International Journal of Advances in Electronics Engineering
Vol:1 Issue:1 ISSN 2278 - 215X

Directional Hexagon-based Block Search Algorithm

AkashAambulkar, MukeshAngrakh, ShaileshWankhede, SujitKautkar and Vidya N. More
Department of Electronics and Telecommunication Engineering
College of Engineering, Pune
Pune, India - 411005
Email: {ambulkaram07, angrakhms07, wankhedess07, kautkarsn07, vnm} .extc@coep.ac.in

***Abstract*— Many fast block matching algorithms have been proposed to reduce computational complexity in motion estimation process. Previously developed hexagonal algorithms focus on improvement of either low resolution coarse search or fine-resolution inner search. In this paper, proposed algorithm named Directional Hexagon-based Block Search (DHEXBS) improves both coarse and inner search. The proposed algorithm reduces the number of search points by exploiting the distortion information in the neighboring search points. Our experimental results show that the proposed algorithm substantially outperforms HEXBS in terms of number of search points while maintaining the Peak Signal-to-Noise ratio. (PSNR)**

***Keywords*—Motion estimation, Block matching algorithm, Hexagon-based block search.**

I. INTRODUCTION

Many fast block matching algorithms have been developedin these recent years to reduce the number of computations compared to the full search. The two-dimensional logarithmic search algorithm [1], new three-step search (NTSS) [2],efficient three-step search algorithm (ETSS) [3], block-based gradient descent search algorithm (BBGDS) [4], and diamond search (DS) [5], [6] algorithms are amongst the class of fast search methods by reducing the number of search points in the process of block motion estimation. The previously developed hexagonal search algorithm (HEXBS) [7] performs better due to reduction in the in the search points thus achieving faster processing. Search process in the HEXBS is divided in two patterns, first one is the coarse search which performs the outer low resolution search and the second one is the inner search which performs the fine-resolution inner search for finding out the most accurate matching block.

The coarse search starts with the hexagonal search pattern which consists of seven search points having six search points surrounding the center point as shown in the Fig.1. Among the six points, the horizontal two points are at a distance of 2 from the center point and the remaining four points are at a distance of from the center. The distance between the neighboringand the center point is either 2 or .Then the optimum pointis evaluated among the seven search points. The optimum point can be found out by using block matching criterion.The matching criterion is theevaluation function that measures the degree of matchingbetween two blocks. Sum of absolute difference (SAD) is used as matching criteria. SAD can be calculated with the equation given by

$$SAD(V_i) = \sum_{x=0}^{M}\sum_{y=0}^{N} |S_l(x,y) - S_{l-1}(x+dx, y+dy)|$$

where M and N are the block width and height, respectively. is the pixel value of frame *l*at relative position x, y from the macro-block origin and $V_i = (d_x, d_y)$ is the displacement vector.

If the optimum point is the center itself, then the inner search is performed around this point. If optimum point is among the six points, then the hexagonal search pattern will advance with the centermoving to any of the six search points. There are always three new search points emerging and the other three search points being overlapped with the previous hexagon which saves thefurther computation. This process is continued till the central global optimum point is found out. After finding out the central point as global optimum it will perform the fine-resolution inner search around that point. The inner search will find out the most accurate matching block.

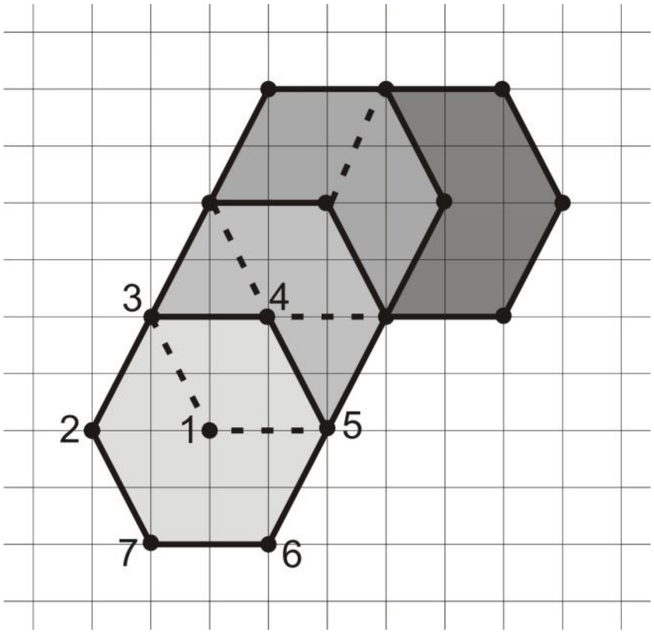


Fig.1. Hexagonal coarse search

In the inner search, the optimum motion vector is expected to lie within the eight points around the global optimum point. Searching all the eight points of the inner search is computationally inefficient. There exist a strong correlation between the inner eight search points and the six search points in the hexagon. Most of the fast inner search algorithms [8], [9] and [10] exploit this correlation to find the optimum motion vector without searching all the eight inner search points. Most of the hexagonal search algorithms focus on improvement of either coarse search or inner search. In the proposed algorithmwe improved both coarse search as well as inner search. This paper is organized as follows. In section II, proposed directional hexagon-based block search is explained. SectionIII briefly discusses improvedinner search. Section IV shows the experimental results in a variety of video sequences to verify the efficiency of the proposed algorithm followed bythe concluding remarks in Section V.

II. DIRECTIONAL HEXAGON-BASED SEARCH

The hexagonal search pattern consists of initial seven search points. After locating Minimum Block Distortion (MBD) at the corners of the hexagon, it searches for the three new non-overlapping search points. But there is strong continuity in the motion vector field around the block with MBD. Thus by exploiting this continuity we can further reduce these non-overlapping search points. The proposed directional hexagon-based block search algorithm can be explained using Fig. 2. Initially suppose if point 4 is having MBD, then center of the hexagon is shifted to this point. The new non overlapping search points can be reduced from three to two by checking the distortion (D) in the neighboring points of the MBD point as the optimum matching block is expected to lie in the direction of the minimum D point. In this case, the D points will be 3 & 5. If minimum D point is 3 then it will search for points 8 & 9 only as new search point. Otherwise, it will search for points 9 & 10. In short, it will further search only points which lie in the area bounded by lines passing through points 1-4 and 1-5 as shown in Fig. 2. Considering point 5 as the minimum D point, algorithm will search for points 9 & 10. Now, if MBD point is 9 then it will check only one neighboring D point i.e. point no. 10. This is because other neighboring D point i.e. point 8 lies outside current bounded area. Thus it will consider point 10 as the minimum D point. Then it searches for points 12 and 13 which lie nearer to minimum D point and also within the converged area bounded by lines 4-12 and 4-10. This process is continued till the MBD point is found as the center which is point 12 in this case. Thus this algorithm reduces the search area and consequently the numbers of search points.

Stepwise procedure for the algorithm is as follows.

Step 1) Calculate the Sum of Absolute Difference (SAD) for the seven search points of the hexagon. If the MBD is centre itself thenproceedto step 4. Otherwise, shift the center of the hexagon to the MBD point.

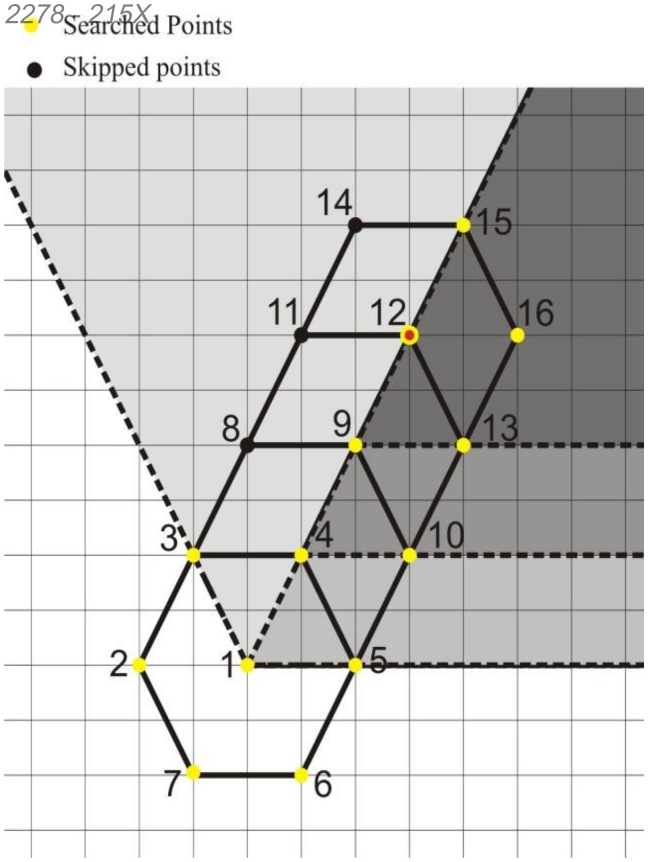


Fig.2 Directional Hexagon-based search pattern

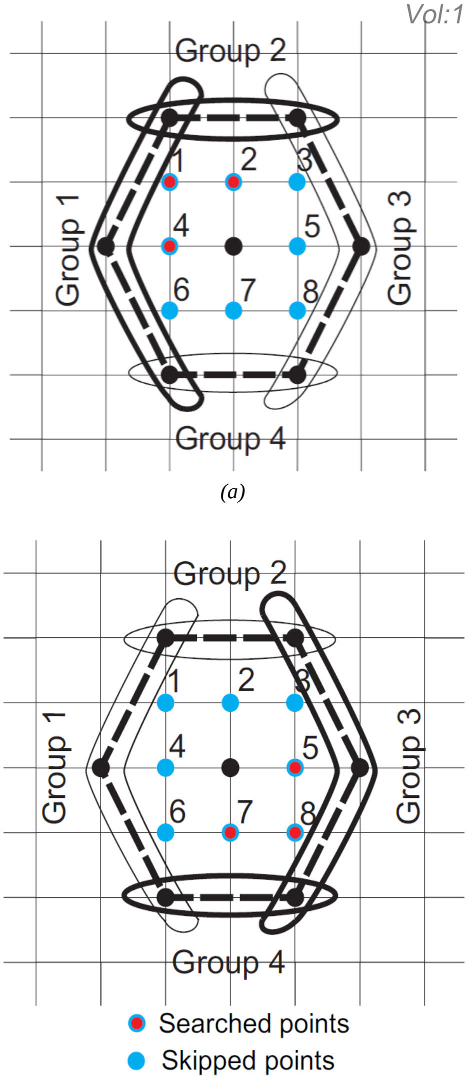
Step 2)Compare the two D points (but lying in the newly converged area and if only one lies then declare it asminimum D point) neighboring to the MBD point. Findout minimum D point.

Step 3)Calculate the SAD for the two search points (lying in the convergence area) which are nearer to the minimum D among the three non-overlapping search points of the newly formed hexagon centering MBD point. Compare the search points. If MBD is the center itself then proceed to Step 4. Otherwise, continue to Step 2.

Step 4) Switchfrom the largecoarse search to fine-resolution inner search to find out absolute motion vector.

III. MODIFIED INNER SEARCH

Full inner search requires checking eight points and thus computationally inefficient. Original HEXBS searches for only 4 points but does not map all the eight points. These eight checking points can be further reduced by exploiting the distortion information of outer six search points of the hexagon. Based on the monotonic distortion characteristic in



(a)

(b)

Fig. 3 (a) Three inner points nearest to Group 1 and Group 2 are checked. (b) Three inner points nearest to Group 3 and Group 4 are checked

the localized area around the global minimum, we propose to check only a portion of the inner search points that are nearer to the checked points with smaller distortions, which can save more than half of the eight search points inside. In order to reduce the number of searches, the fast inner search use four groups as shown in Fig 3. Group 1 and 3 includes three points and group 2 &4 includes two points. For each group, we define a group distortion by summing the distortions of all the points within the group. Now comparison is done between group 1 & 3 for finding smallest distortion group between them. Also group 2 & 4 are compared for finding smallest distortion group. The area near to two groups with the smallest group distortion is considered as the region whereoptimum matching point is expected to lie. Therefore we focus the inner search just in the region near to the groups with the smallest group distortion. Thus for the different combinations of groups, we have different locations of inner points to be searched. As shown in Fig. 3(a), group 1 is smallest distortiongroup between 1 & 3 and group 2 is smallest distortion group between 2 & 4. As group 1 and 2 wins, it searches for inner points labeled 1, 2 & 4. If group 3 and 4 wins as shown in Fig. 3(b) then it searches for inner points labeled 5, 7 & 8. Hence, we reduce the inner points to be searched from eight to only three points reducing lots of computation.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed directional hexagonal search, we compare it against the HEXBS in terms of number of search points and PSNR. Parameters used in the experiment include: sum of absolute difference (SAD) for distortion measurement, block size of 8×8, search window size of ±7. Seven standard video sequences “Garden” (352×288, 115 frames), “Football” (352×288, 260 frames), “Stefan” (352×288, 300 frames), “Foreman” (352×288, 300 frames), “Coastguard” (352×288, 300 frames), “Mobile” (352×288, 300 frames), “Silent” (352×288, 300 frames).

The average number of search points per block and average PSNR values are compared in table I and II, respectively for seven different video sequences. Experimental values show that the proposed algorithm significantly reduces the average search points per block as compared to the previous HEXBS with slight reduction in PSNR value. Table I shows that the average number of search points per block is reduced by about 1.02 over the previous HEXBS. For the analysis of speed improvement, speed improvement ratio (SIR) is calculated. For “Garden”, where motion is large, average SIR is 15.6%.For “coastguard” where motion is medium, average SIR is 10.02%. For “Silent” where motion is less, average SIR is9.37%. This shows that, proposed algorithm achieves better speed improvement over HEXBS for video with large motion. The larger the motion in a video sequence, the larger the speed improvement rate of DHEXBS overHEXBS will be.

Fig. 4 plots a frame-by-frame comparison in number of search points, for the proposed directional HEXBS and the original HEXBS applied to “Garden”. It is seen that number of search points per frame reduced by about 2800 which is a significant improvement. Fig. shows PSNR values for different frame numbers. PSNR remain nearly same for most of the frames except for the frame numbers 41, 47 and 72 where PSNR drops noticeably. Apparently, all the experimental results substantially justify the faster performance of the proposed directional HEXBS algorithm as compared with the original HEXBS.

TABLE I

AVERAGE NUMBER OF SEARCH POINTS PER BLOCK WITH RESPECT TO DIFFERENT METHODS AND DIFFERENT VIDEO SEQUENCES

	Garden	**Football**	**Stefan**	**Foreman**	**Coastguard**	**Mobile**	**Silent**
HEXBS	13.0746	12.9780	10.9046	10.7961	10.7314	10.5902	10.6153
DHEXBS	**11.3103**	**11.2172**	**9.8630**	**9.7961**	**9.7535**	**9.6580**	**9.6768**

TABLE II

AVERAGE PSNR FOR DIFFERENT METHODS AND DIFFERENT VIDEO SEQUENCES

	Garden	**Football**	**Stefan**	**Foreman**	**Coastguard**	**Mobile**	**Silent**
HEXBS	10.3768	13.1243	20.2530	26.4940	23.9399	20.6443	32.2734
DHEXBS	**10.3399**	**12.9819**	**19.9209**	**26.3505**	**23.8577**	**20.6451**	**32.2002**

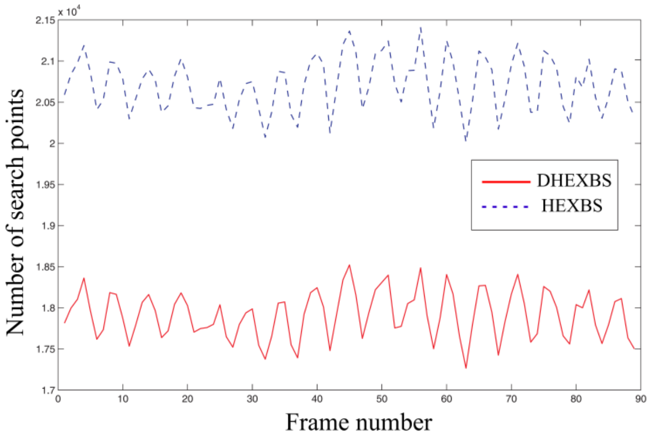


Fig. 4 Frame by frame comparison of search points for "Garden" sequence for both DHEXBS and HEXBS

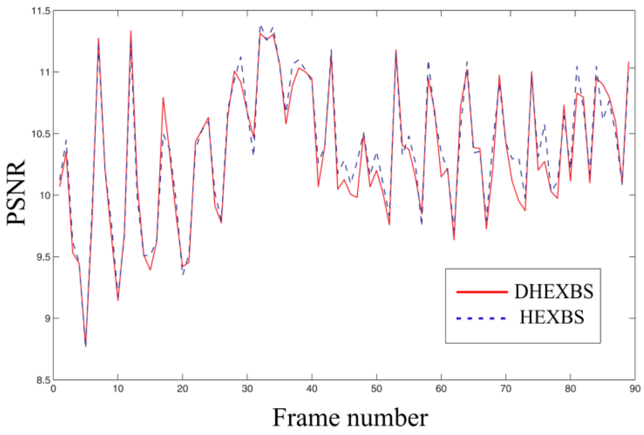


Fig. 5 Frame by frame comparison of PSNR for "Garden" sequence for both DHEXBS and HEXBS

V. CONCLUSION

We have developed a novel directional HEXBS over original HEXBS. Most of the hexagonal search algorithms focus on improvement of either coarse search or inner search. In the proposed algorithm we improved both coarse search as well as inner search. Hence net effect is substantial drop in the number of computations per frame. From theexperimental results, it can be concluded that the proposed algorithm outperforms for videos with large motion. The experimental results have convincingly demonstrated the superiority of the proposed directional HEXBS to the original HEXBS. Thus due to significant reduction in number of computations, this algorithm can be used in applications where low computation is needed.

REFERENCES

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

[2] *R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438–442, Aug. 1994.*

[3] *X. Jing and L. P. Chau, "An efficient three-step search algorithm for block motion estimation," IEEE Trans. Multi., vol. 6, no. 3, pp.435-438, June 2004.*

[4] *L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits and Systems for Video Technology, vol. 6, pp. 419-423, Aug. 1996.*

[5] *J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol, vol. 8, pp. 369–377, Aug. 1998.*

[6] *S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," IEEE Trans. Image Processing, vol. 9, pp. 287–290, Feb. 2000.*

[7] *C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol, vol. 12, pp. 349–355, May 2002.*

[8] *C. Zhu, X. Lin and L. P. Chau, "An enhanced hexagonal search algorithm for block motion estimation, IEEE Int. Symp. Circuits and Systems (ISCAS), Bangkok, Thailand,vol. 2, pp.392-395, May 2003.*

[9] *C. Zhu, X. Lin, L. Chau and L.-M. Po, "Enhanced hexagonal search for fast block motion estimation," IEEE Trans. Circuits and Systems for Video Technology, vol. 14,pp. 1210-1214, Oct. 2004.*

[10] *Chorng-Yann Su,Yi-Pin Hsu, and Cheng-Tao Chang, "Efficient hexagonal inner search for fast motion estimation," IEEE Int. Conf. Image Processing (ICIP), vol. 1, pp. 1093-1096, Sept. 2005.*