Enhanced Diamond Search Using Four-Corner-Based Inner Search For Fast Block Motion Estimation

Lai-Man Po, Chi-Wang Ting and Ka-Ho Ng
Department of Electronic Engineering
City University of Hong Kong, Hong Kong, China.

Abstract—**This paper proposes a very efficient and simple fine-resolution inner search for the well-known diamond search (DS) algorithm. Based on the local unimodal error surface assumption, a new 4-corner-based inner search technique is proposed to speedup the original DS in both large and small motion environments by exploiting the group-distortion information of some evaluated points. Experimental results show that the enhanced diamond search (EDS) is faster than the original DS up to 30% with negligible PSNR degradation.**

I. INTRODUCTION

Motion estimation (ME) is an integral part for most motion-compensated video coding standards. It is a process to remove the temporal redundancy between successive frames by referring current pixels to previously decoded frames. However, in order to find the best reference, the conventional exhaustive method, full search (FS), would introduce very high computational complexity. Over the last two decades, numerous fast motion estimation algorithms have been proposed to tackle this problem. Among them, the most popular class is the block-matching algorithm (BMA) using a fixed set of search patterns which makes use of a well-known assumption – unimodal error surface assumption. It means the matching error is monotonically decreased towards the global minimum. In the 1980s, many fast BMAs were developed based on this assumption and some well-known examples are three-step search (3SS) [1], 2D-logarithmic search, and conjugate directional search. They all can achieve substantial computational reduction but with a drawback of modest estimation accuracy degradation. This is mainly due to whole matching error surface is not monotonically decreasing towards the global minimum and there exists a lot of local minimums. These algorithms are, therefore, easily trapped into a local minimum with use of uniformly distribution search points in the early stage.

In the early 1990s, experimental results [2, 3] shown that the block motion field of real world image sequence is usually gentle, smooth, and varies slowly. It results in a center-biased global minimum motion vector distribution instead of a uniform distribution. This implies that the chance to find the global minimum is much higher within the center 4x4 region of the search window. To make use of this characteristic, center-biased BMAs were then proposed using smaller searching patterns with search points much nearer the center to speedup the estimation time while the average prediction accuracy is also improved especially for the slow motion sequences. Well-known examples of this category are new three-step search (N3SS) [2], four-step search (4SS) [3], diamond search (DS) [4], cross diamond search (CDS) [5], hexagon-based search (HS) [6], etc. Recently, the enhanced hexagon-based search (EHS) algorithm [7] using a 6-side-based fast inner search was proposed to further speedup the original HS algorithm. In EHS, the searching algorithm is divided into two parts: (1) a low-resolution search which maximizes the coverage of the searching area and locates a small region where the motion vector would lies on; and (2) a high-resolution search or so-called inner search which finds the best motion vector inside the small region. In addition, the two major ideas of EHS algorithm raised are (1) further speedup could be achieved on saving search points for inner search; and (2) the inner search speedup can be made use of a local unimodal error surface assumption (LUESA) by checking a portion of the inner search points errors. It is because the final inner search minimum should have much higher chance with smallest sum of distortion for the search points surrounding it. This is statistically strong in a small region with the localized area around the global minimum. Based on the LUESA, the EHS only checks a portion of the inner search points that are nearer to the evaluated points with smaller distortions, which results in saving more than half of the inner checking points. In this paper, an enhanced diamond search (EDS) algorithm is proposed for speeding up the original DS using a more efficient and regular 4-corner-based fast inner search as compared with the 6-side-based inner search for EHS. In the following section, the 6-side-based fast inner search is first discussed and hence some principles of grouping the evaluated points are deduced, which favor the proposed algorithm to reduce the necessary checking points. After that, an analysis of the motion vector distribution in the inner search area is discussed. The simulation results and comparison of EHS are presented at the end with a brief conclusion.

II. INNER SEARCHES

Based on the LUESA, the final inner search minimum should have much higher chance with smallest sum of distortion for the search points surrounding it. Thus, the inner search should be focused on the points inside the coarse search pattern of the final search step. Since this area is relatively small and close to a minimum point, there is a very strong correlation between the neighbors. It is reasonable to believe the surrounding evaluated points should give sufficient information to identify the minimum point in the inner area. An efficient method to utilize the information is to group the points and measure their group distortions. Figure 1(a) show the search point configuration of a hexagonal search pattern (0 to 6) and its eight inner points (*a* to *h*). The 6-side-based inner fast search divides the hexagon into six groups based on the side positions and each group consists of two points. By comparing the group distortions, the method only checks the points near to the minimum distortion group. The corresponding groups and inner points are demonstrated in Figure 1(a). For example, if group 1 has the minimum distortion, then points *e* and *d* will be checked, or if group 2 has the minimum, then points *e*, *a*, and *f* will be checked. While this kind of grouping seems sensible, we find there are some disadvantages. First, the distances between the inner points and their corresponding groups are not equivalent to each other. An example is shown in Figure 1(b) where the group internal distance (GID) of point *b* (D_{b4}+D_{b5}) is longer than that of point *e* (D_{e1}+D_{e2}). Our analysis shows that the distance is a good indication of the correlation between the inner points and their groups as it is linearly proportional to the block distortion in the localized area around the global minimum, as shown in Figure 2. Therefore, the 6-side-based grouping may yield an inaccurate prediction. Second, the weights of the inner points are different. For the points *a* and *c*, they are only correlated to one group which is group 2 and group 5 respectively, while the others are correlated to two nearby groups. This may either lead to the redundancy of checking the over weighted points or missing the under weighted points. To sum up, we suggest two principles for grouping the evaluated points:

1. The GID of the inner points should be equal to each other, and as short as possible.
2. For no biasing, each inner point should be represented by the same number of groups and each group should also have the same number of evaluated points.

Based on these principles, we find the fast inner search is very suitable for the original DS algorithm due to its regularity and symmetric shape. The search points of a large diamond search pattern (LDSP) could be easily divided into four corner groups with four points per group as shown in Figure 3(a). Each inner point is surrounded by four nearest evaluated points with exactly the same GID. Similar to the 6-side-based inner search, a group distortion is used by summing the distortions of all the points within the corner group. However, as all four groups include the central point 0, it is not necessary to be counted for real situation. Then the inner point within the smallest distortion corner group should be searched as shown in Figure 3(b). It has much higher chance to be the final minimum point as compared with the three other points. This is a very efficient inner search for DS as it only requires 1 additional search point and the overhead of the group distortion computations and comparisons are negligible. In addition, this inner search pattern is much more regular and simple in terms of the GID and weights as compared with the 6-side-based inner search, which requires 2 or 3 additional search points with six group distortion comparisons.

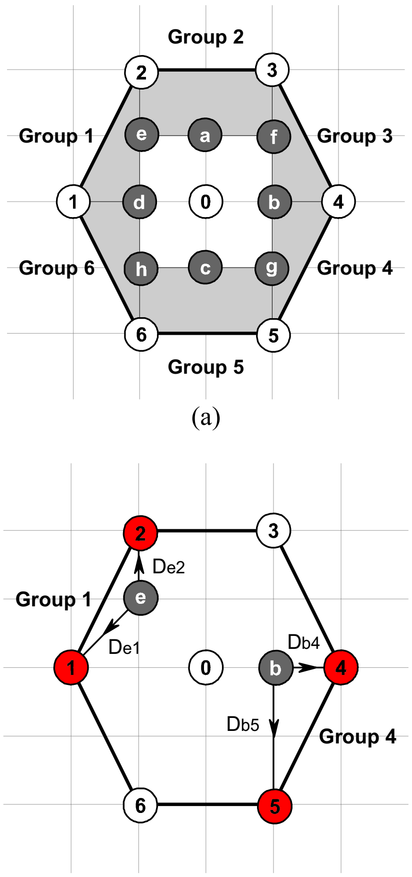
III. Enhanced Diamond Search Algorithm

With the use of the proposed 4-corner-based fast inner search, the enhanced diamond search is summarized as follow:

Step 1: Set the minimum distortion point to the center of the search area (0, 0)

Step 2: A minimum distortion point is found from the 9 checking points of LDSP with the center at the previous minimum distortion point. If the new minimum distortion occurs at the center of LDSP, go to Step 3; otherwise this step is repeated again.

Step 3: Compute the 4-corner-based group distortions of the LDSP and find the minimum group distortion. Based on the minimum group distortion, compute the distortion of the additional search point within the corner and then identify the new minimum distortion point, which is the final motion vector.



(a)

(b)

Figure 1. (a) This figure shows the configuration of a hexagonal search pattern (0-6) and its inner points (a-h). The outer points are divided into six groups based on the sides. (b) The 6-side-based grouping causes irregular group internal distance (GID).

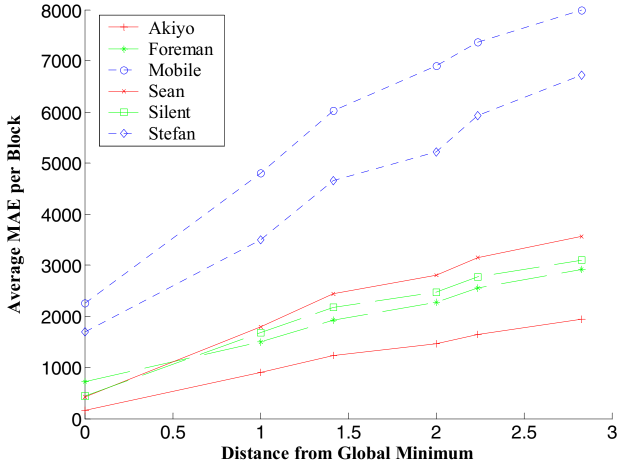


Figure 2. The graph shows an approximately linear relation between the average MAE per block and the distance from the global minimum which is found by full search

Table 1. The inner motion vector distribution of the original diamond search.

Position	Akiyo	Foreman	Mobile	Sean	Silent	Stefan
0	96.8	57.5	42.0	90.3	90.4	64.8
a	0.9	8.0	11.3	3.2	2.3	3.7
b	0.6	14.2	7.0	1.8	2.6	11.5
c	1.1	7.0	14.4	3.2	2.4	6.2
d	0.6	13.4	25.4	1.5	2.4	13.8

Besides the new inner search, an early termination method for the inner search is also proposed. It is reasonable to assume if the current sub-optimal distortion is already small enough, it is not necessary to spend further effort to search another optimum which simply does not reflect any difference on the visual quality. The motion vector distribution of the inner search area is summarized in Table 1. Among the last 5 checking points, around 42% to 97% motion vectors are located at the position of point 0. This locally center-biased characteristic is especially obvious for the sequences containing large amount of static background e.g. Akiyo, Sean and Silent. This method terminates the inner search if the current minimum distortion (point 0) is smaller than a threshold. To maintain the prediction accuracy, a relatively low threshold value of 384 is selected for a mean absolute error (MAE) block distortion measure. The threshold is set to keep the picture quality unchanged, but further reduces the minimum number of inner checking points to zero.

IV. Experimental Results

To demonstrate the performance of the proposed algorithms, the EDS (with fast inner search only) and EDS+ (with fast inner search + early termination) is compared with the original DS. Moreover, the experimental results of EHS (with fast inner search only) and original HS are listed together as a reference to show the efficiency of the proposed fast inner search. The simulation is performed on nine representative CIF sequences from three difference classes of MPEG-4; ClassA: Akiyo, Container, Sean; ClassB: Foreman, News, Silent; ClassC: Mobile, Stefan, Table, which contain various real world motions and objectively justify our method. The simulation settings are: 100 frames, 16x16 block size, ±16 search window, and MAE block distortion measure. Experimental results are tabulated by two testing criteria – average PSNR per frame and average number of search point per block. To compare the visual quality, Table 2 shows that the PSNR changes of EDS and EDS+ against DS are from –0.002dB to –0.048dB which are surely negligible. The difference between EDS and EDS+ is smaller than 0.009dB that the early termination does not ruin the picture quality at all. On the other hand, the PSNR change of EHS against HS is from 0.326dB to –0.415dB. An increase of PSNR is shown, because the original HS only checks a small cross (4 points) around the center. The results show that our fast inner search is more accurate and has a smaller impact on picture quality as compared with the 6-side-based fast inner search.

To show the speed performance, we have two different comparisons. The number of search points reflects the absolute speed of different algorithms, and the speed improvement rate (SIR) reflects the speedup percentage relative to the original search. The SIR of method 1 over method 2 is defined by $SIR = (N_2-N_1)/N_2 \ x \ 100\%$ where N_1 is the number of search point used by method 1 and N_2 is that by method 2. From Table 3, the EDS saves around 2.5 search points on average, which is a bit less than the theoretical result of 3 search points. It is because these additional inner points will be checked if their corresponding groups are out of the frame boundary. For EDS+, it saves around 2.5 to 3.7 search points. It is clear that the early termination can efficiently reduce one more search point for most low to medium motion sequences, e.g. Akiyo, Silent, etc. Besides, the EHS saves around 1.0 to 1.4 points which are also less than the theoretical result due to the same reason. Finally, Table 4 shows the speed improvement rate over the original algorithms. The SIR of EDS and EDS+ over DS are around 14% to 20% and 16% to 30% respectively, while that of EHS over HS is around 9% to 14%, which is significantly smaller than that of the proposed method. The experimental results justify the ideas that inner search are more efficient when applied to DS. The superior performance on both accuracy and speed is also demonstrated for the new fast inner search.

V. Conclusions

The local unimodal error surface assumption facilitates the fast inner search as a complement of the low-resolution search to further reduce the overall checking points. By properly grouping the evaluated points, the group distortion could be used to predict the minimum of the inner points. In this paper, two grouping principles are suggested based on an analyzed statistic of the inner area. It is found that the regularity and symmetric shape of diamond search is very suitable for using fast inner search. Therefore, the enhanced diamond search (EDS) algorithm is proposed with a 4-corner-based fast inner search. It is shown that the EDS significantly reduces 16% to 30% computations over the original DS. At the same time, a negligible PSNR loss is kept below 0.05dB. The new algorithm also out-performs the EHS on both accuracy and speed improvement rate.

Acknowledgment

The work described in this paper was substantially supported by a grant from City University of Hong Kong, Hong Kong SAR, China. [Project No.7001699].

References

[1] T.Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, “Motion compensated interframe coding for video conferencing,” in *Proc. Nat. Telecommun, Conf.,NO, L.A.,* Nov-Dec 1981, pp. G5.3.1-G.5.3.5.

[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. 438–443, Aug. 1994.

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

[4] 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, no. 4, pp. 369–377, Aug. 1998.

[5] C. H. Cheung and L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.12, No. 12, pp. 1168-1177, Dec. 2002.

[6] 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.

[7] C. Zhu, X. Lin, L. P. Chau, and L. M. Po, “Enhanced Hexagonal Search for Fast Block Motion Estimation,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, Issue 10, pp. 1210 - 1214, Oct. 2004.

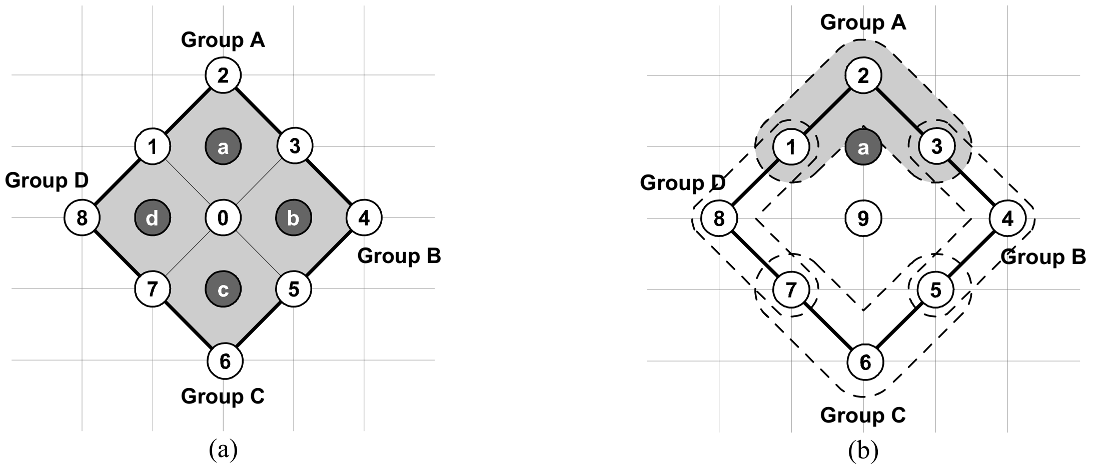


Figure 3. (a) The configuration of large diamond search pattern (0-8) and its inner search points (*a*-*d*). (b) Point *a* is searched finally as group A has the smallest group distortion. .

Table 2. Average PSNR per frame for different algorithms on nine sequences.

PSNR	Akiyo	Container	Sean	Foreman	News	Silent	Mobile	Stefan	Table
DS	42.817	38.331	38.980	33.088	37.189	35.783	23.805	23.837	27.643
EDS	42.791	38.329	38.957	33.067	37.176	35.779	23.757	23.835	27.633
EDS+	42.790	38.328	38.956	33.065	37.167	35.779	23.757	23.835	27.632
HS	42.556	38.331	38.858	32.340	37.038	35.714	23.633	23.717	27.259
EHS	42.314	38.329	38.736	32.666	36.997	35.747	23.217	23.724	27.293

Table 3. Average search points per block for different algorithms on nine sequences.

Pts.	Akiyo	Container	Sean	Foreman	News	Silent	Mobile	Stefan	Table
DS	12.284	12.375	12.758	16.310	12.670	13.350	13.271	17.794	17.104
EDS	9.860	9.938	10.324	13.828	10.245	10.930	10.826	15.318	14.549
EDS+	8.614	8.820	9.208	13.643	9.062	9.872	10.772	14.969	14.390
HS	10.346	10.419	10.569	12.659	10.588	10.958	10.704	13.758	13.304
EHS	9.373	9.012	9.612	11.346	9.499	9.751	9.459	12.353	11.999

Table 4. Speed improvement rate comparisons for 2 sets of algorithms with 4-corner-based fast inner search and 6-side-based fast inner search respectively.

SIR	Akiyo	Container	Sean	Foreman	News	Silent	Mobile	Stefan	Table
EDS and EDS+ over the Original DS									
EDS	19.734	19.687	19.078	15.220	19.141	18.125	18.424	13.918	14.941
EDS+	29.881	28.723	27.831	16.353	28.478	26.050	18.832	15.875	15.866
EHS over the Original HS									
EHS	9.406	13.504	9.062	10.371	10.293	11.020	11.629	10.214	9.810