Three step Diamond Search Algorithm for Fast Block-Matching Motion Estimation

B. Kasi viswanatha Reddy & Sukadev Meher

Electronics & Communication dept, Professor & HOD of EC dept. NIT Rourkela, Orissa,
E-maol : viswanathrdy9@gmail.com,smeher@nitrkl.ac.in

Abstract - Based on the study of motion vector distribution from several commonly used test image sequences,a three step diamond search [TSDS]algorithm for fast block matching motion estimation is proposed in this paper .The performance of this algorithm is compared with other existing algorithms by means of error metrics and no of search points in this the simulation results shows that the proposed three step diamond search algorithm achieves close performance with that of diamond search [DS] and uses less no of search points than the three step search[TSS]. When compared with original diamond search [DS] algorithm, this algorithm requires less computation time and gives an improved performance.

Index Terms*—Block motion estimation, block matching algorithm, search pattern, three step search, diamond search.*

I. INTRODUCTION

Video coding is an important process in many multimedia applications. In addition to spatial redundancy, temporal redundancy plays an important role in the transmission of video frames .An effective and popular method to reduce the temporal redundancycalled block matching (BMME), motion estimation is the technique used to reduce the temporal redundancy. It uses the correlation between the successive frames to predict the content of frames it has been widely adopted in various video coding standards such as H.261, H.263, MPEG-1, MPEG-2, and MPEG-4 [1] and in any motion compensated video coding technique. Therefore fast and accurate block-based search technique is highly desirable to assure much reduced processing delay while maintaining good reconstructed image quality. In the motion estimation process the frame is divided into number of non overlapping areas known as macro blocks. Each macro block can be with a standard size of 16×16. The difference between the current frame and the predicted contents is calculated inmotion estimation [4]. In addition to motion estimation, some additional information's are also needed to indicate any changes in the prediction process this is knownas motion compensation [6].

By exhaustively testing all the candidate blocks with in the search window, full search algorithm gives the global optimum solution (i.e., the minimum matching error point over the search window) to the motion estimation, while a substantial amount of computational load is demanded. To overcome this drawback, many fast block matching algorithms (BMA's) have been developed[5]. For example (TSS) [2], New three step search (NTSS) [2],four step search(4SS) [7], diamond search (DS) ([3],[4]) etc. These fast BMA's 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. In TSS ,NTSS, and 4SS algorithms square shaped search patterns of different sizes are employed in TSS a large search pattern with size of 9 × 9 and sparse checking points as exploited in the first step of TSS is most likely to mislead the search path to wrong direction and hence misses the optimum point. In other ways the diamond search algorithm maintains the diamond shape search pattern, it gives faster search pattern and minimum absolute difference when compared with TSS, NTSS, DS and 4SS. This inspires us to investigate why DS pattern can yield speed improvement over some square shaped search patterns and what the mechanism behind is. as a result we try to reduce the step size it gives the better performance than the DS and TSS, based on these search patterns we propose a three step diamond search algorithm that can achieve a substantial speed improvement. The search speed and the performance of an algorithm are determined by the shape and size of the search patterns. The TSS and NTSS[5] algorithms are using a squared shape pattern, whereas the diamond search algorithm uses a Diamond shape. This diamond search (DS)([3],[4]) algorithm uses unrestricted center biased searching concept and so it is computationally inefficient. In this paper, a three step diamond search algorithm is proposed to attain a computationally efficient search with a reasonable distortion performance gives the minimum number of search points and minimum error when compared to the TSS and DS patterns.

The section II deals with the existing diamond search DS algorithm and its different search patterns and the draw backs of the algorithm. Section III explains about the modified DS algorithm. Simulation results are presented in our proposed algorithm compared with the FS, TSS and DS. the performance will be presented in section IV and section V concludes the paper.

II. DIAMOND SEARCH ALGORITHM

The shape of the search pattern has an impact on the performance of the algorithm. Fast block matching algorithm such as TSS, NTSS are having a square shape search pattern and provides reasonable performance. The distribution of global minimum points is centered at the Centre of search window. A Centre biased NTSS is used to achieve better performance than TSS but it losses the regularity and simplicity. The diamond search provides a better performance than the TSS, NTSS algorithms.

The diamond search (DS) algorithm uses a diamond shape pattern with nine search points, four points located at the corners and another four points located at the midpoint of the edges of the diamond shape. This algorithm uses an unrestricted center biased searching process. The diamond search employs a large diamond search pattern (LDSP) [fig 1(a)] and small diamond search pattern (SDSP) [fig 1(b)].

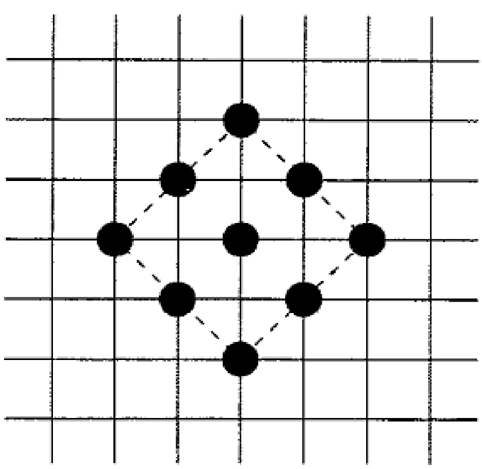

Figure 1(a) Large Diamond Search Pattern

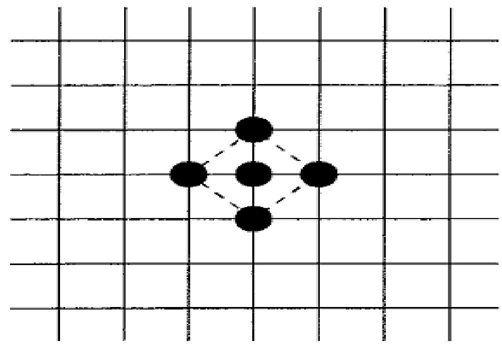

Figure 1(b) Small Diamond Search Pattern.

As some of the search points in the newly formed LDSP are overlapping, only the non over lapping points need to be evaluated. This greatly reduces the number of search points compared to other existing fast search algorithms. Therefore the search pattern uses five search points in the new LDSP if the MBD point is the corner point [fig 2(a)] and the three search points if the MBD point is at the edge of the pattern [fig 2(b)] . The LDSP is used until the centerpoint becomes the MBD point. Once the minimum block distortion (MBD) point is at the center, the search is switched to SDSP which uses four checking points [fig 2(c)].

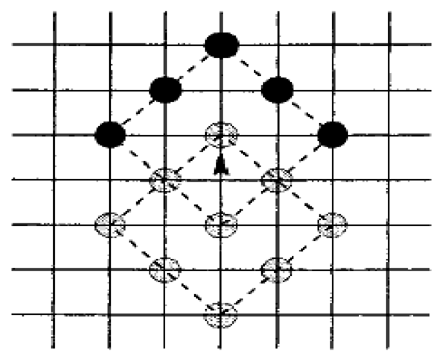

Figure 2(a) The corner point.

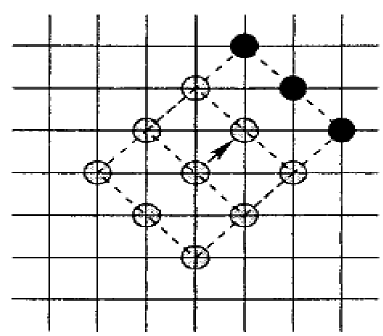

Figure 2(b) The edge point.

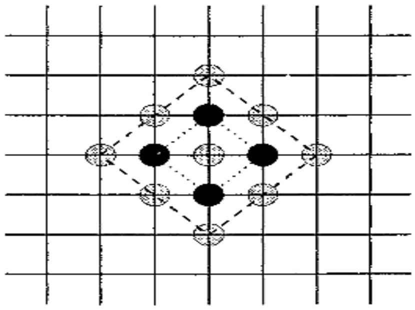

Figure 2(c) The centre point

Fig. The above figures represent the different search patterns over lapping in LDSP in this the fig 1(a) represent the large diamond search pattern when the minimum block distortion point occurred in center the large diamond change to small diamond search pattern the fig 2(a represent the corner search points the fig 2(b)represent the edge search point and center point. The solid block dots are the new checking points where the computation of block-distortion measurement is required for the current search step.

Theminimumblockdistortion point (MBD) thus obtained will give the motion vector. The DS algorithm reduces the susceptibility of getting stuck at local minimadue to its compact shape and relatively large step size in the horizontal and vertical direction. Thus the diamond search algorithm gives a faster processing and similar distortion performance with the other fast searching algorithms. The increase in number of steps leads to more number of search points which has an effect on the speed of the algorithms. This algorithm gives the less complexity when compared to the previous algorithms. A three step diamond search (TSDS) is proposed to overcome this disadvantage.

III. THREE STEP DS ALGORITHM

The proposed TSDS algorithm uses the same type of patterns used in DS algorithm with a reduction in a step size. Based on the location of the MBD point, the number of checking points to be used in the successive steps varies.

The number of searching steps is reduced to three and the SDSP search is reached at the third step regardless of the location of the MBD point. The LDSP is repeatedly used until the centre point becomes the MBD point. The algorithm for this TSDS algorithm is summarized as follows.The number of searching steps is reduced to three and the SDSP search is reached at the third step regardless of the location of the MBD point. The LDSP is repeatedly used until

Thus the compact configuration and reduced number of search points provide an improved performance than the other existing algorithms. The algorithm for this TSDS algorithm is summarized as follows.the centre point becomes the MBD point. Thus the compact configuration and reduced number of search points provide an improved performance than the other existing algorithms given below. The algorithm for this TSDS algorithm is summarized as follows.

(a)Algorithm:

Step 1: Initial LDSP is centered at the origin of the search window. Now, test each points in the search pattern. If the MBD point is the center point go to step3. Otherwise go to step2.

Step 2: Form a new LDSP with the MBD point as the center point. If the new MBD point is at the center position, go to step3. Otherwise repeat this step for one more time.

Step 3: Form the SDSP with previous MBD point as the center point. The new MBD point obtained in this step becomes the final solution i.e., the motion vector (x ,y). The number of search points depends on the location of MBD point also determines the search direction.

TABLE I

Image sequences used for simulation result

Image sequences	Frame size	Length
CALTRAIN	400×512	32
FOREMAN	176×144	55
TENNIS	240×352	90
CRAFTER	256×256	40

The image sequence used for simulation in this the number of search points and MAD for each macro block wise finding and depending on sequence length so in the proposed method we find the MAD for "**caltrain**" sequence this is shown in the figure and flow chart also shows that the search steps and error calculation.

(b) Flow chart

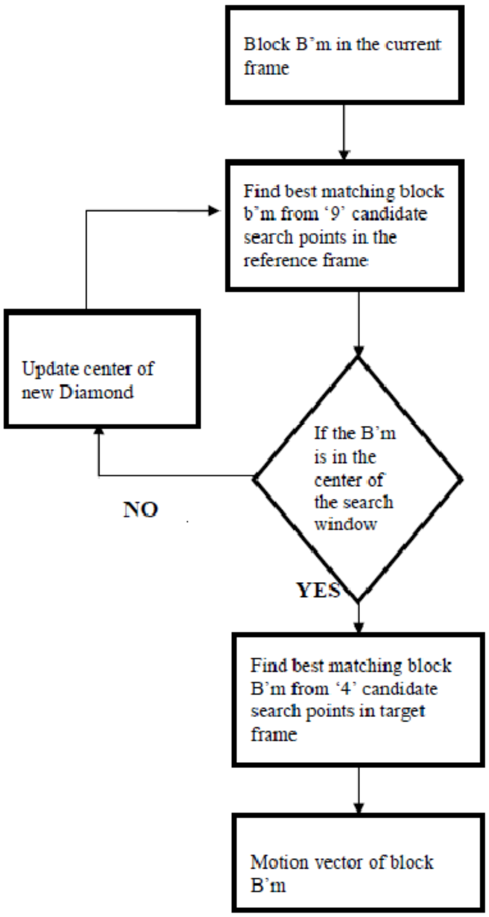

Table-II

Calculation of search points and search steps theoretically

Search Alg.	Searchpoints [MIN]-[MAX]	No ofsearch steps [MIN]----[MAX]		Formulae
TSS	17 ----33	1	3	9+8n
DS	13 ---- 22	0	3	9+3n+4
FS	225-- 225	1	1	(2q+1)^2
TSDS	10 ---- 16	1	3	4+3N+3

Theoretical calculations to find the number of search points and number of search steps. FS takes enter frame as search window size 15×15 so that it will take the large computational complexity (n represents the number of search steps)

TABLE III

Average number of search points per macro block

Algorithm	Caltrain	Foreman	Tennis	Crafter
TSDS	**14.86**	**16.48**	**16.33**	**12.31**
DS	**16.23**	**17.23**	**17.52**	**13.12**
TSS	**21.06**	**24.23**	**25.01**	**24.02**
FS	**225**	**225**	**225**	**225**

TABLE IV

Average MAD per pixel

Algorithm	Caltrain	Foreman	Tennis	Crafter
TSDS	**53.6678**	**160.2307**	**149.5830**	**50.2650**
DS	**54.1875**	**164.7500**	**156.5000**	**53.6193**
TSS	**64.1091**	**217.2374**	**208.1700**	**69.3177**
FS	**50.2344**	**141.2005**	**132.9361**	**48.7834**

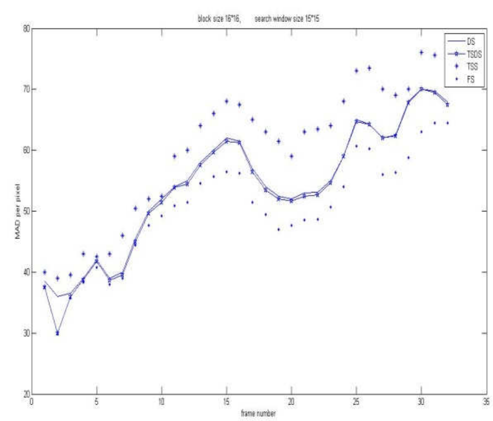

FIG: MAD Comparison of TSDS, DS, TSS and FS "caltrain" sequence.

IV) Simulation result:

In our simulation results, the block size is fixed at 16×16. To make a consistent comparison, block matching is conducted within a search window size of 15×15 is used for the experimentation of this algorithm and the center point of initial LDSP is at the origin of the search window. The performance of the algorithm is evaluated by error metrics, such as the number of search points and means absolute difference. The performance analysis has been done for an **''caltrain''** sequence and the average MAD values and search point numbers of "**caltrain**" and other test sequences are presented in Tables III and IV and Table II shows that the theoretical calculations of number of search points and search steps. The result shows that, it gives a better performance compared with the existing [TSS] and [DS] algorithmswith a reduction in step size also. The search is confined within the search window and the reduction in number of steps results in reduction in computational complexity. Easy way and good pattern of this algorithm provides good implementation. The criterion used for the distortionmeasurement is Sum of Absolute Difference (SAD), which gives the MBD point for the motion vector calculation. The pels are arranged in such a way that two in horizontal direction and in vertical direction, and one in each diagonal direction. This makes the algorithm to reach a globalminimum point. The maximum number of search points used is 23 whereas the TSS uses 25 search points. It achieves a close MSE performance with the DS, TSS, NTSS algorithm for the image sequences with small motion as well as large motion content.

V. CONCLUSION:

In this paper we proposed a Three step Diamond Search Algorithm for computationally efficient block motion estimation. This algorithm having compact shape of the search pattern and step size it outperforms the other existing algorithms such as FS,NTSS, TSS and DS in terms of complexity efficiency with better performance. This algorithm can be used in video coding standards such as MPEG-4, H.264 AVC because of its ease of implementation, better performance and reduced computational complexity.

REFERENCES:

[1] K. R. Rao and J. J Hwang, *Techniques and Standards for Image Videoand Audio Coding*. Englewood Cliffs, NJ: Prentice Hall, 1996.

[2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algori- thm forblock motion estimation," *IEEE Trans. CircuitsSyst.Video Technol.*,vol. 4, pp. 438–442, Aug. 1994.

[3] S. Zhu and K. K. Ma,"A new diamond search algorithm For fast Blockmatching motion estimation,"IEEE Trans. Image Processing,vol. 9, pp. 287–290, Feb. 2000.

[4] J. R. Jain and A. K. Jain, "Displacement measurement and its applicat-ion in interframe image coding," *IEEE Trans.Commu*vol. COM-29,pp.1799–1808, Dec. 1981.

[5] J. Lu and M. L. Liou, "A simple and efficient searchalgorithm for blockmatchingmotion estimation," *IEEE Trans. Circuits Syst. VideoTechnol*,vol. 7, pp. 429–433, Apr. 1997.

[6] R. Srinivasan and K. R. Rao, "Predictive coding based on efficint motionestimation," IEEE ***Trans. Commun.,*** vol.COM-33,pp.1011-1014,Sept. 1985.

[7] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fastblock motion estimation," *IEEE Trans. Circuits Syst Video Technol.*,vol. 6, pp. 313–317, June 1996.

◈◈◈