THREE STEP SEARCH METHOD FOR BLOCK MATCHING ALGORITHM

[1]DISHA D. BHAVSAR, [2]RAHUL N. GONAWALA

[1,2]B.E. Student, Assistant Professor

Abstract- This paper includes algorithmic simulation of three-step search (TSS) block matching algorithm for motion estimation. This method is based on the real world video frame sequence's feature of centre-biased motion vector allocation and uses centre-biased checking point patterns and a small number of search locations to perform fast block matching. Many fast block matching algorithms have been developed to improve the heavy computations of block matching algorithm. These are based on various techniques like fixed search pattern, variable search range, hierarchal and multi resolution algorithms, sub-sampling techniques, partial distortion elimination, spatio-temporal correlation etc. Among early block ME algorithms found in literature, fixed search pattern algorithms are the most famous, these algorithms reduce the computational requirement significantly by checking only some points inside the search window, while keeping a good error performance when compared with Full Search algorithm.

I. INTRODUCTION

The high bit rates that result from the various types of digital video make their transmission through their intended channels very difficult. Even entertainment video with modest frame rates and dimensions would require bandwidth and storage space far in excess of that available from conventional storage memory. Thus delivering consumer quality video on compact disc would be impossible. Hence, there are requirement of efficient method of video compression algorithms. For this purpose, motion estimation plays vital role. In addition to spatial redundancy, temporal redundancy plays an important role in video compression.

In the motion estimation process the frame is divided into number of non overlapping areas known as macro blocks. The difference between the current frame and the predicted contents, motion vector, is calculated in motion estimation.

However motion estimation is the most computational part, it consumes up to 50% of video encoding time [1]. Block matching algorithm, widely used for motion estimation, is mainly classified in full search and fast search. By exhaustively testing all the candidate blocks within the search window, full search algorithm gives the global optimum solution to the motion estimation at the cost of high computational and hardware complexity. To overcome this drawback, many fast block matching algorithms (BMA) have been developed. Various fast BMA such as Three Step Search (TSS), Logarithmic search (LS), Diamond Search (DS), Hexagon Search (HS), Four Step Search (4SS) are reviewed. These fast BMA exploit different search patterns and search strategies for finding the optimum motion vector with drastically reduced number of search points as compared with FS algorithm.

Based on analysis of variants of BMA, three step search method is simulated and presented in this paper. The paper is organized as follows: Section 2 includes preliminaries and introduction related to fast BMA. Section 3 presents TSS algorithm. Section 4 includes simulation set up and result analysis. Section 5 concludes the paper.

II. PRELIMINARIS ABOUT BMA

In block matching, motion estimation is performed between current frame and reference frame of a video sequence. A frame is divided into a grid of blocks, non-overlapped square macro blocks, of pixels with size N x N. They are then compared with corresponding blocks and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block (MB) from one location to another in the previous frame called motion vector. If there is no motion, there is a high correlation between two blocks. This motion vector 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. If there is no motion, there is a high correlation between the two blocks. If something has shifted, the same place position in the next field will not provide good correlation, and it will be necessary to search for the best correlation in the next image. The position that gives the best correlation is assumed to be the new location of a moving object [6]. A Block Matching Algorithm (BMA) is a way of locating matching blocks in a sequence of digital video frames for the purposes of motion estimation. 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 (1). Another cost function is Mean Squared Error (MSE) given by Equation (2).

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

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

$$PSNR = 10\log_{10}\left(\frac{\left(2^b-1\right)^2}{MSE}\right) \tag{3}$$

Where, C_{ij}= Current macroblock
R_{ij}=Reference macroblock
Here, 'b' in the Equation (3) is the number of bits in a pixel.

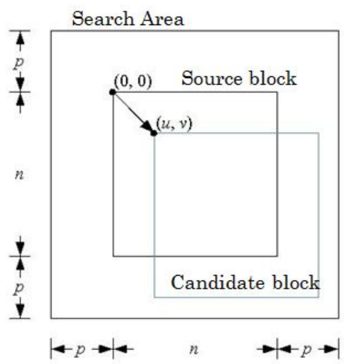


Fig.1 Motion Estimation

Where, Search Area= (n+2p) x (n+2p)
Motion vector= (u, v)

III. THREE STEP SEARCH METHOD

Three step search method, proposed in 1981 [4], is a fine-coarse search mechanism. It 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 TSS is the algorithm that limits the number of checking points in a search area. In Fig. 2, an example is shown to illustrate the procedure of the TSS for W = 7.

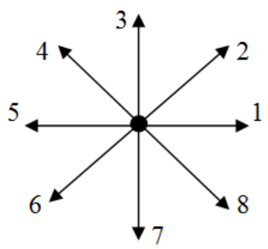


Fig.2 Search direction in a search pattern of the TSS

It must be noted that, though the TSS is originally proposed for low-bit rate video applications where the search window is relatively small, which is typically limited at W = 7, its procedure can be extended for cases with W > 7 [6] while the number of steps becomes more than three. In general, given W, the number of steps required is

$$L = [\log_2(W+1)] \tag{4}$$

Consequently, the step-size (distance between pixels in a search step) for n^{th} step is given by

$$ss(n) = 2^{L-n} \tag{5}$$

It is shown that the TSS uses a uniformly distributed search pattern in each step and hence exhibits simplicity and regularity. In particular, the number of checking points is nine for the first step and eight for the subsequent steps (excluding the location which is already checked in the previous step). For W = 7, the total number of checking points for TSS is 25. Flowchart of TSS is shown in Fig. 3.

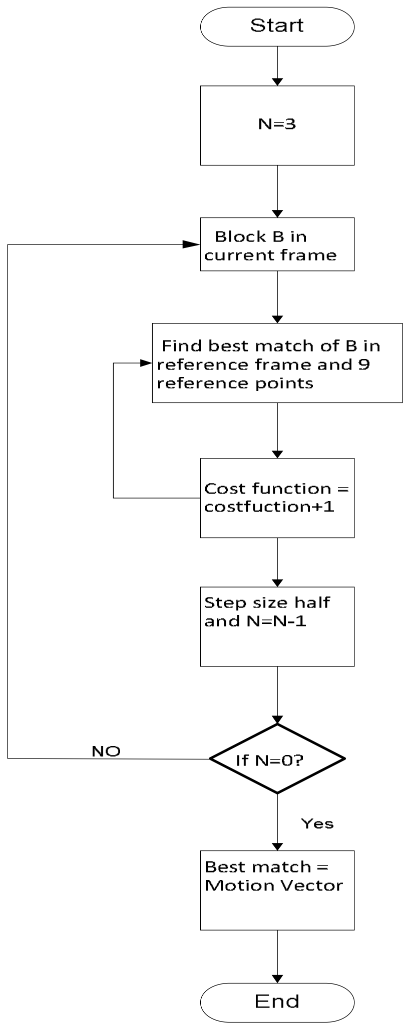


Fig.3 Flowchart for Three step Search method

The algorithm may be described as:
Step 1: An initial step size is picked. Eight blocks at a distance of step size from the centre (around the centre block) are picked for comparison.
Step 2: The step size is halved. The centre is moved to the point with the minimum distortion.
Steps 1 and 2 are repeated till the step size becomes smaller than 1. A particular path for the convergence of this algorithm is shown below:

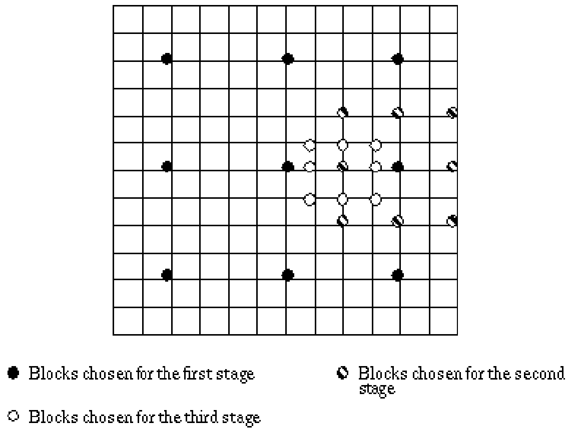


Fig.4 Example path for convergence of Three Step Search

One problem that occurs with the Three Step Search is that it uses a uniformly allocated checking point pattern in the first step, which becomes inefficient for small motion estimation.

IV. SIMULATION SETUP AND RESULT ANALYSIS

In this research work, four standard Quarter Common Intermediate File Format (QCIF) video sequences of different motion contents are used for performance comparison of different algorithms. Simulations are done on Intel(R) Core(TM)i5-3337U CPU@1.80GHz configured system. Various macro-block size and search window combinations are used to test motion estimation. Computational cost is calculated for each case and compared for various cases. Results are shown in Table 1. PSNR is calculated for first 30 frames of video sequence. Table 1 indicates data for average PSNR and average computation/Macroblock. Here search window p=7 is used.

Akiyo
Fig. 5 First frame of video sequences 'Akiyo'

From, Table 1 it is observed that increasing macro block size reduce computational load per macroblock. Table 2 indicates data for average PSNR and average computation/Macroblock. Here search window is altered by keeping Macroblock size to 16.

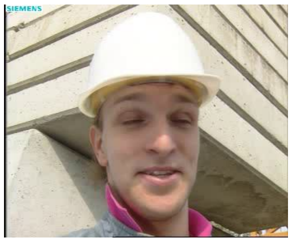
TABLE 1
Average PSNR and Computation for first 30 frames

Macro Block Size	PSNR (dB)	Computation/ Macroblock
4	44.86	8.8491
8	44.64	8.6995
16	44.64	8.4040

TABLE 2
Average PSNR and Computation for first 30 frames

Search window 'P'	PSNR (dB)	Computation/ Macroblock
5	44.66	8.4040
7	44.64	8.4040
9	44.64	8.4040

Now, various standard video sequences are tested for motion estimation. For the simulation, Search window p=7 and Macro-block size = 16 is used to find motion vector and PSNR is calculated and tabulated in Table 3.



Foreman

Crew

News
Fig. 5 First frame of video sequences Foreman, Crew, News

For mathematical computational, computation per macro-block is calculated and compared.

TABLE 3
Average PSNR and Computation for first 30 frames

Frame	PSNR(dB)	Computation/ Macroblock
Foreman	25.110	8.4040
Akiyo	44.64	8.4040
News	29.02	8.4040
Crew	24.06	8.4040

CONCLUSION

In this paper TSS algorithm for motion estimation is described. From the literature survey, it is observed that fast search method gives optimum motion vector and hence, good compression ratio with reduced computational complexity at cost of reduced reproduction quality. Out of variants of fast search algorithm, literature survey suggests TSS as a better option for motion estimation. Experimental results from the simulations on MATLAB for various combinations of standard test video sequences, search window and macro-block demonstrate that TSS provide better performance in terms of PSNR and computation cost.

REFERENCES

[1] Prof. Atalla I.Hashad, Dr. Rowayda A. Sadek, “A Novel Reduced Diamond Search (RDS) Algorithm with Early Termination for Fast Motion Estimation”, Member in IEEE, Sara K. Mandour.

[2] Deepak J. Jayaswal 1, Mukesh A. Zaveri 2, “Fast Predictive Search Motion Estimation Algorithm”, International Journal of Signal and Image Processing. Jan- 2010.

[3] Jianhua Lu and Ming L. Liou, “A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation”.

[4] M.Ezhilarasan and P.Thambidurai, “Simplified Block Matching Algorithm For Fast Motion Estimation in Video Compression”, Journal of Computer Science 4(4):282-289, 2008.

[5] Rahul Gonawala and Prof. Nehal Shah, “Full Search and Fast Search Algorithms for Motion Estimation in Video Compression”, International conference on Information, Knowledge & Research in Engineering, Technology & Sciences,March 2012,Rajkot.

[6] T.Ramaprabha MSc MPhil, Dr.M.Mohamed Sathik, “Three step Vs Four step Block matching search algorithm in Stereo image Compression”.

[7] Vanshree Verma, Sr. Asst. Prof. Ravi Mishra, “Various Fast Block Matching Algorithm for Video Shot Boundary Detection.” Department of Electrical and Electronics Engineering, Shri Shankaracharya College of Engineering & Technology. Junwani, Bhilai (C.G.)

★★★