# The Comparative Analysis of Various Approaches Used In Motion Estimation.

Ruchi Jain, Trupti Baraskar
Research Scholar Department of Information Technology
Assistant Professor, Department of Information Technology
Maharashtra Institute of Technology, Pune University, Pune 411038
rj.ruchijain23@gmail.com,trupti.baraskar@mitpune.edu.in

**A B S T R A C T**

**Motion estimation has become a serious problem in many video applications. Motion estimation has been traditionally used in the video encoding only but nowadays researchers from different fields are turning towards motion estimation to solve various real life problems in their respective fields. Thus, the development of efficient fast motion estimation is the basic requirement of the video encoder. Among all ME algorithms BMA is commonly used because of its simplicity and efficiency. BMA are used to reduce the memory requirements for videos. Motivated by the specific requirements of motion estimation, a variety of algorithms have been developed. The aim of this paper is to give some constructive information to the succeeding researchers for fast ME algorithms. In this paper, the comparative performance of 5 different block matching algorithms are presented and compared based on the parameters of PSNR and (NC).**

**Index Terms: Motion estimation, Block matching algorithms, PSNR, MAD, MSE, Number of computations.**

## I. INTRODUCTION

With the advancement of internet and multimedia systems, digital networking and video storage systems have been gaining a lot of popularity. 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]. The most computationally expensive and resource hungry operation in the entire compression process is motion estimation. Motion estimation is defined as finding the best Motion Vector (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. Motion estimation 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].

This paper is organized as follows: In section 2, Motion estimation is explained in detail. Section 3, gives the brief idea about motion estimation algorithms and its classification. The detail of block matching algorithm is explained in section 4. Section 5 contains the results and analysis. And finally, section 6 concludes the paper followed by references.

## II. MOTION ESTIMATION

### A. Need for effective storage

  

Video compression is needed for effective storage as well as for efficient data transfer in the video system. There are two types of video compression techniques, lossless compression & lossy compression. In lossless compression, the image reconstructed is same as the original image. So, there is no loss of information occurs. Whereas in lossy compression, there is some loss of information is tolerable [2].

### B. Intra-frame & Inter-frame compression

A video can be viewed as a sequences of images. Therefore, video compression can be accomplished through image compression with additional processes like inter-frame compression. In image coded terminology, an image is known as a frame.

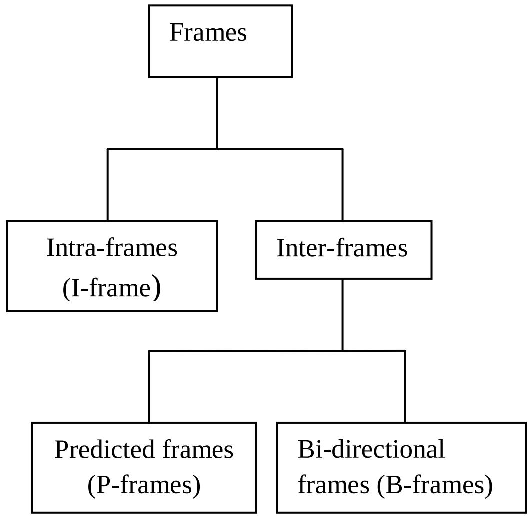


**Figure 1. Classification of frames**

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 consider 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, thus aiming to reduce spatial and temporal redundancies. This also increases the efficient of data compression [4].

According to the predications Inter-frame is further divided into P-frames and B-frames. In encoding frame k, if we are using 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. In the reverse case, if we are using future frame for reference to encode current frame then we are predicting the past going ahead in time so this process is called Backward Prediction or Forward Motion Estimation. 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].

## III. MOTION ESTIMATION ALGORITHMS

Motion estimation algorithms have 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].

## A. Classification

Motion estimation algorithms can be categorized into time-domain and frequency-domain algorithms.

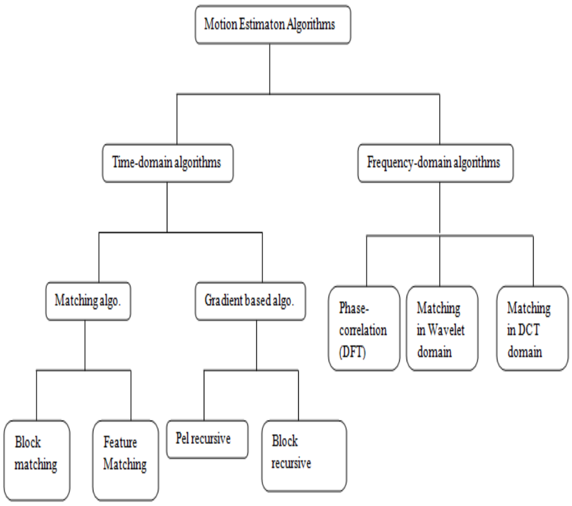


**Figure 2. Classification of motion estimation algorithms**

## B. Matching algorithms

1. **Block-matching:** 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. When we compare 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 that, instead of matching pixel to pixel value, a region or block matching is done on the frames. The matching is done 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].

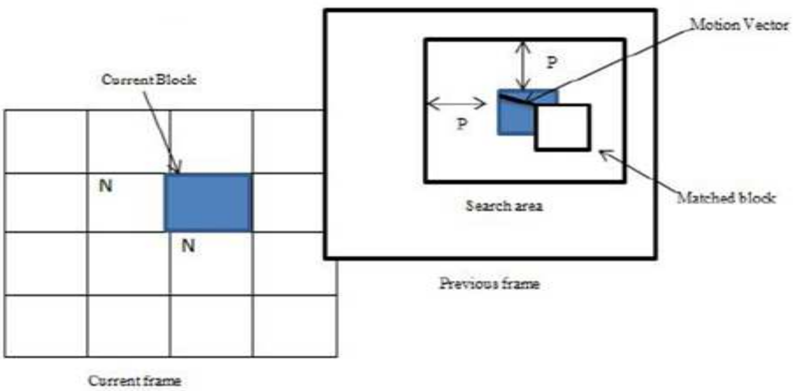


**Figure 3. Basic idea of block motion estimation**

2. **Feature- matching:** In this once we have extracted features and their descriptor from two or more images, the next step is to establish some preliminary feature matches between these images. In this technique, we can divide the problem into two separate components:

  

- Firstly, it selects a matching strategy, which determines which correspondences are passed on to the next stage for further processing.
- The second is to devise efficient data structure and algorithms to perform this matching as quickly as possible.

The simplest way to find all corresponding feature points is to compare all features against all other features in each pair of potentially matching images. Unfortunately, this is a quadratic in the number of extracted features which makes it impractical for most applications [7].

## IV. BLOCK MATCHING ALGORITHM

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.

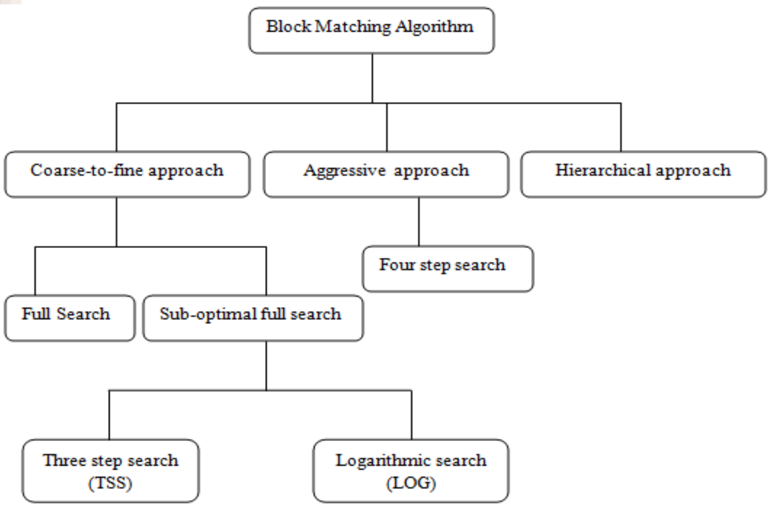


**Figure 4. Classification of block matching algorithm**

### A. 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 [8].

### B. 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 neighbor 8 points at the end of '+' & 'x' both. And which is minimum out of these 8 points, MV is set according to that [5].

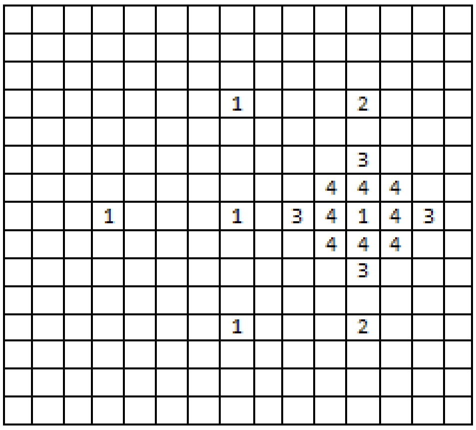

**Figure 5. 2D logarithmic search**

### C. 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 neighbors are selected around it, at half of the initial distance. This process is repeated until the distance is equal to one [5][8].

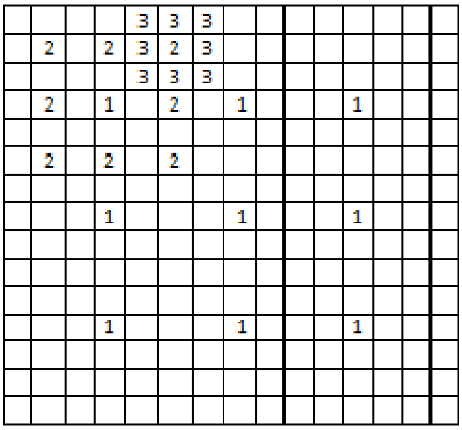

**Figure 6. TSS search**

### D. 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][9].

### E. Four-step search

 

The Four- Step Search (FSS) 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 [8][10].

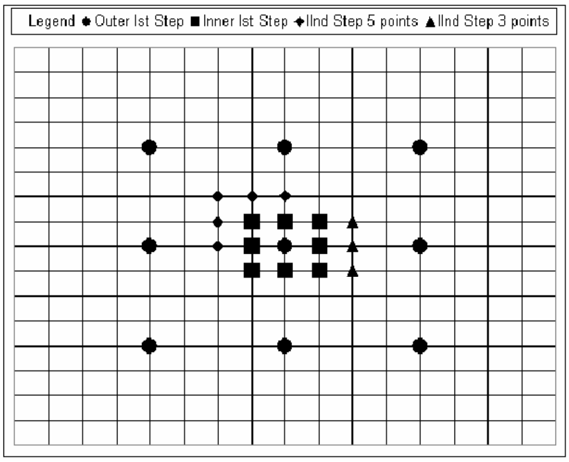


**Figure 7.NTSS search**

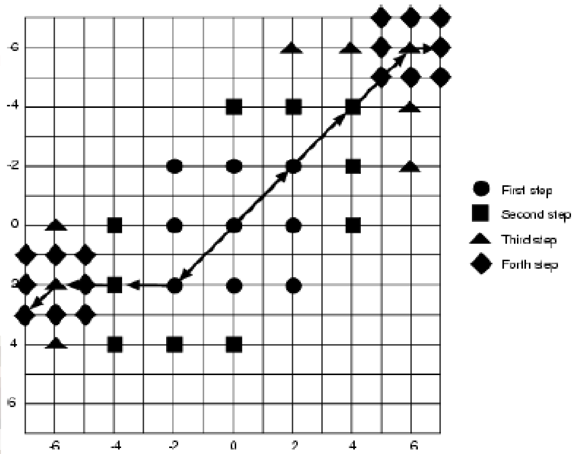


**Figure 8.Two different search paths of FSS**

## V. RESULT ANALYSIS

### A. Comparative parameters

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. A large number of cost functions are available, of which the most popular are Mean Absolute Difference (MAD) and Mean Squared Error (MSE) [11]:

 

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

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

The Peak Signal to Noise Ratio (PSNR) is used in the field of image processing for the determination of the quality of an image. This is introduced due to image compression and this is the ratio of the Power of the Signal to the Power of the Noise.

$$\mathrm{PSNR} = 10 \log_{10} \left\{ \frac{(\mathrm{MAX})^2}{\mathrm{MSE}} \right\}$$

Where, MAX represents the maximum pixel value in the image [9].
Number of Computations (NC) per frame determines how many number of points that need to be compared between adjacent frames in order to detect any motion in the video. Lesser the NC the lesser will be the complexity of the algorithm [12].

### B. Comparative Analysis

K. V. Arya and Purnima Parsad [12] have done the performance evaluation of various Motion Estimation techniques such as: FS, TSS AND NTSS using CIPR video sequences of Sun Raster file [13][14][15] in terms of number of computation (NC) and PSNR.

We have conclude from the results obtained by them as: When number of frames are more or either less TSS has less PSNR then FS and NTSS. But, with respect to NC when the number of frames are more NTSS performs better than FS and TSS. But, when number of frames are less again TSS performs better than FS and NTSS. So, TSS is an efficient algorithm which has less NC and less PSNR in an average requirement.

In Table 1 we have analysis various BMA and also give the advantage and disadvantage for these algorithms with their computational complexity of the searching points [10][16].

**Table 1**

| Method | Algorithm | Maximum searching time | Advantages | Disadvantages |
|---|---|---|---|---|
| Non linear | FS | 2*(2P+1)*(2P+1) | Best picture quality and highest PSNR | Very high computational cost |
| | 3SS | $[1+8\mathrm{Log}_2(p+1)]$ | Optimum performance, less complexity, recommended in MPEG2 | can't detect small motion |
| | NTSS | $[1+8\mathrm{Log}_2(p+1)]+8$ | More efficient then TSS for small motion | More complex then TSS |
| | FSS | $[18\log_2(p+1)/4+9]$ | Initial small SS so more efficient for small MV | More complex then TSS |
| | CSA | $[5+4\mathrm{Log}_2(p)]$ | Less complex the TSS | Some potential locations are not searched. |
| | Hexagonal | N=7+3*n+4 | More search points as compare to other methods. | Improve speed rate as compare to other |

 

| | | | | |
|---|---|---|---|---|
| | | | | methods. |
| | | | | |
| linear | CDS | 2*(2*(P+1)) | More efficient in picture quality | More complex |

## VI. CONCLUSION

Motion estimation is a classical 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 the all motion estimation methodologies, the block matching received very much attention by researcher because of their simplicity and regularity. In this paper, the performance of different block matching motion estimation algorithms: FS, 2DLS, TSS, NTSS, FSS are studied and compared with respect to their results of PSNR, MAD/MSE and number of computations.

## VII. REFERENCES

[1] M. Marzougui, A. Zoghlami, M. Atri and R. Tourki, “Preliminary Study of Block Matching Algorithms for Wavelet-based t+2D Video Coding,” $10^{th}$ 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,” $2^{nd}$ 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 Monjul Saikia, “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] Richard Szeliski, “Computer vision: Algorithms and Applications,” Texts in computer science, DOI: 10.1007/978-1-84482-935-0_4, Springer London, pp 181-234, 2011

[8] Maria Santamaria and Maria Trujillo, “A Comparison of Block-Matching Motion Estimation Algorithms,” IEEE Computing Congress (CCC) $7^{th}$ Colombians, page no. 1-6, 2012

[9] Aroh Barjatya, “Block Matching Algorithms for Motion Estimation,” IEEE spring 2004

[10] L.C. Manikandan and Dr. R. K. Selva kumar, “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

[11] Jobin T Philip, Binoshi Samuvel, 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

  

[12] K.V. Arya and Purnima Prasad, “A New Motion Estimation Technique for Video Compression,” IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, page no.1 -5, 2013

[13] CIPR Video sequences available at URL: http://www.cipr.rpi.edu/resource/sequences/sequence01.html

[14] CIPR Video sequences available at URL: http://www.cipr.rpi.edu/resource/sequences/itu601.html

[15] CIPR Video sequences available at URL: http://www.cipr.rpi.edu/resource/sequences/sif.html

[16] 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