

## Various Fast Block Matching Algorithm for Video Shot Boundary Detection

Vanshree Verma, Sr. Asst. Prof. Ravi Mishra

*Department of Electrical and Electronics Engineering, Shri Shankaracharya College of Engineering & Technology, Junwani, Bhilai (C.G.)*

---

**Abstract:** .Various methods have been proposed and developed for video shot boundary detection. Commonly used method for shot detection is Block Matching Algorithm. It is the most commonly used motion estimation technique. Various block matching algorithms have been proposed for shot detection such as Exhaustive Search, Three Step Search, New Three Step Search, Simple and Efficient Search, Four Step Search, Diamond Search, Adaptive Rood Pattern Search. This paper presents the comparative study of various Block Matching Algorithm used for boundary detection in videos.

**Keywords:** Block Matching, Motion Estimation, Motion Vectors, PSNR.

---

### I. Introduction

Recent advantages in digital technology have made many video readily available. Therefore it is necessary to develop efficient and effective methods for indexing and retrieving videos. Video shot boundary detection is the initial and fundamental step of indexing, browsing and retrieval applications. Shot boundary detection is the most basic video segmentation task. It also referred as video temporal segmentation. General objective is to segment a given video sequence into its constituent shots and to identify the different transitions between adjacent shots in the sequence. The first step of any video processing method is to segment the input video into elementary shots. A shot is defined as the continuous frame from start to the end of recording in the camera. It shows continuous image in sequence. There are two types of transitions that can occur between shots, they are abrupt change and gradual transition. Shot detection identify and classify this different shot transitions in the sequence.

An efficient method for fast motion estimation is the basic requirement for shot detection. Commonly used motion estimation technique is Block matching technique. Block matching algorithm proposed a variety of algorithm for motion estimation. They are Full Search (FS), Three-Step Search (TSS), New Three-Step Search (NTSS), Simple and Efficient Search (SES), Four-Step Search (FSS), Diamond Search Algorithm (DS) and Adaptive Rood Pattern Search (ARPS). In this paper, section II gives an introduction to block matching algorithm in detail, section III describes various block matching algorithms, section IV gives experimental results and PSNR comparison of algorithms mentioned above then conclusion and references.

### II. Block Matching Algorithm

For shot boundary detection technique, the motion of the moving objects has to be estimated first. This is called motion estimation. Motion estimation is a technique which determines the motion activity between frames in sequence. Motion activity is one of the motion features which describes the intensity of motion in video sequence. The commonly used motion estimation technique in all the standard video is the block matching algorithm (BMA). BMA find motion vectors by comparing blocks in frames. BMA is a way of locating matching blocks in a sequence of digital video frames for the purposes of motion estimation.

In this typical algorithm, current frame is divided into blocks of  $M \times N$  pixels or square blocks of  $N^2$  pixels. Then, each block undergoes translation only with no scaling or rotation. The blocks in the first frame are compared to the blocks in the second frame. Then, for each current block, we find that best match for the current block in the neighborhood to calculate motion vectors. Motion Vectors can be calculated for each block by observing where each block from the first frame ends up in the second frame.

Block matching divides the current frame into macro blocks that are then compared with reference block and its adjacent neighbors in the previous frame to create a vector that defines the motion in a macro block from one frame to another frame. This movement calculated for all the macro blocks constitutes the motion estimation of the current frame. To find the motion vectors for a current block, a search parameter  $p$  is used. The neighborhood of current block match is constrained up to  $p$  pixels on all four sides of the macro block in the frame. For large motion activity, a larger  $p$  is required, and hence the computational process of motion vectors becomes more expensive. The macro block, for example, is taken as a square of side 16 pixels, and the search parameter  $p$  is 7 pixels. The idea is represented in fig 1.

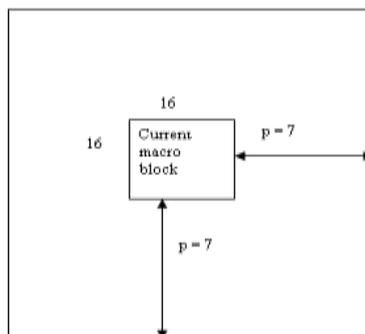


Fig.1. Block Matching a macro block of side 16 pixels and a search parameter  $p$  of size 7 pixels.

The matching of one macro block with another is based on the output of a cost function. There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (i). Another cost function is Mean Squared Error (MSE) given by equation (ii).

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

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

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. Peak-Signal-to-Noise-Ratio (PSNR) given by equation (iii) characterizes the motion compensated image that is created by using motion vectors and macro blocks from the reference frame.

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

In the last two decades, a variety of fast BMAs have been proposed and developed for motion estimation. Next section gives an overview of all those algorithms.

### III. Various Block Matching Algorithms

A vast number of fast BMAs have been developed for motion estimation in videos. They are:

1. Exhaustive Search (ES)
2. Three Step Search (TSS)
3. New Three Step Search (NTSS)
4. Simple and Efficient Search (SES)
5. Four Step Search (4SS)
6. Diamond Search (DS)
7. Adaptive Rood Pattern Search (ARPS)

#### 3.1 Exhaustive Search (ES) Algorithm:-

Exhaustive Search Algorithm (ES) is a simple block matching algorithm also known as Full Search Algorithm. This algorithm compares the current block with all macro blocks in the searching area of the reference frame, then we search for a block which is best matched to the current block in the reference frame shown in fig 2. Then motion vector is then calculated by MSE (Mean Squared Error) formula (i). Since we need to check all macro blocks for finding the motion vectors of current block, the calculation cost of motion vectors in this algorithm is very high.

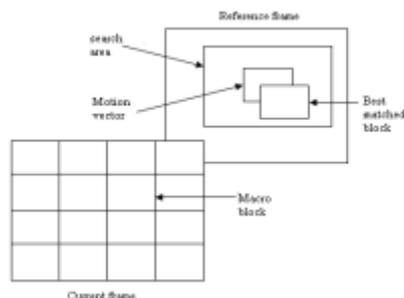


Fig.2. Full Search Algorithm

Another fast block matching algorithm is developed known as Fast Computation of Full Search (FCFS) which can also referred as a type of Full Search Algorithm. This technique reduces the processing time by stopping the computation if the sum absolute different value between the current macro block and each macro blocks in the search area exceeds the previous sum absolute different value.

### 3.2 Three Step Search (TSS) Algorithm:-

This fast block matching algorithm proposed in 1980s and became very popular because this algorithm is very simple and also its performance is robust and optimal. It searches for the best motion vector to make search pattern accurate. The general idea of process is shown in fig 3. The first step consisting of searching the location in the centre and taking the search parameter  $p=7$  to set the step size. It then searches at eight locations around location (0,0). Then considering that searched location that gives least cost, begin a new search. The step size is set to new value and then repeats similar search for more iterations. Then it searches a location which is the best match to the current block and which also gives least cost function. Then it calculates the motion vector for that block and saved it. TSS is efficient for those shot sequence which have large motion activity. But it becomes inefficient for those shot sequence which have small motion estimation.

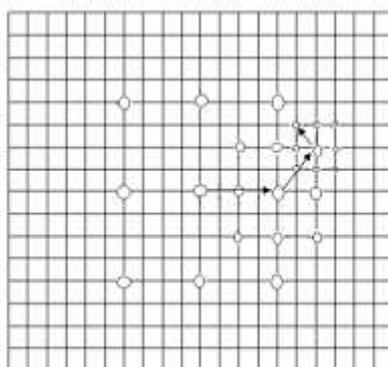


Fig.3. Three Step Search Algorithm

### 3.3 New Three Step Search (NTSS) Algorithm:-

New Three Step Search (NTSS) is the another fast BMA which is used for motion estimation. This Algorithm provide some improvement for better estimation of the motion with low amplitude. NTSS improves the TSS results by providing a center biased searching scheme and stop in the middle of the process to reduce computational cost. It was one of the first widely accepted fast algorithms and frequently used for implementing earlier standards. The NTSS process is shown in fig 4.

In the first step 16 points are checked in addition to the search origin for lowest weight using a cost function. If the lowest cost is found at the origin then the search is stopped and the motion vector is set as (0, 0). If the lowest weight is at any one of the 8 locations, then change the origin of the search to that point and check for lowest weights adjacent to it.

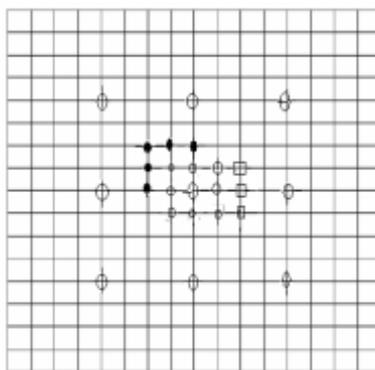


Fig.4. New Three Step Search Algorithm

**3.4 Simple and Efficient Search (SES) Algorithm:-**

Simple and Effective Search is another BMA that provide improvement over TSS method. This algorithm has steps similar to TSS but the difference is that each step in this method has two phases. The search area is divided into four quadrants and the algorithm searches for the location A, B and C (fig 5) depending on their weight. The rules for determining a search quadrant for seconds phase are as follows: If  $MAD(A) \geq MAD(B)$  and  $MAD(A) \geq MAD(C)$ , select (b); If  $MAD(A) \geq MAD(B)$  and  $MAD(A) \leq MAD(C)$ , select (c); If  $MAD(A) < MAD(B)$  and  $MAD(A) < MAD(C)$ , select (d); If  $MAD(A) < MAD(B)$  and  $MAD(A) \geq MAD(C)$ , select (e);

After selection of points in second phase, search for the location which has lowest weight and set it as origin then change the step size as in TSS and repeat the above procedure. The location which has lowest weight is then searched and recorded as of motion vectors. The procedure of SES method is shown in fig 6.

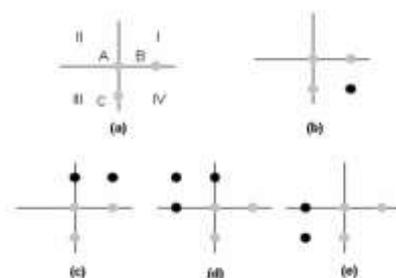


Fig.5. search pattern for SES method

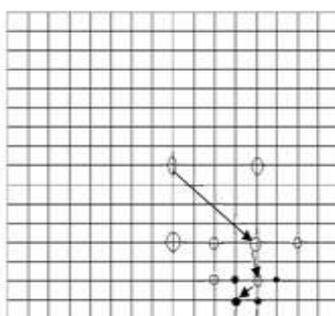


Fig.6. the procedure of SES Algorithm

**3.5 Four Step Search (4SS) Algorithm:-**

Four Step Search also applied a centre biased search pattern for the sequence having maximum motion activity. It is similar to NTSS and stop in the middle of the process like NTSS. 4SS reduces the search points compared to TSS but have worst case performance comparing to NTSS. First, set the step size equals to 2. A search window of  $5 \times 5$  is created which contain nine points around the centre of search area. Then search for the location that has least weight. If centre point of search area has least weight then move to fourth step otherwise find that location out of eight other locations that has least weight, consider it as origin and jump to the next

step. In second step, the search window is still 5x5 pixels wide. Again search for location that has least weight and if it is at centre then jump to fourth step otherwise to third step. Repeat this procedure for third step also. In fourth step, step size becomes equal to 1 and search window becomes 3x3. Then find the location that has least weight which will be the best matched block and record the motion vector. The 4SS method is shown in fig 7. This search algorithm has worst case of 27 checking points.

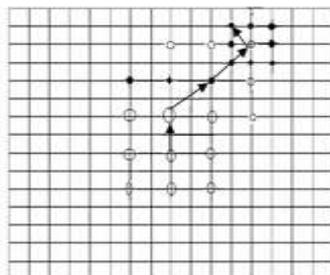


Fig.7. the Four Step Search procedure.

### 3.6 Diamond Search (DS) Algorithm:-

Diamond Search algorithm, another BMA, is the same as 4SS, but the search pattern is changed from square to diamond. DS uses two different types of fixed patterns, they are Large Diamond Search Pattern (LDSP) and Small Diamond Search Pattern (SDSP). These two patterns are shown in fig. 8. Here the search pattern is neither too small nor too big and since there is no limit to the number of steps, this algorithm can find global minimum accurately.

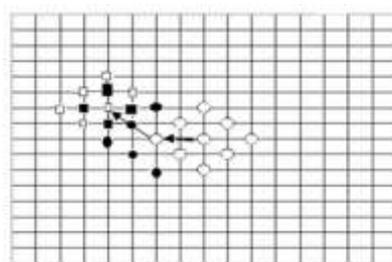


Fig.8. Diamond Search procedure.

### 3.7 Adaptive Root Pattern Search (ARPS) Algorithm:-

ARPS 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 9.

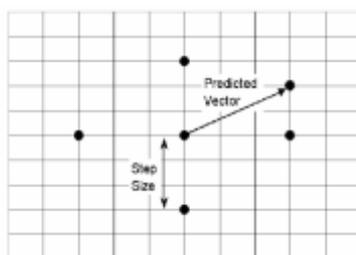


Fig.9. Adaptive Root Pattern Search Method

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 root pattern distributed points, as shown in fig 9, 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 root 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. The point that has the least weight becomes the origin for subsequent search steps, and the search pattern is changed to SDSP. The procedure keeps on doing SDSP until least weighted point is found to be at the center of the SDSP.

#### IV. Experimental Results

In this project, all block matching algorithms described above have implemented in matlab using caltrain video sequence. The results we got is similar to [1], [5] and [6]. Fig.10 shows search point in each macro block for calculation of Peak Signal to Noise Ratio (PSNR) for all algorithms. Fig.11 shows peak signal to noise ratio performance for algorithms. Here caltrain sequence is used with a frame distance of 2. Peak-Signal-to-Noise-Ratio (PSNR) characterizes the motion compensated image that is created by using motion vectors and macro clocks from the reference frame.

In fig 10 and 11, ARPS takes less number of search points and its PSNR is also good in comparison to other algorithms. Although PSNR performance of 4SS, DS and ARPS is nearly same but 4SS and DS more search points in comparison to ARPS. SES also takes less search points but not less than ARPS, its PSNR performance is very low. Hence ARPS takes less steps of computation compared to other algorithms and hence it is the best block matching algorithm.

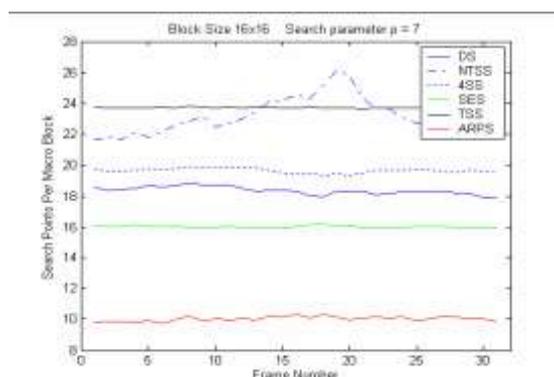


Fig.10. Search points per macro block while computing the PSNR performance of Fast Block Matching Algorithms.

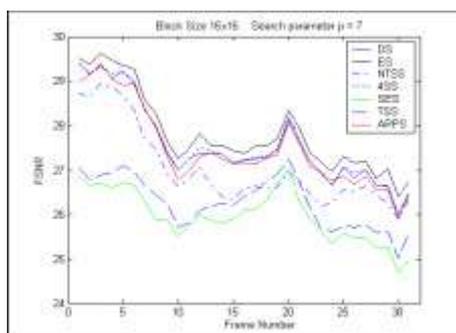


Fig.11. Peak-Signal-to-Noise-Ratio (PSNR) performance of Fast Block Matching Algorithms. Caltrain Sequence was used with a frame distance of 2.

An example of the Matlab implementation results of the motion vector description technique is given by fig 12.

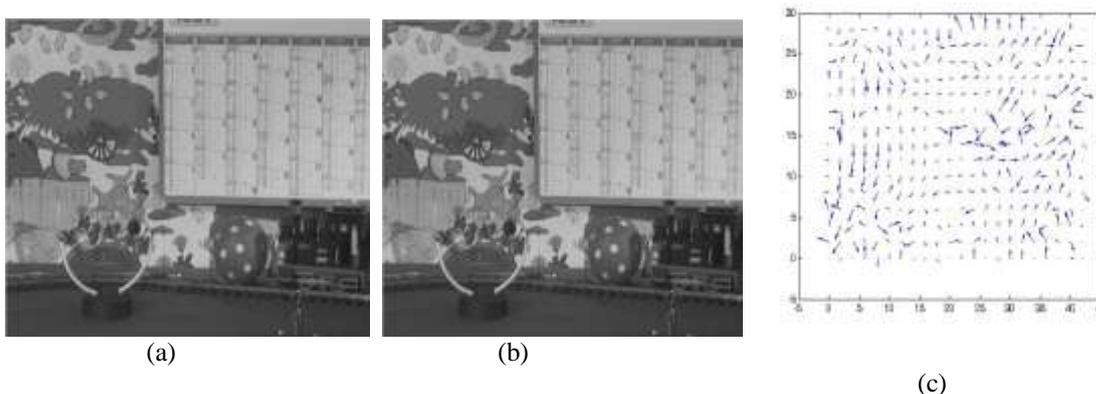


Fig.12. (a) and (b) two successive caltrain frames and (c) motion vector result of (a) and (b) frames.

## V. Conclusion

This paper gives an overview of all Block Matching Algorithms that have been developed. Section II gives introduction of Block matching Algorithm process in detail and section III introduces all block matching algorithms that is used for motion estimation in video. From the experimental results we got and comparison of PSNR performance of all algorithms, we concluded that ARPS gives better results than other algorithms.

## References

- [1] Video shot boundary detection using motion activity descriptor Abdelati Malek Amel, Ben Abdelali Abdessalem and Mtibaa Abdellatif, april 2010.
- [2] Aroh Barjatya, "Block Matching Algorithms For Motion Estimation, "Student Member, IEEE, DIP 6620 Spring 2004 Final Project paper.
- [3] 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
- [4] Lai-Man Po, and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 6, no. 3, pp. 313-317, June 1996.
- [5] 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.
- [6] 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.
- [7] Renxiang Li, Bing Zeng, and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion
- [8] Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, No. 4, pp. 438-442, August 1994.
- [9] Chun-Ho Cheung, and Lai-Man Po, "A Novel Small Cross-Diamond Search Algorithm for Fast Video Coding and Video Conferencing Applications", Proc. IEEE ICIP, September 2002.
- [10] Chun-Ho Cheung, and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", IEEE Trans. CircuitsAnd Systems For Video Technology, vol 12., no. 12, pp. 1168-1177, December 2002.