

Performance Analysis Of Predictive Motion Field Segmentation And Overlapped Block Matching For Video Compression

Kalyani N. Pampattiwar

M.E. Student

*Computer Department,
Datta Meghe College Of Engg,
Airoli, Navi Mumbai-400 708*

Dr. Prof. S. D. Sawarkar

Principial

*Computer Department,
Datta Meghe College Of Engg,
Airoli, Navi Mumbai-400 708*

Abstract

Block matching techniques are used for motion estimation in video compression. The approach used here focuses on two block matching techniques namely predictive motion field segmentation and overlapped block matching. The first method improves motion compensation along boundaries of moving objects, while second one removes the blocking artifacts. Motion estimation and compensation are used to reduce temporal redundancies. Motion compensated prediction is then applied to spatial model where the Discrete Cosine Transform (DCT) is then applied to encode the motion-compensated prediction difference. This reduces the spatial redundancy. The output of the spatial model is a set of quantized transform coefficients. The quantized DCT coefficients, motion vectors are then entropy coded using variable-length codes (VLCs) which will removes the statistical redundancy. The video decoder reconstructs a video frame from the compressed bit stream. This paper intends to compare the performance of both methods with respect to various parameters such as speed , peak-signal-to-noise-ratio (PSNR), MSE.

Keywords- *Block matching, Motion estimation, peak-signal-to-noise-ratio(PSNR), Mean Square Error (MSE), Video compression*

I Introduction

In this era of multimedia there is a wide spread of Internet, video storage on CD or DVD. So, to reduce the space on these devices efficient video compression methods are required. In this work the flow for entire compression and decompression process is same as that of standard MPEG process but the difference is in case of temporal model where the two block matching techniques predictive motion field segmentation and overlapped block matching are applied for motion estimation. Motion estimation techniques form the core of video compression and video processing

applications. Motion estimation extracts motion information from the video sequence in the form of motion vector. Motion information is used in video compression to find best matching block in reference frame to calculate low energy residue. Block Matching Algorithm is the most popular motion estimation algorithm. Block Matching (BM) is a very important stage in the video compression, and it provides an effective way to estimate an object's motion from time varying image sequences.[1][10]

Block matching techniques match blocks from the current frame with blocks from a reference frame. The displacement in block location from the current frame to the location in the reference frame is the motion vector. Block matching techniques can be divided into three main components as shown in Figure 1: block determination, search method, and matching criteria.

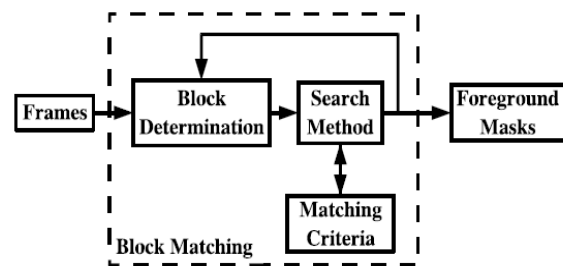


Fig 1: Block Matching Flowchart

The first component, block determination, specifies the position and size of blocks in the current frame, the start location of the search in the reference frame, and the scale of the blocks. We focus on fixed size, disjoint blocks spanning the frame, with initial start location at the corresponding location of the block in the reference frame. In tracking, a predictive method may be used to improve the start location of the search.[4]

The search method is the second component, specifying where to look for candidate blocks in the reference frame. A fully exhaustive search consists of

searching every possible candidate block in the reference frame.

The third component is the matching criteria. The matching criteria is a similarity metric to determine the best match among the candidate blocks. Best match can be decided with the help of parameters such as MSE, PSNR, MAD.

The motion vectors are fed to the block determination to implement multiresolution blocks. A coarse to fine resolution of the blocks is generated. The start location of the search at each resolution is the location of the best match (motion vector) from the previous coarser resolution.

II Literature Review

Peak-Signal-to-Noise-Ratio (PSNR) given by equation (i) characterizes the motion compensated image that is created by using motion vectors and macro clocks from the reference frame.[7] [8][15]

$$PSNR = 10 \log_{10} \left[\frac{(\text{Peak to peak value of original data})^2}{MSE} \right] \quad (i)$$

A. Exhaustive Search

This algorithm, also known as Full Search, is the most Computationally expensive block matching algorithm of all This algorithm calculates the cost function at each possible location in the search window. As a result of which it finds the best possible match and gives the highest PSNR amongst any block matching algorithm. Fast block matching algorithms try to achieve the same PSNR doing as little computation as possible . The obvious disadvantage to ES is that the larger the search window gets the more computations it requires.

B. Three Step Search(TSS)

This is one of the earliest attempts at fast block matching algorithms and dates back to mid 1980s. The general idea is represented in Figure 2. It starts with the search location at the centre and sets the 'step size' $S = 4$, for a usual search parameter value of 7. It then searches at eight locations $\pm S$ pixels around location (0,0). From these nine locations searched so far it picks the one giving least cost and makes it the new search origin.[3] 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 least cost function and the macro block at that location is the best match. The calculated motion vector is then saved for transmission. It gives a flat reduction in computation by a factor of 9. So that for $p = 7$, ES will compute cost

for 225 macro blocks whereas TSS computes cost for 25 macro blocks.

The idea behind TSS is that the error surface due to motion in every macro block is unimodal. A unimodal surface is a bowl shaped surface such that the weights generated by the cost function increase monotonically from the global minimum.

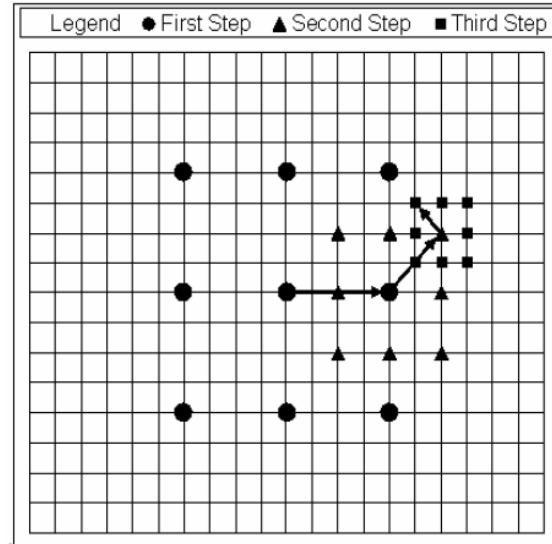


Fig 2: Three Step Search Procedure. The motion vector is (5,-3)

C. Adaptive Rood Pattern Search(ARPS)

ARPS [5] algorithm makes use of the fact that the general motion in a frame is usually coherent, i.e. if the macro blocks around the current macro block moved in a particular direction then there is a high probability that the current macro block will also have a similar motion vector. This algorithm uses the motion vector of the macro block to its immediate left to predict its own motion vector. An example is shown in Fig. 3.

The predicted motion vector points to (3, -2). In addition to checking the location pointed by the predicted motion vector, it also checks at a rood pattern distributed points, as shown in Fig 3, where they are at a step size of $S = \text{Max}(|X|, |Y|)$. X and Y are the x-coordinate and y-coordinate of the predicted motion vector. This rood pattern search is always the first step. It directly puts the search in an area where there is a high probability of finding a good matching block.[6] The point that has the least weight becomes the origin for subsequent search steps, and the search pattern is changed to Small Diamond Search Pattern (SDSP). The procedure keeps on doing SDSP until least weighted point is found to be at the centre of the SDSP. [2] A further small improvement in the algorithm can be to check for Zero Motion rejudgment [5], using which the search is stopped half way if the least weighted point is already at the centre of the rood

pattern.

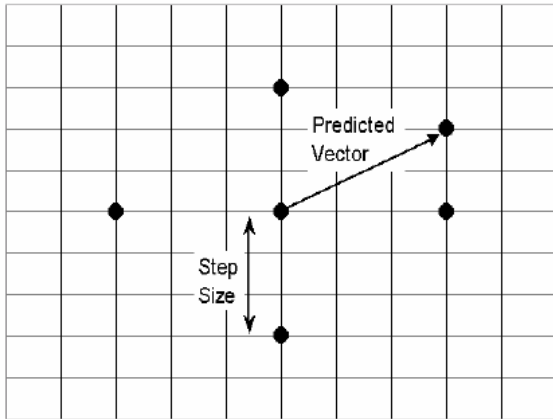


Fig 3. Adaptive Rood Pattern: The predicted motion vector is (3,-2), step size S=3

III. PROPOSED METHODOLOGY

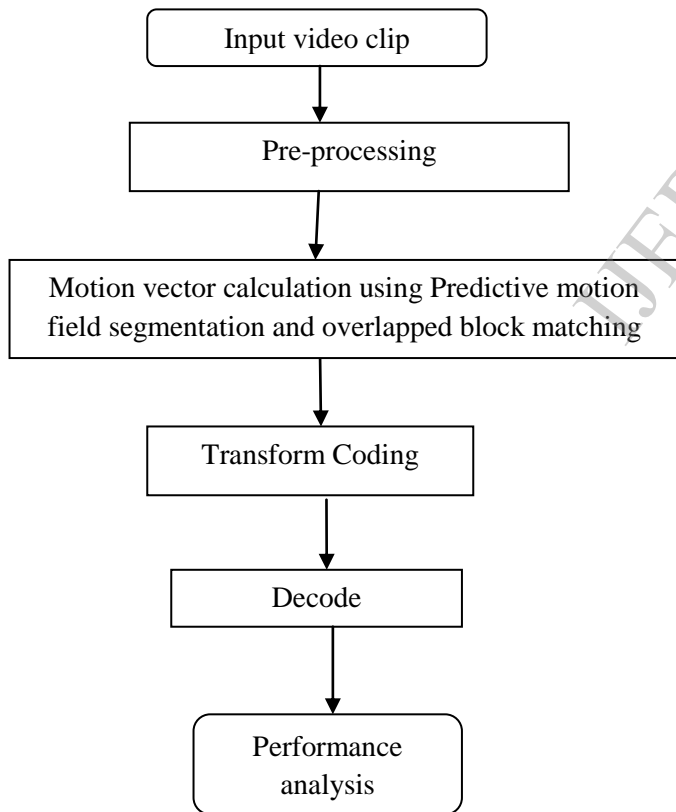


Fig 4: Flow diagram for video compression using predictive motion field segmentation and overlapped block matching

A) Input Video Clip

The input should be taken as a video clip in an uncompressed video format. Here .y4m video clips are used.

B) Pre-processing

Source video is taken in uncompressed format. For random access and highly efficient coding MPEG suggests the file pattern in the format 'IPPPP'. I-pictures are coded without reference to the previous picture and p-pictures are predictively coded. The color format is converted from RGB to YCbCr. Frames are segmented into macroblocks of size 16x16. Every macroblock is either interframe or intraframe coded.

C) Motion Vector calculation

Motion vector can be obtained by using the following two proposed techniques.

1) Predictive Motion Field Segmentation

This method improves motion compensation along boundaries of moving objects by segmenting the motion field of previous frames of the sequence, and using the segmentation to predict the location of motion-field discontinuities in the current frame.

A common problem with block-based motion compensation methods is that blocks located on boundaries of moving objects are not compensated accurately. Blocks on these boundaries contain objects moving in at least two different directions, and a single motion vector applied to the entire block inevitably compensates one of the objects incorrectly. Fig. 5 illustrates a typical artifact caused by this problem in a section of a frame containing a ball falling against a still background. The motion of the ball can be deduced from the top two images which magnify a 30 x 40 pixel region of frames I^n and I^{n-1} . The bottom image is the corresponding region of the motion-compensated prediction I^{n*} , using motion vectors computed by full-search block matching (using 16 x 16 blocks). A large portion of the lower right side of the ball has been compensated incorrectly, because it is located in a block which has been assigned the motion vector (0,0) corresponding to the background. In frames with multiple moving objects, artifacts like the one seen on the ball account for a significant percentage of the bit rate used to transmit the DFD, and overall picture quality is degraded. The problem experienced by block-based methods at object boundaries can be viewed as a problem in motion field interpolation. The decoder receives a decimated motion field in the form of block motion vectors, and it needs to compute an interpolated motion field which it uses for motion compensation.

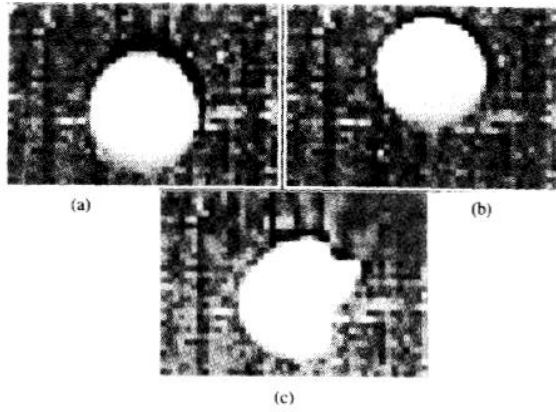


Fig 5: Boundary problems of block-based motion compensation (16x16 blocks) (a) current frame I^n (b) previous frame I^{n-1} (c) Motion compensated frame I^n

This technique produces higher resolution motion field estimates without a significant increase in the bit rate used for describing the motion field. The higher resolution motion field allows more accurate motion compensation along boundaries of moving objects, resulting in more efficient coding and reducing noticeable artifacts at those boundaries in low-bit-rate coding systems.

This method can improve the performance of block-based methods by relaxing the constraint of constant translational motion at blocks located on boundaries of moving objects. This improvement can be achieved by segmenting the motion field of these blocks and applying different motion vectors to each segment. To avoid transmitting additional motion vectors, the vector applied to each segmented region is selected from a set of neighbouring motion vectors. The proposed method allows us to reconstruct discontinuities in the motion field at pixel resolution while transmitting the same number of motion vectors as standard block-based methods. At the decoder, the proposed method needs both the set of block motion vectors and the computed segmentation to perform motion compensation. In order to reduce the extra bit rate to transmit this segmentation to the decoder the segmentation can be done at the decoder based on a segmentation computed from previously decoded frames. [12]

2) Overlapped Block Matching

The idea is to relax the restriction of a nonoverlapped block partition imposed in the blockbased model in block matching. After the nonoverlapped, fixed size, small rectangular block partition has been made, each block is enlarged along all four directions from the centre of the block. Refer to Figure 6. [14]

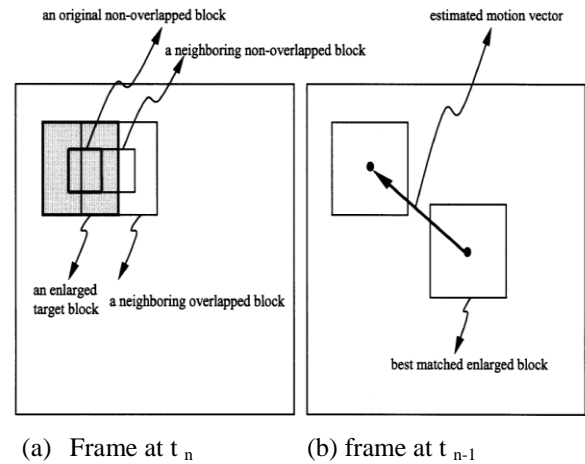


Fig 6: Overlapped Block Matching

Both motion estimation (block matching) and motion-compensated prediction are conducted in the same manner as that in block matching except for the inclusion of a window function. That is, a 2-D window function is utilized in order to maintain an appropriate quantitative level along the overlapped portion. The window function decays towards the boundaries. Consider one of the enlarged, overlapped original (also known as target) blocks, $T(x, y)$, with a dimension of $l \times l$. Assume that a vector v_i is one of the candidate displacement vectors under consideration. The predicted version of the target block with v_i is denoted by $v_i, P_{v_i}(x, y)$. Thus, the prediction error with $v_i, E_{v_i}(x, y)$ can be calculated according to the following Equation[16]

$$E_{v_i}(x, y) = P_{v_i}(x, y) - T(x, y) \quad (ii)$$

The window function $W(x, y)$ is applied at this stage as follows, resulting in a window-operated prediction error with v_i, WE_{v_i} .

$$WE_{v_i}(x, y) = E_{v_i}(x, y) \times W(x, y) \quad (iii)$$

Assume that the MAD is used as the matching criterion. It can then be determined as usual by using the window-operated prediction error $WE_{v_i}(x, y)$. That is,

$$MAD = \frac{1}{l^2} \sum_{x=1}^l \sum_{y=1}^l |WE_{v_i}(x, y)| \quad (iv)$$

The best match, which corresponds to the minimum MAD, produces the displacement vector v . In motion-compensated prediction, the predicted version of the enlarged target block, $P_v(x, y)$ is derived from the frame at t_{i-1} by using estimated vector v . The same

window function $W(x, y)$ is used to generate the final window-operated predicted version of the target block. That is,

$$WP_v(x, y) = P_v(x, y) \times W(x, y) \quad (v)$$

A block size of 16×16 was used for conventional block matching, while a block size of 32×32 was employed for the proposed overlapped block matching. The maximum displacement range d was taken as $d = 15$, i.e., from -15 to $+15$ in both the horizontal and vertical directions. It was observed that the blocking edges originally existing in the prediction error signal with conventional block matching was largely removed with the proposed overlapped block matching technique.[13]

D) Transform Coding

1) Apply DCT

The fundamental operation performed by DCT is to transform the space domain representation of an image to a spatial frequency domain. DCT operates on X , a block of $N \times N$ samples and creates Y , an $N \times N$ block of coefficients. The DCT of $N \times N$ sample block is given by :

$$Y = AXA^T$$

The elements of A are

$$A_{ij} = \frac{C_i \cos((2j+1) i\pi)}{2N} \quad (vi)$$

Where $C_i = 1/2$ ($i=0$), $C_i = (2/N)^{1/2}$ ($i>0$)

2) Quantization

Human eye is insensitive to high frequency content in an image; therefore it can be removed via quantization process. DCT coefficients prior to quantization are divided by weighting matrix. Weighted coefficients are then quantized by the quantization step size. Higher orders of DCT coefficients are quantized with coarse quantization step sizes than the lower frequency ones.

3) Entropy Encoding

The quantizer indices as well as motion vectors are compressed by the entropy encoder. This removes statistical redundancy in the data and produces a compressed bit stream or file that may be stored and/or transmitted.

A compressed sequence consists of coded motion vector parameters, coded residual coefficients and header information.

E) Decode

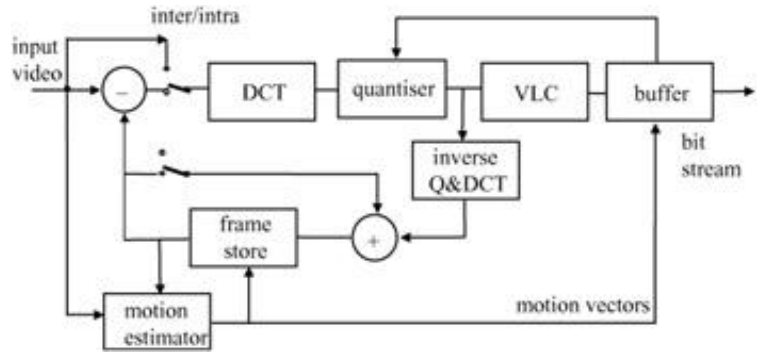


Fig 7: A generic interframe predictive coder

The video decoder reconstructs a video frame from the compressed bit stream. The coefficients and motion vectors are decoded by an entropy decoder after which the spatial model is decoded to reconstruct a version of the residual frame. The decoder uses the motion vector parameters, together with one or more previously decoded frames, to create a prediction of the current frame and frame itself is reconstructed by adding the residual frame to this prediction.[9]

F) Performance Analysis

In this the performance of the predictive motion field segmentation and overlapped block matching techniques are compared with respect to various parameters such as MSE, PSNR, NSP and speed up ratio which is given by following formulas.

1) MSE (Mean Square error)

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 Square Error (MSE) given by equation

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (vii)$$

where N is the side of the macro block, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively.

2) PSNR (Peak Signal to Noise Ratio)

Peak-Signal-to-Noise-Ratio (PSNR) characterizes the motion compensated image that is created by using

motion vectors and macro blocks from the reference frame.

$$PSNR = 10 \log_{10} \left[\frac{(\text{Peak to peak value of original data})^2}{MSE} \right]$$

3) Number of Search points

A search window with a maximum motion of $\pm W$ in both the horizontal and vertical directions will require $(2W + 1)^2$ candidate search points for each block.

4) Speed up ratio

Predictive motion field segmentation and overlapped block matching techniques will be compared as how much they speed up the compression process in terms of percentage with respect to each other

IV. EXPECTED RESULTS

The work is under progress. This work illustrates the comparison of both methods with respect to various parameters such as PSNR, Compression ratio, number of search points. Comparison w.r.t. Compression ratio and PSNR for bus sequence is shown in figure 8 and figure 9 respectively.

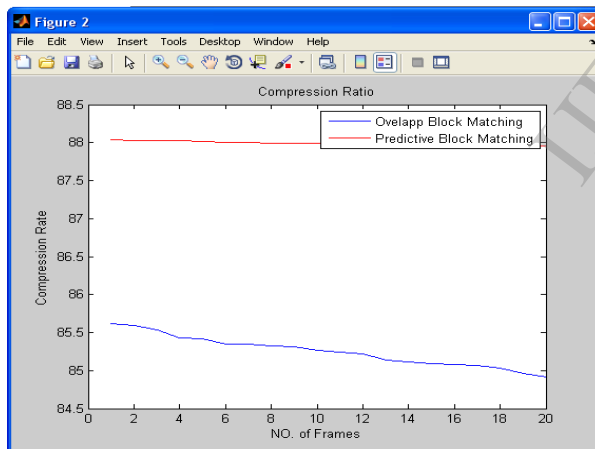


Fig 8: Comparison w.r.t. Compression ratio

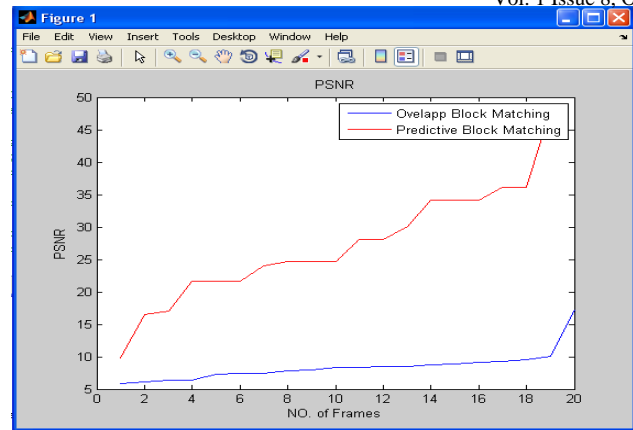


Fig 9: Comparison w.r.t. PSNR

V. CONCLUSION

A video compression method is proposed which uses Predictive motion field segmentation and Overlapped block matching technique for calculation of motion vectors. Pre-processing is followed by motion estimation using above techniques. Then the Discrete Wavelet Transform is applied to encode the motion-compensated prediction difference for reducing spatial redundancy and results are quantized. The quantized DCT coefficients, motion vectors are then entropy coded using variable length codes such as Huffman code and Run length code. The video decoder then reconstructs the video frame from compressed bit stream. Finally the performance analysis of both block matching methods will be carried out.

Thus Predictive motion field segmentation may improve motion compensation along boundaries of moving objects by segmenting the motion field of previous frames of the sequence, and using the segmentation to predict the location of motion-field discontinuities in the current frame. The study says that by using Predictive motion field segmentation and Overlapped block matching the quality of decoded image can be improved and blocking artifacts can be reduced.

REFERENCES

- [1] Nicole S. Love and Chandrika Kamath, 'An Empirical Study of Block Matching Techniques for the Detection of Moving Objects'
- [2] 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.
- [3] Li, R., Zeng, B., and Liou, M.L.: 'A new three-step search algorithm for block motion estimation', *IEEE Trans. Circuits Syst. Video Technol.*, 1994, 3, (4), pp. 438-442

- [4] Lu, J., and Liou, M.L.: 'A simple and efficient search algorithm for block matching motion estimation', *IEEE Trans. Circuits Syst. Video Technol.*, 1997, 7, (2), pp. 429–433
- [5] Nie, Y., and Ma, K.-K.: 'Adaptive rood pattern search for fast block matching motion estimation', *IEEE Trans. Image Process.*, 2002, 11, (12), pp. 1442–1448
- [6] Ma, K.-K., and Qiu, G.: 'Adaptive rood pattern search for fast block matching motion estimation in JVT=H.26L'. Proc. 2003 Int. Symp. On Circuits and Systems, May 2003, Vol. 2, pp. 25–28
- [7] B.-G. Kim, S.-T. Kim, S.-K. Song and P.-S. Mah 'Fast-adaptive rood pattern search for block motion estimation' *Electronics Letters online no: 20051507 doi: 10.1049/el:20051507*
- [8] Milind Phadtare, 'Motion estimation techniques in video processing', *Electronic engg. Times India Aug-2007* eetindia.com
- [9] Mohammed Ghanbari, 'Video coding and introduction to standard codecs', *IEEE telecommunications series 42*
- [10] Iain E.G. Richardson, 'H.264 and MPEG4 video compression', *Wiley Publication*.
- [11] Manoranjan Paul PhD research project proposal on New approach to Very low bit rate coding systems.
- [12] M.T.Orchard, 'Predictive Motion field segmentation for image sequence coding', *circuit and systems for video technology* vol 3.Issue-1
- [13] Jonathan K. Su, and Russell M. Mersereau, 'Motion Estimation Methods for Overlapped Block Motion Compensation', *Fellow, IEEE*
- [14] Michael T. Orchard and Gary J. Sullivan, 'Overlapped Block Motion Compensation: An Estimation-Theoretic Approach', *IEEE transactions on image processing*, Vol. 3, No. 5, Septembek 1994 **693**
- [15] Aroh Barjatya, 'Block Matching Algorithms For Motion Estimation' *IEEE*.
- [16] Satoshi Nogaki and Mutsumi Ohta, 'An overlapped block motion compensation For high quality motion picture coding. *C&C Systems*.