

## Simplified Block Matching Algorithm for Fast Motion Estimation in Video Compression

M. Ezhilarasan and P. Thambidurai

Department of Computer Science and Engineering and Information Technology,  
Pondicherry Engineering College, Pondicherry-605 014, India

---

**Abstract:** Block matching motion estimation was one of the most important modules in the design of any video encoder. It consumed more than 85% of video encoding time due to searching of a candidate block in the search window of the reference frame. To minimize the search time on block matching, a simplified and efficient Block Matching Algorithm for Fast Motion Estimation was proposed. It had two steps such as prediction and refinement. The temporal correlation among successive frames and the direction of the previously processed frame for predicting the motion vector of the candidate block was considered during prediction step. Different combination of search points was considered in the refinement step of the algorithm which subsequently minimize the search time. Experiments were conducted on various SIF and CIF video sequences. The performance of the algorithm was compared with existing fast block matching motion estimation algorithms which were used in recent video coding standards. The experimental results were shown that the algorithm provided a faster search with minimum distortion when compared to the optimal fast block matching motion estimation algorithms.

**Key words:** Block matching algorithm, motion estimation, video compression

---

### INTRODUCTION

Block-matching motion estimation is the most important module for any motion compensated video coding standards such as ISO/IEC MPEG<sup>[1]</sup> and ITU-T<sup>[2]</sup>. The block-matching algorithms eliminate the temporal redundancy, which is found predominantly in any video sequence. It divides frames into equal sized non-overlapping blocks and calculates the displacement of the best-matched block from the previous frame as the motion vector of the block in the current frame within the search window. During block matching, each target block of the current frame is compared with a previous frame in order to find the best matching block. Block-matching algorithms calculate the best match using Mean Absolute Difference (MAD)<sup>[3]</sup>. The Full search algorithm provides the best result by matching all possible blocks within the search window. On the other hand, it lacks significantly in computation time, which necessitates improvement.

To improve the motion estimation search time, there has been a tremendous contribution by researchers, experts from various institutions and research laboratories for the past two decades for refining the block-matching algorithms<sup>[4-8]</sup>. Few fast block matching motion estimation algorithms were

considered in different video coding standards such as Two-Dimensional Logarithmic Search<sup>[9]</sup>, Three Step Search<sup>[10]</sup>, Four Step Search<sup>[11]</sup>, Block-based gradient descent search<sup>[12]</sup>, Diamond Search (DS)<sup>[13]</sup>, Cross-Diamond Search (CDS)<sup>[14]</sup>, Efficient Three Step Search (E3SS)<sup>[15]</sup> and Novel Hexagon-based Search (NHS)<sup>[16]</sup>. Among these, Hexagonal Search Motion Estimation Algorithm has been incorporated in recently developed H.264/AVC video coding standard<sup>[17-19]</sup>. All these block-matching algorithms were minimizing the search time either by having different search patterns or less number of searching points.

The Full Search or exhaustive search algorithm (FS) acts as a benchmark for evaluating the efficiency of all existing fast block-matching motion estimation algorithms. To minimize the search time of the block matching, a simplified and efficient Direction-based Block Matching (DBM) algorithm for fast block motion estimation. To evaluate the algorithm, Full Search, Diamond Search, Cross-Diamond Search, Novel Hexagon-based Search and Efficient Three Step Search algorithms were considered.

The study was organized as follows. In the second section, various existing fast block-matching motion estimation algorithms were discussed. The detailed discussion of simplified and efficient direction-based

block matching algorithm was given in third section. Experimental results conducted on various SIF and CIF video sequences were provided for validation in fourth section followed by Conclusion and References.

## MATERIALS AND METHODS

In a conventional predictive coding<sup>[20-23]</sup>, the difference between the current frame and the predicted frame is encoded. The prediction is done using any of the BMA. BMA are used to estimate the motion vectors. Block-matching consumes a significant portion of time in the encoding step.

The search performed in a restricted region called the search area, which is usually rectangular in dimension. An assumption is made on the maximum distance; objects in the video sequence tend to move between adjacent frames. This distance is called the maximum displacement. The larger the value of maximum displacement assumed, the greater the accuracy of reconstruction. In the exhaustive search procedure, all the blocks in the search area are considered for block matching. The motion vector describes the location of the matching block from the previous frame with reference to the position of the target block in the current frame. Distortion between the current block and reference block of previous frames are normally measured by Mean Squared Error (MSE)<sup>[24]</sup> or Mean Absolute Difference (MAD). Out of these, MAD is efficient best measure due to its computation, which does not require multiplication. The MAD for a block A of size MxN located at (x, y) inside the current frame, compared to a block B located at a displacement relative to A in the previous or reference frame is given as follows:

$$MAD = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (1)$$

Where, M x N is the size of the macro block,  $C_{ij}$  and  $R_{ij}$  denote the pixel intensity in the current frame and previously processed frames respectively. After checking each location in the search area, the motion vector is then determined as the (x, y) at which the MAD has the minimum value. The smaller the magnitude of MAD the greater is the accuracy of prediction. Once the motion vectors are determined, they must be assigned with bit sequences. The difference between a predicted frame and the original frame are encoded along with the motion vectors.

The above procedure is usually consumes more time. Hence, the requirement arises for the development of fast BMA to reduce the search time<sup>[25]</sup>. The fast

block-matching motion estimation algorithms considered for evaluation are briefly given below.

**Cross-diamond search:** In this algorithm, a cross-shaped search pattern is used as the initial step and large/small diamond search patterns as the subsequent steps for fast block motion estimation. The initial cross-search pattern is designed to fit the cross-center-biased motion vector distribution characteristics of the video sequences by evaluating the nine relatively higher probable candidates located horizontally and vertically at the center of the search grid. The CDS uses a small cross-shaped search patterns in the first two steps to speedup the motion estimation of stationary and quasi-stationary blocks.

**Efficient three step search algorithm:** This algorithm is a refinement of existing Three Step Search algorithm and is found to provide a better computational complexity and a comparable distortion to its counterpart. This algorithm starts with a small diamond search pattern at the search window center. If the minimum block distortion measure point is at one of the points on the 9x9 grid proceed as in Three Step Search but if the minimum is one of the four points on the small diamond, the small diamond center is set to the minimum point and another three points will be checked.

**Novel Hexagon-based Search Algorithm:** In this algorithm, a circle-shaped search pattern with a uniform distribution of a minimum number of search points is desirable to achieve the fastest search speed. Each search point can be equally utilized with maximum efficiency. In the diamond search pattern, it is observed that the diamond shape is not approximate enough to a circle, which is just 90 degree rotation of a square. Consequently, a more circle-approximated search pattern is expected in which a minimum number of search points are distributed uniformly.

The searching points in different search pattern and limited searching steps are the prime criteria have been followed in existing motion estimation algorithms. In addition to these, the characteristic of the object motion such as direction has been considered in the proposed algorithm to minimize the search time.

**Simplified and efficient direction-based block matching algorithm:** An object in a video sequence continues to move in the same direction or may be passive for a period of time. Turbulence is a rare phenomenon. Hence, for comparison of the matching block and the target block, it is not required to

search the entire set of candidate blocks in the search area. Instead, the candidate block to be searched can be predicted with a high probability of accuracy using the motion vectors of the previous frame as shown in Fig. 1a. If the prediction coincides with the matching block, then the motion vectors of the matching block in the previous frame can be considered as the motion vectors of the target block in the current frame, else the motion vectors may require some refinement as shown in Fig. 1b. This refinement of the motion vectors is achieved using any of the existing fast block matching algorithms with a smaller search area thus contributing to the reduction in time.

The algorithm reduces search time by making use of motion vector positions using the relationship between frames. The reduction of search time and the prediction of motion vector are achieved as follows. The algorithm involves two steps namely, prediction step and refinement step.

**Prediction step:** The DBM algorithm utilizes the motion vectors of the previous frame to predict the motion vectors of the current frame. Before executing the prediction step, the predicted motion vectors for all the macro blocks of the current frame will be assigned as (0, 0).

Consider the block (i, j) of the both previous frame  $I_{k-1}$  and current frame  $I_k$ . The values of motion vectors of  $I_{k-1}$  (previous frame) are used to predict the values of motion vectors of  $I_k$  (current frame). If the motion vector of the k-1<sup>th</sup> frame ( $PMV_{k-1}(i, j)$ ) is (m, n) then the Predicted Motion Vector (PMV) of the k<sup>th</sup> frame as illustrated in Figure 1 is given by

$$PMV_k(i + \frac{m}{M}, j + \frac{n}{N}) = (m, n) \quad (2)$$

Here,  $m/M$  signifies the number of blocks in the vertical direction, the object at position (i, j) moved in the next frame. Similarly,  $n/N$  signifies the number of blocks in the horizontal direction.

The Eq. 2 derived from the fact that the block (i, j) in frame  $I_{k-1}$  is obtained by moving the block (i - m/M, j - n/N) from the frame  $I_{k-2}$  by a distance (m, n). The block (i - m/M, j - n/N) in frame  $I_k$  is obtained by moving the block (i, j) in frame  $I_{k-1}$  by a distance (m, n) if the block continues to move in the same direction. This step is repeated for all the blocks in the current frame ( $I_k$ ).

During the prediction step, some of the macro blocks may not be referred. For those macro blocks that are not referred, the prediction vectors will be (0, 0). For such macro blocks, the corresponding matching will be searched during the refinement step to get their

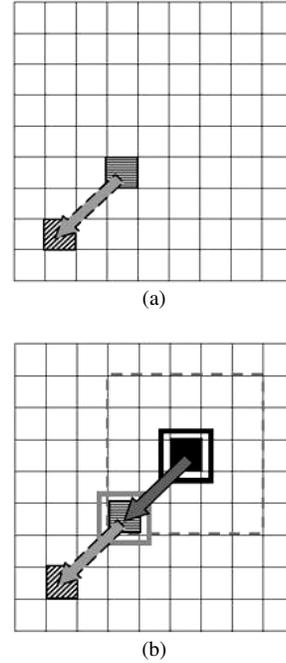


Fig. 1: Simplified and Efficient Direction-based Block Matching procedure: (a) Position of objects in k-1<sup>th</sup> frame and k-2<sup>th</sup> frame with motion vector and (b) Search procedure of the DBM algorithm

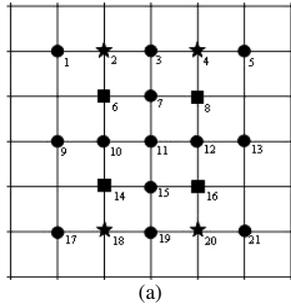
motion vectors. There is a chance for more than one predicted motion vector for the same macro block. In that case, the motion vector which gives minimum Mean Absolute Difference (MAD) is considered.

**Refinement Step:** Let (m, n) be the predicted motion vector for the block (i, j) in the current frame  $I_k$ . The accuracy of the predicted vector can be improvised by refining:

- Compute  $MAD_{(i, j)}(0, 0)$  and  $MAD_{(i, j)}(m, n)$  for the current frame  $I_k$
- If  $MAD_{(i, j)}(0, 0)$  is minimum, then the refinement is centered on the block (i, j) else the refinement is done around the block (i+m/M, j+n/N)

Most probable searching points used in recent existing fast BMAs are categorized as shown in Fig. 2a. Fast block-matching motion estimation algorithms applying any one of the searching patterns to determine the motion vector for the candidate block of the current frame in the reference frame in a search window ( $W = \pm 7$ ) is illustrated in Fig. 2 b.

The algorithm is expressed as DBM1, DBM2, DBM3, DBM4 and DBM5 for having different



Pattern (I)	8 search -points: 1, 3, 5, 11, 13, 21, 19, 17, 9
Pattern (II)	8 search -points: 9, 10, 11, 12, 13, 3, 7, 15, 19
Pattern (III)	8 search -points: 6, 7, 8, 11, 12, 16, 15, 14, 10
Pattern (IV)	4 search -points : 10, 11, 7, 12, 15
Pattern (V)	8 search -points : 9, 6, 3, 8, 11, 1, 3, 16, 19, 14
Pattern (VI)	6 search -points : 9, 2, 4, 13, 20, 18

Fig. 2: Refinement patters of SBMA (a) Searching points in fast BMAs and (b) Different Search patterns

combination of searching patterns. Various searching patterns used in refinement step of these schemes are explained hereunder.

**Searching patterns used in refinement step:** DBM1- All pixels in the small search window centered around the  $MAD_{(i,j)}(m, n)$  region are searched with  $(i+m/M, j+n/N)$  displacements to locate the best match block.

**DBM2:** Apply searching patterns (V) and (IV) in refinement step:

- (i) Searching originates from centre and checks other eight search points of pattern (V). If the minimum MAD (MMAD) point calculated is at centre position, then go to step (iii) otherwise continue
- (ii) The MMAD determined in step (i) is re-assigned as the center for the new search with pattern (V). If the new MMAD is at the centre position, then continue otherwise repeat step (ii)
- (iii) Apply pattern (IV) centered around the new MMAD to check all four search points for best matching block

**DBM3:** Apply searching patterns (II), (V) and (IV) in refinement step

- (i) Searching originates from centre and checks other eight search points of pattern (II). If the minimum MAD (MMAD) point calculated as centre position, then goes to step (iii) otherwise continue

- (ii) The MMAD determined in step (i) is re-assigned as the centre for the new search with pattern (V) and checks other eight search points. If the new MMAD is at the center position, then continue otherwise recursively repeat step (ii)
- (iii) Apply searching pattern (IV) centered around the new MMAD to check all four search points for best matching block

**DBM4:** Apply searching patterns (VI) and (III) in refinement step:

- (i) Searching originates from center and checks other six search points of pattern (VI). If the MMAD point calculated is center position, then go to step (iii) otherwise continue
- (ii) The MMAD determined in step (i) is re-assigned as the center for the new search with pattern (VI) and check other six search points. If the new MMAD is at the center position, then continue otherwise recursively repeat step (ii)
- (iii) Apply pattern (III) centered around the new MMAD to check all eight search points for best matching block

**DBM5:** Apply search patterns (I) and (IV) in refinement step:

- (i) Searching originates from centre and checks other eight search points of pattern (I). In addition to this, four more points of pattern (IV) are also checked for MMAD. If the MMAD point calculated is at center, it is found to be a best matching block and terminates the searching process. Otherwise, continue.
- (ii) By reducing the large 9x9 search window size by half, the new MMAD determined in step (i) is re-assigned as the center for the new search with pattern (I) and check other eight points. If the new MMAD is at the center position, then continue otherwise repeat step (ii).
- (iii) The new MMAD determined in step (ii) is re-assigned as the center for the new search with pattern (VI) and check all other four search points. If the MMAD point calculated is at centre, it is found to be a best matching block and terminates the searching process. Otherwise recursively repeat step (iii).

Refinement is achieved by applying any one the above procedure at the minimum distortion position  $((m, n)$  or  $(0, 0))$  as the centre, but with minimum number of search points to reduce the motion estimation search time consistently.

**RESULTS AND DISCUSSION**

The experiments were conducted on three SIF (Source Input Format) video sequences such as Bike, (352x240, 147 frames, 30 fps, 24 bpp), Flower Garden (352x240, 147 frames, 30 fps, 24bpp), Table Tennis, (352x240, 147 frames, 30 fps, 24 bpp) and a CIF (Common Intermediate Format) Football, (352x288, 50 frames, 25 fps, 24 bpp) video sequence. The simulation has been conducted for 147 frames of the Bike sequence, which is a typical slow varying with bike object motion and most of the background objects are stationary or quasi-stationary. No foreign object intervention is anticipated in the video sequence. Flower Garden, sequence with 147 frames consists of mainly stationary objects, but with a fast camera panning motion. There is a lot of new foreign object intervention in the middle of video sequence. The simulation was also conducted on 147 frames SIF Table Tennis, sequence, which contains different combinations of still, slow, panning and fast moving objects with camera zoom. Football, sequence contains large displacement and fast local object motion, different combinations of still, slow and fast moving objects, camera zoom and panning.

Two important measures considered for analysis are average MAD per pixel and average number of search points (NOP) per block. The search window  $w = \pm 7$  is used for a block size of  $8 \times 8$ . The experimental results are discussed below.

Table 1 shows the performance of the developed schemes with existing optimal and sub-optimal BMA in terms of MAD per pixel and Average search points for Bike, sequence. There is a tradeoff between these two measures.

As per the searching speed is concerned, DBM1 is 4.6 times faster than FS, DBM2 is 9.8 times faster than FS, DBM3 is 5.0 times faster than FS, DBM4 is 13.4 times faster than FS and DBM5 is 7.1 times faster than

FS. It is also found from the above that DBM4 is a fastest scheme for Bike sequence over fast BMAs such as DS by 2.8, CDS by 2.7, NHS by 1.9 and E3SS by 2.4 times.

Table 2 shows the performance comparison for Flower Garden, sequence. It clearly demonstrates that DBM1 is 4.6 times faster than FS, DBM2 is 11.6 times faster than FS, DBM3 is 7.3 times faster than FS, DBM4 is 14.5 times faster than FS and DBM5 is 9.3 times faster than FS. It is also found from the above that DBM4 is a fastest scheme for “Flower Garden” sequence over fast BMAs.

The performance comparison for Table Tennis, sequence is given in Table 3. The empirical results show that DBM1 is 4.6 times faster than FS, DBM2 is 8.7 times faster than FS, DBM3 is 7.0 times faster than FS, DBM4 is 7.7 times faster than FS and DBM5 is 10.7 times faster than FS. From the Table 3, it is observed that the DBM5 scheme outperforms all fast BMAs and other DBM schemes in terms of MAD per pixel and average search points.

Figure 3 shows the frame by frame performance comparison of DBM5 for Flower Garden, sequence with other fast BMAs considering DS, CDS and E3SS in terms of MAD per pixel with a minimum of 12.4 to a maximum of 18.4 and also it predicts the required target block in the reference frame with faster rate over DS, CDS and E3SS. A comparable performance is also obtained with NHS on both MAD per pixel and Average search points per block.

Figure 4a shows the frame by frame performance of DBM5 with other fast BMAs such as DS, CDS, NHS and E3SS in terms of MAD per pixel. Diamond-shaped search points used in DS algorithm gives 0.003-1.1, whereas CDS gives 0.003-1.3 more prediction error values than the DBM5 algorithm. Hexagon-shaped search points used in NHS algorithm gives 0.001-2.4 more prediction error values and E3SS gives 0.001-1.8 more prediction error values than the DBM1 algorithm.

Table 1: Performance comparison for Bike sequence

BMA	FS	DS	CDS	NHS	E3SS	DBM 1	DBM 2	DBM 3	DBM 4	DBM 5
MAD per pixel	4.05	5.02	4.74	4.80	5.04	4.58	5.17	4.76	5.19	5.411
Average search points	225.00	47.72	45.99	32.21	41.04	49.00	22.86	44.43	16.83	31.670

Table 2: Performance comparison for Flower Garden sequence

BMA	FS	DS	CDS	NHS	E3SS	DBM1	DBM2	DBM3	DBM4	DBM5
MAD per pixel	13.05	16.3	15.0	14.0	14.89	13.59	20.56	13.73	20.50	14.01
Average search points	225	36	28.6	22.19	28.13	49	19.47	30.9	15.47	24.23

Table 3: Performance comparison for Table Tennis sequence

BMA	FS	DS	CDS	NHS	E3SS	DBM1	DBM2	DBM3	DBM4	DBM5
MAD per pixel	5.94	6.89	7.144	7.35	7.30	6.61	7.39	6.78	7.38	6.69
Average search points	225	33.88	27.51	24.78	28.93	49	25.86	31.77	29.10	21.07

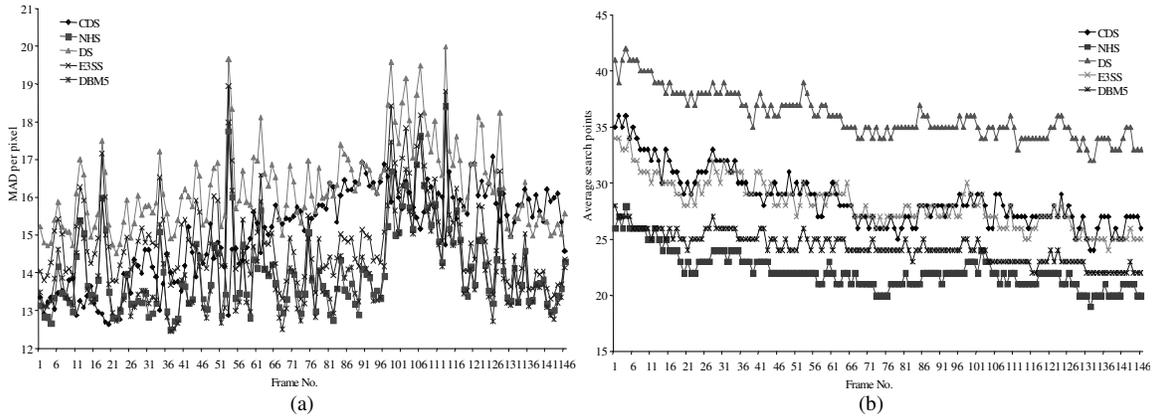


Fig. 3: Frame by frame performance comparison among DBM5 and other fast BMAs for Flower Garden SIF sequence: (a) MAD per pixel and (b) Average search points per block

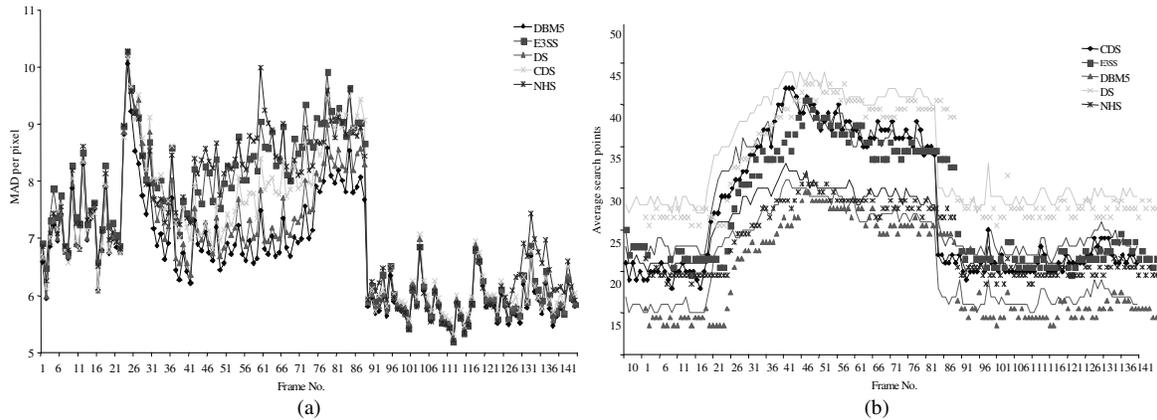


Fig. 4: Frame by frame performance comparison among DBM5 and other fast BMAs for Table Tennis SIF sequence: (a) MAD per pixel and (b) Average search points per block

Table 4: Performance comparison for Foot ball sequence

BMA	FS	DS	CDS	NHS	E3SS	DBM1	DBM2	DBM3	DBM4	DBM5
MAD per pixel	4.02	4.61	5.31	4.66	4.59	4.89	5.99	4.56	4.98	5.65
Average search points	225	33.84	29.8	21.2	34.08	49	17.68	29.7	14.56	27.46

The performance comparison on average search points per block is also shown in Fig 4b. The graph demonstrates that the DBM5 algorithm outperforms other fast BMAs by 1.60797 times faster than DS, 1.3 times faster than CDS, 1.1 times faster than NHS and 1.3 times faster than E3SS algorithm.

Table 4 shows the performance comparison for Football, sequence. It justifies that DBM1 is 4.6 times faster than FS, DBM2 is 12.7 times faster than FS, DBM3 is 7.5 times faster than FS, DBM4 is 15.4 times faster than FS and DBM5 is 8.2 times faster than FS.

It is also found from the above that DBM4 is a fastest scheme for Football, CIF sequence over fast

BMAs such as DS by 2.3, CDS by 2.0, NHS by 1.4 and E3SS by 2.3 times. Every proposed scheme has the advantage of either minimum MAD or less number of search points over some of the fast BMAs. From the Table 4, it is observed that DBM5 scheme outperforms all fast BMAs and other DBM schemes in terms of MAD per pixel and average search points per block. Marginal improvement is also observed between DBM5 and DBM1 in terms of MAD per pixel.

Table 1-4 gives the overall comparison among all optimal and sub-optimal fast BMA and proposed DBM schemes for four different SIF and CIF sequences. Statistical comparisons given in the tables show that the

proposed Simplified and Efficient DBM schemes outperform the existing fast BMAs in terms of MAD per pixel and average search points per block.

### CONCLUSION

A simplified and efficient direction-based block matching algorithm for fast motion estimation was developed. The direction of the previously processed frame for predicting the motion vector of the candidate block was considered during prediction step of the algorithm. Different combination of search points was also incorporated in the refinement step of the algorithm which subsequently minimize the search time. The performance of the algorithm was compared with bench-marking FS and existing fast block matching motion estimation algorithms such as DS, CDS, NHS and E3SS. The developed algorithm outperform s the optimal Full Search algorithm in terms of search points and other fast BMAs in terms of MAD per pixel and average search points per block for different SIF and CIF video sequences.

### ACKNOWLEDGEMENTS

The authors wish to thank Arun Swaminathan, ManiSekhar, Jyotsna Pradhan, Ramya and Sushma Bhatt, the undergraduate students for their valuable help.

### REFERENCES

1. Didier Le Gall, 1991. MPEG: A video compression standard for multimedia application. Commun. ACM, DOI: ACM 0002-0782/91/0400-046, 34: 46-58.
2. Ming Liou, 1991. Overview of the px64 Kbps video coding standard. Commun. ACM, DOI: ACM 0002-0782/91/0400-059, 34: 59-63.
3. C. Cafforio and F. Rocca, 1976. Methods for measuring small displacements of television images. IEEE Trans. Inform. Theory, ISSN: 0018-9448, 22: 573-579.
4. R. Srinivasan and K. R. Rao, 1985. Predictive coding based on efficient motion estimation. IEEE Trans. Commun., ISSN: 0096-2244, 33: 888-896.
5. A. Puri, H. M. Hang and D. Schilling, 1987. An efficient block-matching algorithm for motion compensated coding. Proc. International Conference on Acoustic, Speech and Signal Processing, Texas, USA, April, pp: 1063-1066.
6. J. K. Aggarwal, and N. Nandhakumar, 1988. On the computation of motion from sequences of images: A review. Proc. IEEE, DOI: 10.1109/5.5965, 76: 917-935.
7. Frederic Dufaux and Fabrice Moscheni, 1995. Motion estimation techniques for digital TV: A review and a new contribution. Proc. IEEE, DOI: 0018-9219/95, 83: 858-876.
8. Bing, Z., L. Rexiang and L.L. Ming, 1997. Optimization of fast block motion estimation algorithms. IEEE Trans. Circuits Syst. Video Technol., DOI: 10.1109/76.644063, 7: 833-844.
9. Jaswant R. Jain and Anil K. Jain, 1981. Displacement measurement and its application in inter frame image coding. IEEE Trans. Commun., DOI: 0090-6778/1200-1799, COM-29: 1799-1808.
10. Koga, T., K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, 1981. Motion compensated inter frame coding for video conferencing. Proc. National Telecommunications Conference, New Orleans, LA, December, pp: 63-67.
11. Lai-Man Po and Wing Chung Ma, 1996. A novel four step search algorithm for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol., DOI: 1051-8215/96, 6: 313-317.
12. Lurmg-Kuo Liu and Ephraim Feig, 1996. A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Trans. Circuits Syst. Video Technol., DOI: 1051-8215/96, 6: 419-422.
13. Shan Zhu and Kai-Kuang Ma, 2000. A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans. Image Process., DOI: 1057-7149/00, 9: 287-290.
14. Chun-Ho Cheung and Lai-Man Po, 2002. A novel cross-diamond search algorithm for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol., DOI: 1051-8215/02, 12:1168-1177.
15. Xuan Jing and Lappui Chau, 2004. An efficient three-step search algorithm for Block Motion Estimation. IEEE Trans. Multimedia, DOI:10.1109/TMM.2004.827517, 6: 435-438.
16. Ce Zhu, Xiao Lin, Lappui Chau and Lai-Man Po, 2004. Enhanced hexagonal search for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol., DOI:10.1109/ TCSVT.2004. 833166, 14: 1210-1214.
17. Thomas, W., J.S. Gary and L. Ajay, 2003. Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol., DOI: 10.1109/TCSVT.2003.815165,13: 560-576.

18. Garry J. Sullivan and T. Wiegand, 2005. Video compression-from concepts to the H.264/AVC standard. Proc. IEEE, DOI: 10.1109/JPROC. 2004. 839617, 93: 18-31.
19. Yu-Wen, H., H. Bing-Yu, C. Shao-Yi, M. Shyh-Yih and C. Liang-Gee, 2006. Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC. IEEE Trans. Circuits Syst. Video Technol., DOI: 10.1109/TCSVT.2006.872783, 16: 507-522.
20. Netravali, A.N. and J.D. Robbins, 1979. Motion compensated television coding: Part I. Bell. Syst. Technical J., 3: 631-670.
21. Netravali, A.N. and J.O. Limb, 1980. Picture coding: A review. Proc. IEEE, DOI: 0018-9219/80/0300-0366, 68: 366-406.
22. Musmann, H.G., P. Pirsch and Hans-Joachim Grallert, 1985. Advances in picture coding. Proc. IEEE, DOI: 0018-9219/85/0400-0523, 73: 523-548.
23. Kappagantula, S. and K.R. Rao, 1985. Motion compensated inter frame image prediction. IEEE Trans. Commun, DOI: 0090-6778/85/0900-1011, 33: 1011-1015.
24. Haskell, B.G., P.G. Howard, Y.A. LeCun, A. Puri, J. Ostermann, M.R. Civanlar, L. Rabiner, L. Bottou and P. Haffner, 1998. Image and video coding-emerging standards and beyond. IEEE Trans. Circuits Syst. Video Technol., DOI: 10.1109/76.735379, 8: 814-837.
25. Iain, E.G. Richardson, 2002. Video Codec Design-Developing Imaging and Video Compression Systems. 1st Edn., John Wiley and Sons (England) Ltd, pp: 94-117.