neighbour 8 points at the end of ‘+’ & ‘x’ both. And which is minimum out of these 8 points, MV is set according to that [5] [11].

3. *Three-Step Search*

The Three-Step Search (TSS) algorithm starts with evaluating the distortion in the central block and eight blocks around it, at an initial distance in pixels. The best candidate is taken as a new search center and eight block neighbours are selected around it, at half of the initial

distance. This process is repeated until the distance is equal to one [5] [11].

4. New Three Step Search

New Three Step Search (NTSS) improves TSS results by providing a center biased searching scheme pattern (additional 8 points) in the first step. Although there is a complexity due to the addition of points-in the first step, this disadvantage is compensated by having a half-way stop technique. This makes the complexity of NTSS comparable to that of TSS. The Algorithm is widely accepted algorithm for implementing standards like MPEG 1 and h.261 [3] [11].

B. *Comparative analysis of various BMA*

In Table I we have categorized the algorithms into nonlinear method and linear method based on maximum searching points. Most of the algorithms follow nonlinear approach. In our survey CDS algorithm only follow linear search approach and TABLE I also shows the advantage and disadvantage of various BMA [7][8][11].

TABLE I

Method	Algorithm	Advantages	Disadvantages
Non linear	FS	Best picture quality and highest PSNR	Very high computational cost
	3SS	Optimum performance, less complexity, recommended in MPEG2	can't detect small motion
	NTSS	More efficient then TSS for small motion	More complex then TSS
	FSS	Initial small SS so more efficient for small MV	More complex then TSS
	CSA	Less complex the TSS	Some potential locations are not searched.
	Hexagonal	More search points as compare to other methods.	Improve speed rate as compare to other methods.
Linear	CDS	More efficient in picture quality	More complex

III. PROPOSED METHODOLOGY

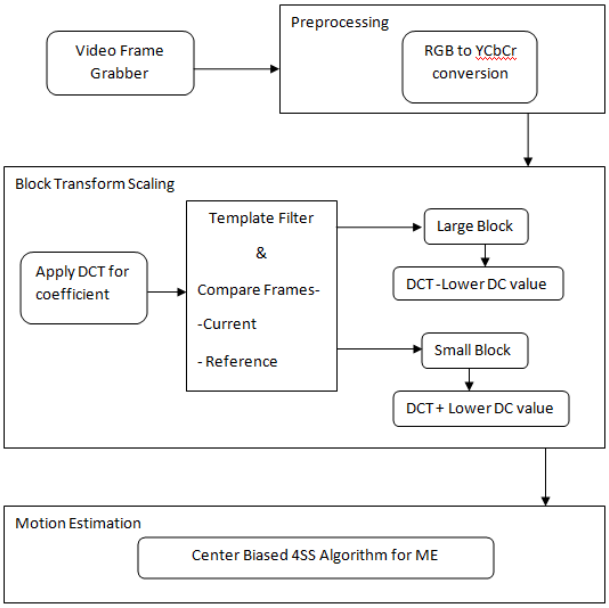

Figure 4. Proposed Architecture

A. *Preprocessing*

1. Color Space Conversion

When frames are obtained from the video, we convert the RGB image into YCbCr image to remove unwanted noise. This is done by using this formula [5]-

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

This conversion into YCbCr is necessary as the RGB image gives only the color information but conversion into YCbCr where, Y represents the luma component or luminance, which gives the light intensity, Cb and Cr are the chroma components which represents blue-difference and red-difference components respectively. The reason for this transformation is that most of the information ends up in the Y image, with much less information in the chroma (Cb and Cr) image.

The human eye is more sensitive to error in brightness (the Y image) than chrominance. After this color conversion, we move to Block transformation Scaling.

B. *Block Transform Scaling*

1. Discrete Cosine Transformation

DCT separates the image/ frame into parts of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transforms (DFT), where it transforms a signal or image from the spatial domain to the frequency domain.DFT consists of

both sine & cosine components which results in frequency leakage therefore it is not suitable for image & video compression. DFT uses the complex computation which makes the hardware implementation complex than DCT. In Discrete Wavelet transforms (DWT) wavelet coefficient are more difficult to construct and it is more complex to implement than DCT and DFT. DCT provides a simple and efficient implementation with real components that is reason we are considering DCT.

In this, Y image is divided into blocks and DCT is applied to each block. This generates coefficient which is a real number. So, there is no need to store complex number. As a frame is in 2D we use 2D formula for DCT is [1]-

$$G_{u,v} = \frac{1}{4} \propto (u) \propto (v) \sum_{x=0}^{7} \sum_{y=0}^{7} gx, y \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right] \quad (2)$$

Where,

u is the horizontal spatial frequency, for the integers $0 \leq u < 8$.

v is the vertical spatial frequency, for the integers $0 \leq v < 8$.

$$\propto (u) = \begin{cases} -\frac{1}{\sqrt{2}}, & \text{if u} = 0 \\ 1, & otherwise \end{cases} \quad (3)$$

Is a normalizing scale factor to make the transformation orthonormal.

$g_{x,y}$ is the pixel value at coordinates (x,y).

$G_{u,v}$ is the DCT coefficient at coordinates (u, v).

We can apply the above formula by computing 1D coefficient for each row and then by using this row-coefficients to compute the coefficients of each column (using the 1D forward transform). If each block size is of 8x8 then DCT produces 64 coefficients. These DCT coefficients affect every pixel value.

G(0,0) define DC & AC components. DC value represents the average of the block, and AC value describes progressively finer details.

2. Template Filter

By using Template filters of 8x8, 16x16, 8x16, 16x8 we are able to find the size of reference and current frame. Every previous frame acts as the reference frame for the current frame in the video sequence.

3. Scaling

By exploiting the fact that human eye is more sensible for lower frequency component i.e AC than higher frequency component i.e DC, we can add or subtract the lower DC value from the block. Doing so does not affect the visibility of an image.

After, finding the DCT coefficient we perform scaling over it by comparing the reference and current frame-

1) If the size of current frame is smaller than the reference frame, i.e, the reference frame is of 16x16 and current frame is of 8x8 then-

DCT + lower DC value (4)

2) If the size of current frame is larger than the reference frame, i.e, reference frame is of 8x8 and current frame is of 16x16then-

DCT- lower DC value (5)

As, we know that not one algorithm is suitable for various types of videos. Gradient search is more suitable for slow videos, Full search is mainly used for non-real videos and 4SS is more suitable for fast videos. So, by scaling we provide a uniformity to the blocks of the frames itself which removes the above dependency on a particular algorithm and also improves the PSNR and Estimation time of the ME algorithm.

C. *Motion Estimation*

1. Four-Step Search

Four Step Search (FSS/4SS). There as on of selecting this ME algorithm out of the all other algorithms is that 4SS is more efficient for detecting the small motion, which are developed for real time videos and it also gives the optimal performance in worst case.

4SS starts with evaluating the distortion in the central block and eight blocks around it, at an initial distance of two pixels. The best matched is calculated and eight new neighbors are selected around the best matched block – also at a distance of two pixels. Finally, the previous best matched block is used to explore the eight blocks around it at a distance of one pixel, and return the best of them [7][9][11]

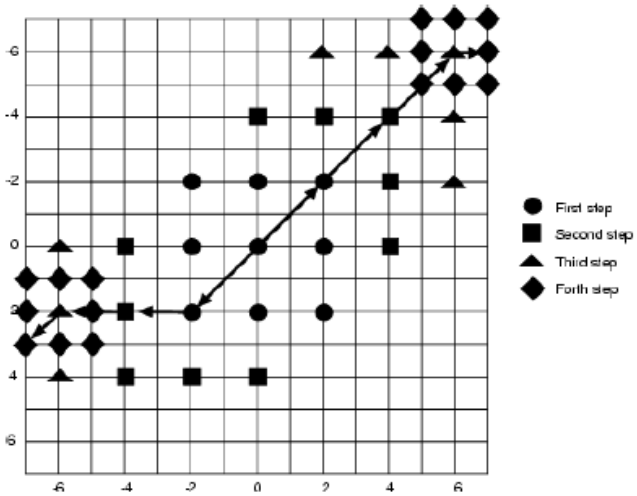

Figure 5.Two different search paths of FSS

Matching Criteria: A cost function is used for matching one macro block with another. The output of this cost function gives a numerical value for the amount of mismatch between the macro blocks that are compared.

The macro block which results in the least cost is the one that matches best for the current block [10]. A large number of cost functions are available, out of that one of the most popular criteria is Sum of absolute Difference (SAD)

$$SAD(k,l) = \sum_{i=1}^{16} \sum_{j=1}^{16} Pt(i,j) - Pt - 1(i+k, j+1) \quad (6)$$

Where (k, l) is the location in the search window, Pt(i, j) is a pixel at (i, j) in the current frame, and Pt−1(i, j) is a pixel in the previous frame. When the value SAD (k,l) is minimum,(k,l) is the motion vector of the macro block [12].

IV. RESULT ANALYSIS

By using video grabber we grab the video for extracting frames from the video and then each pixel of the frame is zoomed into 16x16 near by window.

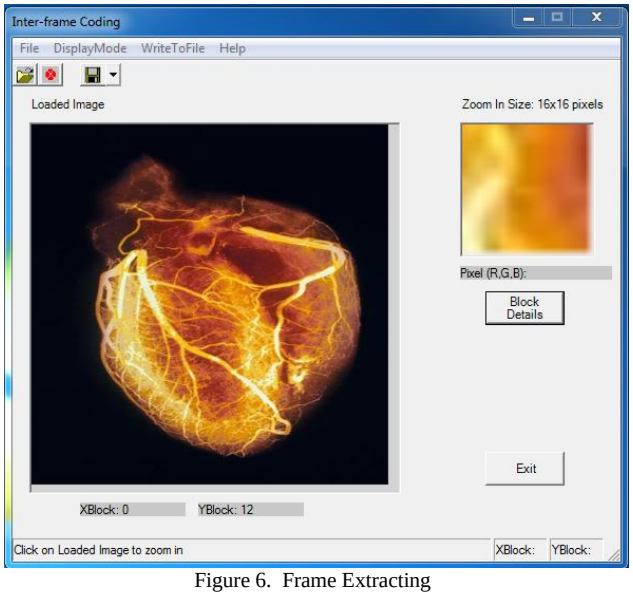

Figure 6. Frame Extracting

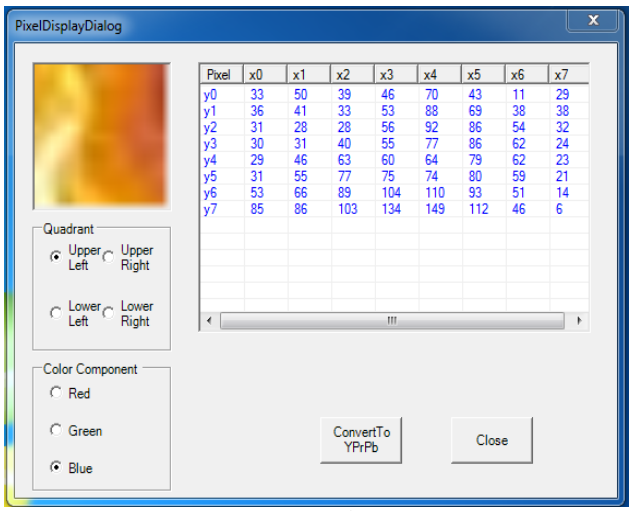

Figure 7. RGB information

In above mentioned figure, we calculate the block details of 16x16 window which is divide into 4 quadrants: upper left, upper right, lower left & lower right. It gives the color component in the RGB form of one quadrant out of 4 quadrants.

Then we do the color space conversion which is RGB to YCbCr. Most of the information is in Y component so we divide Y component into 4 quadrants.

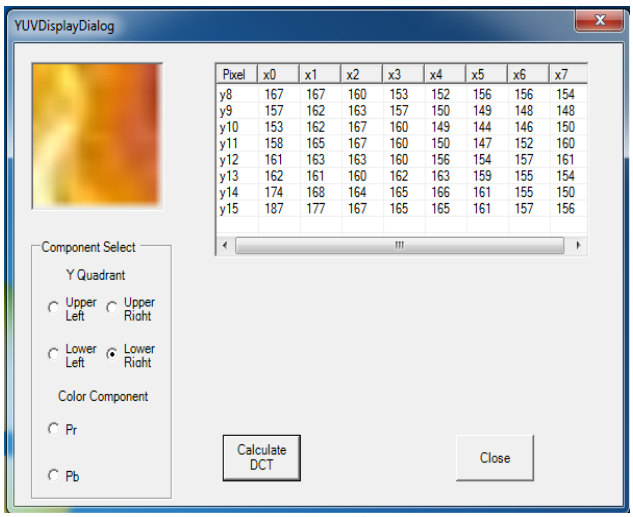

Figure 8. YCbCr information

Then we calculate the DCT coefficients.

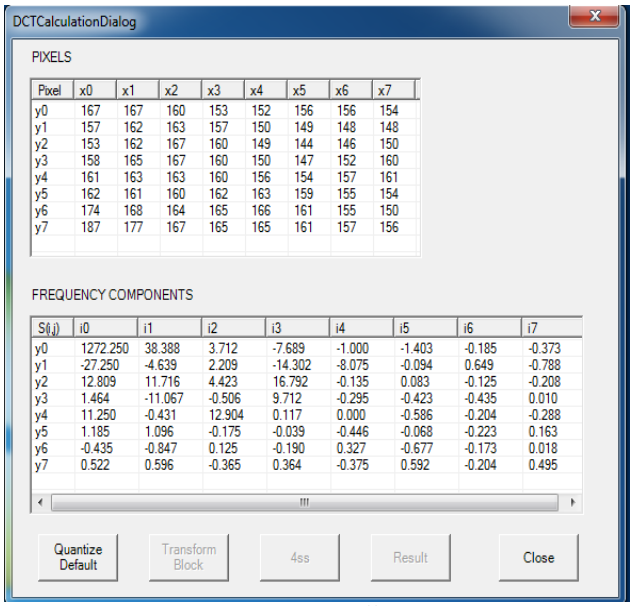

Figure 9. DCT coefficient

The preprocessing and applying of the DCT for finding coefficient is done in all the above mentioned steps. In future we will concentrate on filtering, Scaling and on 4SS for Motion Estimation.

V. CONCLUSION

Motion estimation is a problem faced by the world of artificial intelligence as well as computing. Still today there is no particular algorithm that is well suited

for varying requirements of motion analysis in applications like robotic vision and sensible video processing. Among all the motion estimation techniques, the block matching received very much attention by researcher because of their simplicity and regularity. In this paper, we have presented the comparison of various BMA and concluded that 4SS is the best matching motion estimation algorithm that achieves best tradeoff between search speed and reconstructed picture and also 4SS are developed for real time videos.

The Block Transform Scaling technique exploits the fact that human eyes are more sensible to the AC component of an image than DC component. So, by performing scaling over the blocks of frames we can remove the dependency of videos to the particular algorithm and also may improves the PSNR and Estimation Time.

REFERENCES

[1] M. Marzougui, A. Zoghlami, M. Atri and R. Tourki, "Preliminary Study of Block Matching Algorithms for Wavelet-based t+2D Video Coding," 10th IEEE International Multi-Conference on Systems, Signals & Devices, page no. 1-6, March 2013

[2] P.C. Shenolikar and S.P. Narote, "Different Approaches for Motion Estimation," IEEE International Conference on Control, Automation, Communication and Energy Conservation, page no. 1-4 2009-

[3] Fenta Adnew Mogus , Xinying Liu and Lei Wang, "Evaluation of the Performance of Motion Estimation Algorithms in Video Coding," 2nd IEEE International Conference on Information Science and Engineering, page no. 3693-3696, 2010

[4] Cognizant 20-20 insights, "Using compression techniques to streamline image and video storage and retrieval" May 2014

[5] Hussain Ahmed Choudhury and MonjulSaikia, "Comparative Study of Block Matching Algorithm for Motion Estimation," International Journal of Advanced Computational Engineering and Networking, ISSN:2320-2106, Volume-1, Issue-10, Dec-2013, Arunachal Pradesh, India

[6] Peter Kuhn, "Algorithms, Complexity analysis and VLSI architecture for MPEG-4 motion estimation," DOI: 10.1007/978-1-4757-4474-3_2, Springer US, pp 17-60, 1999

[7] L.C. Manikandan and Dr. R. K. Selvakumar, "A New Survey on Block Matching Algorithms in Video Coding," International Journal of Engineering Research ISSN:23196890, Vol. No.3, Issue No.2, page no. 121-125, 01 Feb. 2014

[8] Ms. Bhavina Patel, Dr. R.V. Kshirsagar and Dr. Vilas Nitnaware, "Review and comparative study of Motion estimation techniques to reduce complexity in video compression," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, ISSN (Print): 2320 – 3765, ISSN (Online): 2278 – 8875, Vol. 2, Issue 8, August 2013

[9] Maria Santamaria and Maria Trujillo, "A Comparison of Block-Matching Motion Estimation Algorithms," IEEE Computing Congress (CCC) 7th Colombians, page no. 1-6, 2012

[10] Jobin T Philip, BinoshiSamuvel, Pradeesh K &Nimmi N K, "A Comparative Study of Block Matching and Optical Flow Motion Estimation Algorithms," IEEE Annual International Conference on Magnetics, Machines and Drive, page no. 1-6, 2014

[11] Ruchi Jain and TruptiBaraskar, "The Comparative Analysis of Various Approaches used in Motion Estimation," International Journal of Advance Foundation and Research in Computer (IJAFRC), ISSN: 2348-4853, Volume 1, Issue 12, December 2014