Novel Point-Oriented Inner Searches for Fast Block Motion Estimation

Lai-Man Po, Chi-Wang Ting, Ka-Man Wong, and Ka-Ho Ng

Abstract—**Recently, an enhanced hexagon-based (EHS) search algorithm was proposed to speedup the original hexagon-based search (HS) using a 6-side-based fast inner search. However, this 6-side-based method is quite irregular by inspecting the distance between the inner search points and the coarse search points that would lower prediction accuracy. In this paper, a new point-oriented grouping strategy is proposed to develop fast inner search techniques for speeding up the HS and diamond search (DS) algorithms. Experimental results show that the new HS and DS using point-oriented inner searches are faster than their original algorithms up to 30% with negligible peak signal-to-noise ratio degradation.**

Index Terms—**Diamond search, fast motion estimation, hexagon search, inner search.**

I. Introduction

VIDEO coding is a sophisticated process that converts the digitized raw video data into a comprehensive representation. Motion estimation (ME) plays an important role in video coding. However, it is also well-known that the ME process introduces extremely high computational requirement for the video encoder if the conventional full-search (FS) method is used. 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 decreasing 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], two-dimensional (2-D) logarithmic search [2], and conjugate directional search [3]. They can all achieve substantial computational reduction, but with the drawback of modest estimation accuracy degradation. It is because the matching error surface is not monotonically decreasing and the surface consists of many local minimums. These algorithms are, therefore, easily trapped into a local minimum with the use of uniformly distributed search points in the early stage.

In the early 1990s, experimental results [4], [5] showed that the block motion fields of real-world image sequences are usually gentle, smooth, and vary 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 4×4 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 to the center. Well-known examples of this category are the new three-step search (N3SS) [4], four-step search (4SS) [5], block-based gradient-descent search (BBGDS) [6], diamond search (DS) [7], [8], cross diamond search (CDS) [9], hexagon-based search (HS) [10], cross-diamond-hexagonal search (CDHS) [11], and efficient three-step search (E3SS) [12], etc.

Recently, the enhanced hexagon-based search (EHS) algorithm [13] 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: a low-resolution search and a high-resolution search. The former is sometimes called coarse search, which try to maximize the coverage of the search for the searching area with a relatively large and sparse search grid. The aim of this part is to locate a smaller region where the optimal motion vector possibly lies on. The high-resolution search or so-called inner search uses a small and compact search grid to find the best motion vector inside the small region. Most of the earlier efforts are spent on reducing the searching points for coarse search with use of different patterns such as square, diamond, and hexagon. However, 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 make use of a local unimodal error surface assumption (LUESA) by checking a portion of the inner search points. It is because the final inner search minimum should have a much higher chance with the 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 with smaller group distortions, which results in saving more than half of the inner checking points. However, we find that the 6-side-based method is quite irregular and such a structure would result in lower prediction efficiency. In order to tackle this problem, a point-oriented grouping principle is proposed in this paper to develop more efficient inner search techniques for HS and DS.

In Section II, we will first introduce the characteristics of DS and HS and then introduce the 6-side-based inner search algorithm for EHS. To deduce principles for grouping the coarse search points, a few analyses of the neighboring correlations

Manuscript received November 2, 2005; revised May 5, 2006. This work was supported by a grant from City University of Hong Kong, Hong Kong SAR, China (Project No. 7001699). The associate editor coordinating the review of this paper and approving it for publication was Dr. Pascal Frossard.

The authors are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, China (e-mail: eelmpo@cityu.edu.hk; cwting@ee.cityu.edu.hk; kmwong@ee.cityu.edu.hk; kahomike@gmail.com).

Color versions of Figs. 4 and 5 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2006.886330

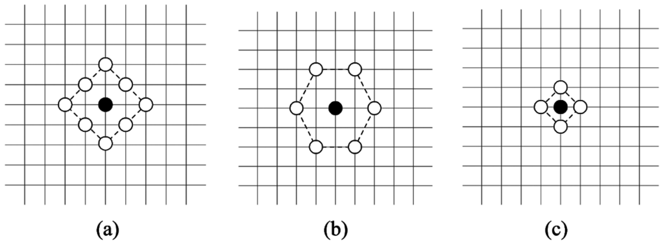

Fig. 1. Three search patterns used by diamond search and hexagonal-based search: (a) large diamond search pattern (LDSP), (b) large hexagon-based search pattern (LHSP), and (c) small diamond search pattern (SDSP) or small hexagon-based search pattern (SHSP).

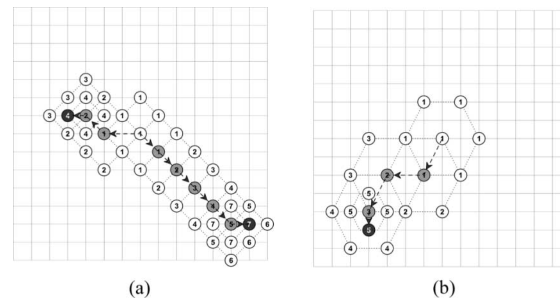

Fig. 2. (a) Two examples of DS. (b) Example of HS.

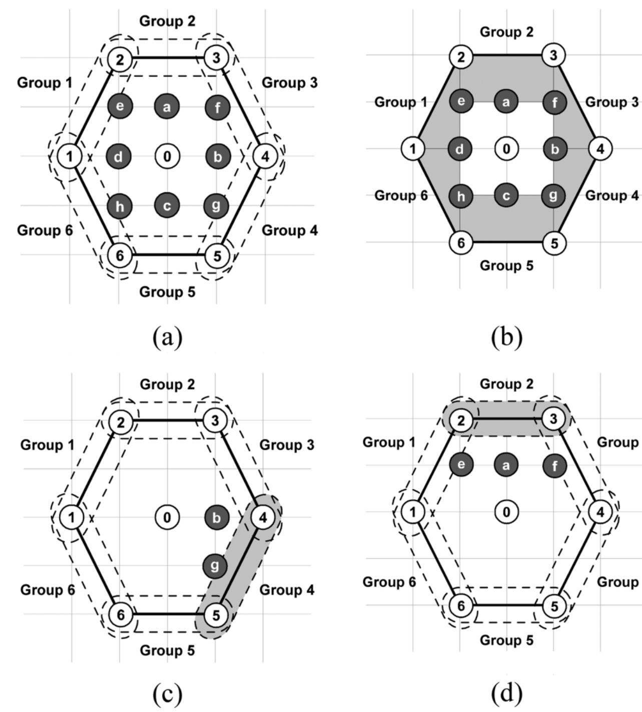

Fig. 3. Configurations and examples of the 6-side-based inner search: (a) groups of evaluated points; (b) defining shaded region for each group; (c) group 4 wins and checks point "b" and "g"; (d) group 2 wins and checks point "a", "e", and "f".

based on search point's distance and distortion are given in Section III. Based on these new principles, two new algorithms—EHS using point-oriented inner search and enhanced DS—are proposed in Section IV. The simulation results are presented in Section V, and finally, the conclusion is addressed in Section VI.

II. From Diamond Search to Enhanced Hexagon-Based Search

A. *Diamond Search and Hexagon-Based Search*

The DS [7], [8] was first proposed in 1998 using two search patterns called large diamond search pattern (LDSP) and small diamond search pattern (SDSP) as shown in Fig. 1(a) and (c). The search is started by placing the LDSP in the center of the search window. The center of the next search grid is shifted to the minimum block distortion measure (BDM) point. LDSP is used continuously for successive search until the minimum BDM is located in the center of the grid, then SDSP is used for the final search step. Two examples of DS search paths are shown in Fig. 2(a). Similarly, the HS also has two search patterns but with hexagonal shape as depicted in Fig. 1(b) and (c). These two search patterns are called large hexagon-based search pattern (LHSP) and small hexagon-based search pattern (SHSP). The HS adopts the same searching strategy as DS, in which a coarse search pattern keeps moving its center to the newly identified minimum point. When the minimum occurs in the center of search pattern, a diminished pattern is used for inner search. An example of an HS search path is shown in Fig. 2(b). As you can see, the SHSP is the same as the SDSP, but conceptually regarded as a hexagon for convenience. The purposes of turning to a hexagonal shape are that it is closer to the shape of a circle than a diamond and it also requires a lower number of search points.

B. *Enhanced Hexagonal-Based Search*

The efforts of 3SS, 4SS, DS, and HS are spent on reducing the number of searching points for coarse search using different patterns such as square, diamond, and hexagon. However, an exhaustive search is still often applied to the inner small region. It is because people have a common misconception that the long-range coarse search should be much heavier than the short-range inner search in term of computational complexity. However, this is not always true. By the continuous improvement of search patterns and strategies, the number of search points required for the coarse search has been dramatically decreased.

The EHS is an improved algorithm of the original HS using the 6-side-based fast inner search technique. The 6-side-based fast inner search is a group-oriented method. That means it first divides the evaluated coarse search points into a number of groups and accordingly assigns the inner points to different candidate groups. Fig. 3(a) shows the groups for the LHSP (group 1 to group 6) and the eight inner points ("a" to "h"). This method separates the hexagon into six groups by the sides, and so each group consists of two evaluated points. A group distortion is calculated by summing up all the individual distortions in the group. By comparing the group distortions, this method only checks the points near to the minimum distortion group. The corresponding groups and confined regions are demonstrated in the Fig. 3(b). Two examples are shown in Fig. 3(c) and (d), if group 4 has the minimum distortion,

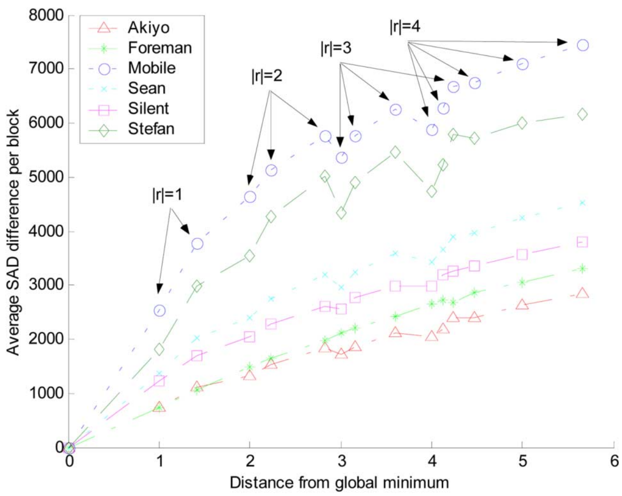

Fig. 4. Analysis of the relation between block SAD and distance.

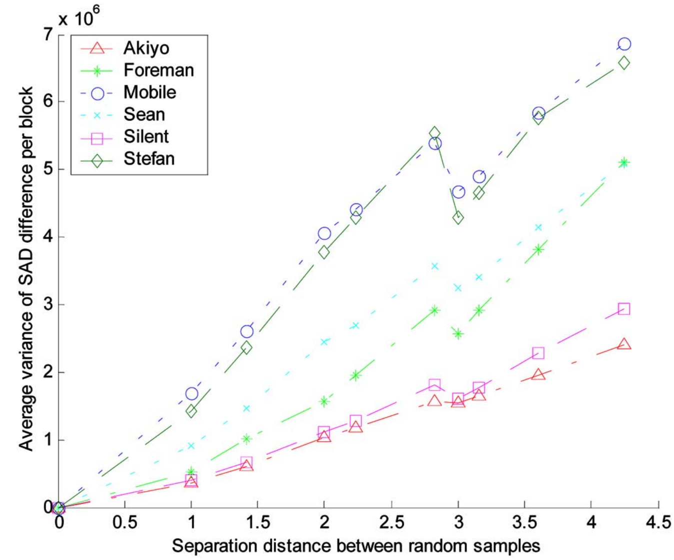

Fig. 5. Variance of block SAD difference for the random sample pairs taken from the 4×4 region around the global minimum.

then point "b" and "g" will be checked, or if group 2 has the minimum, then point "e", "a", and "f" will be checked.

While the above grouping method seems sensible, we find that it cannot optimize the prediction result for each inner point. Instead of using such a group-oriented method, we believe a point-oriented method should yield better prediction accuracy. That means each inner point is considered as an individual. A dedicated group is formed by selecting tailor-made neighbors for an inner point. To find out the principles of determining these points, a few analyses are conducted with respect to distortion, distance, and correlation.

III. Point-Oriented Inner Search Strategy

A. Locally Unimodal Error Surface

The locally unimodal error surface assumption is the basis for inner search techniques. Within a localized region, if the block-matching error is smoothly increased in a monotonic way apart from the global minimum point, then the distortions of the other points can be easily approximated by their neighbors' distortions and separation distances. In other words, we can find out an approximate value of the distortion for a particular point without calculating its actual BDM. Here, we first define a metric for this purpose as

$$\overline{D}(d) = \sum_{i=1}^{N} \frac{D_i(d)}{N} = \sum_{i=1}^{N} \frac{\mathrm{SAD}_i(d) - \mathrm{SAD}(0)}{N}. \quad (1)$$

For (1), $\overline{D}(d)$ is the mean value of the sum of absolute difference (SAD) between the global minimum and any other points with a separation distance d. Accordingly, $D_i(d)$ is the SAD difference of the sample i from the global minimum, SAD (d) is the SAD of a point with distance d from the global minimum, and N is the total number of samples at the distance. The function is visualized and plotted against the distance in Fig. 4. Since the set of testing sequences has similar motion contents from class to class, only six representative sequences of CIF format are picked for the analysis. The first 100 frames, totally 39 600 blocks, are used to build the statistic.

In Fig. 4, we find that the SAD difference gently increases with the distance, but not monotonically on the whole. For the high-motion sequences, "Mobile" and "Stefan", there are two obvious valleys at $d = 3$ and $d = 4$, and they are referred to as the local minima of a motion search. The main reason for the drops is that the SAD is normally lower for a vertical or horizontal displacement rather than a diagonal displacement. However, for the localized inner search region, we may just consider the fore part of the curve (i.e., $|r| \leq 2$), and the SAD difference will then become approximately linearly proportional to the distance. This characteristic is not only valid for the global minimum found by FS, but should also be valid for other minimum points found by the suboptimal search method. Therefore, we may generally assume the distortion is linearly increased in the targeted inner search area.

B. Correlation and Distance

The analyses reveal that the distance between two points not only affects the distortions, but also influences the correlation of them. An experiment was set up to verify the correlation between any two points with a separation distance d. Ten pairs of samples are randomly selected from each macroblock for a particular distance. For inner search analysis, the samples are restricted in a radius $|r| = 2$ region around the global minimum. Let $D_i''(d)$ be the SAD difference for the pair i, and $\overline{D''}(d)$ be the mean SAD difference for all the samples. Then the variance is defined as

$$V(d) = \sum_{i=1}^{N} \frac{(D_i''(d) - \overline{D''}(d))^2}{N}. \quad (2)$$

The variance $V(d)$ is a measure of how spread out the distribution of the block difference from the mean is. Fig. 5 shows the variance increases linearly along with the distance, and has a sudden drop when $d = 3$ due to the same reason mentioned in the last subsection. The higher the variance is, the lower the correlation which exists between the pair. As the samples are randomly taken from the 4×4 region, the behavior shown by the figure can be generalized and applied to any position inside. It delivers an important message that the neighbors do not give a significant meaning to their central point when they are separated too far away.

Assumptions and Principles for Inner Search

To sum up the analysis results, a point-oriented inner search method requires two basic assumptions. For the localized inner search region:

1) there is only one minimum point;
2) the distortion increases linearly apart from the minimum.

Based on these assumptions, one's distortion can be approximated by the average of its neighbors with the following grouping principles:

1) the number of neighbors should be as many as possible;
2) neighbors are within the shortest distance;
3) the distortion of each neighbor is normalized for calculating group distortion;
4) each group has the same size for comparing group distortion.

The first two principles simply emphasize the resulting precision and correlation of a group. The third principle states the varying intra-group weighting that the neighboring correlations decrease with distance, and so the distortion should be normalized by the distance before adding. The last principle requires the uniformity of each group so that inter-group comparisons are fair and under the same conditions. In order to measure the group size, a simple metric called mean internal distance (MID) is proposed as

$$\mathrm{MID} = \sum_{i=1}^{N} \frac{d_i}{N}. \tag{3}$$

For each group where the inner point is positioned in the center, the MID is defined as the mean of the distances d_i from the central point (inner point) to each neighbor (evaluated point), and N is the number of neighbors in the group.

Up until now, we have inspected the 6-side-based fast inner search and our statistical analyses. Two different philosophies for the fast inner search, group-oriented and point-oriented, are brought out. The difference between them is that the group-oriented approach does not make a dedicated group for each inner point, while the point-oriented approach specifically assigns one single group for each. Owing to this reason, some shortcomings of the 6-side-based method are observed. First, it has relatively large grouped regions, which contain up to three inner points and require more subsequent effort to converge. Besides, if the inner points are considered individually, we find that its internal structure is quite irregular and that there are four different values of MID, as shown in Fig. 6.

IV. Fast Inner Searches for HS and DS

A. *Point-Oriented Inner Search for EHS*

Based on the principles proposed in the previous section, a more regular and efficient grouping method can be defined for the hexagon-based search pattern. The new grouping of the evaluated points is based on minimizing the MID for each inner point, as shown in Fig. 7(a). Since the hexagon shape is not entirely symmetric, we unavoidably have two different values of MID inside the hexagon. For fair comparisons between groups, they are further classified into two sets by the MID—Set-1: $\{a, c, e, f, g, h\}$ and Set-2: $\{b, d\}$. Within the same set, the inner points have the same MID to the others so that the evaluation can be made under equivalent correlation. These points are either surrounded by three or two nearest evaluated points, for instance, point "a" is grouped with points 0, 2, 3 and point "b" is grouped with point 0, 4. Similar to the 6-side-based inner search, a group distortion is used as a reference to predict the minimum location, but the evaluated distortions are normalized before adding up. The normalized group distortion (NGD) is expressed as

$$\mathrm{NGD} = \sum_{i=1}^{N} \frac{\mathrm{SAD}_i}{d_i} = \sum_{i=1}^{N} \frac{\mathrm{SAD}_i}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}. \tag{4}$$

The NGD is the sum of all the distortions (SAD is used here) of the neighbors divided by the distance d. Accordingly, (x_i, y_i) and (x, y) are the coordinates of neighbor i and the inner point respectively, and N is the total number of grouped neighbors.

Within each set (Set-1 and Set-2), the NGDs of the different groups are compared. Each set will find one inner point with the smallest NGD. Finally the 2 inner points will be searched. This is depicted by the example in Fig. 7(b). The two selected points have a much higher chance to be the final minimum point, as compared with the other six points. This grouping has higher utilization of the neighboring correlations than the 6-side-based grouping and only requires two additional search points constantly. The computational overhead for these group

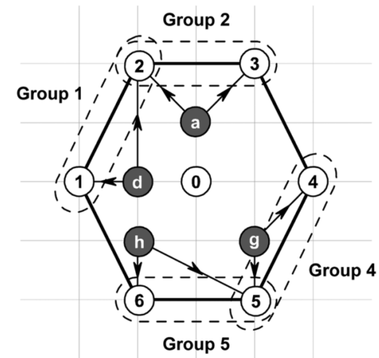

Fig. 6. Four values of mean internal distance are raised by 6-side-based fast inner search.

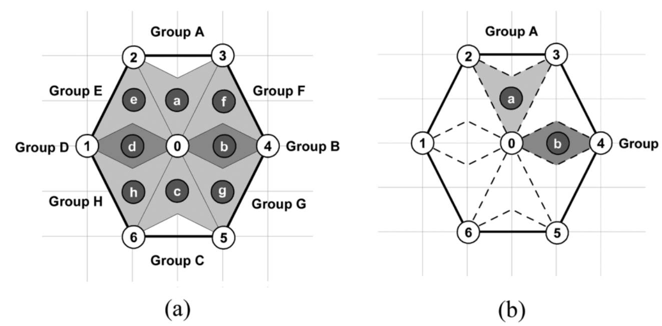

Fig. 7. (a) New grouping for the hexagonal search pattern based on minimizing MID. The eight inner points are separated into 2 sets by the MID value—Set-1: $\{a, c, e, f, g, h\}$ and Set-2: $\{b, d\}$. (b) Point "a" and "b" are searched finally, because group A and group B have the smallest NGD in the corresponding set.

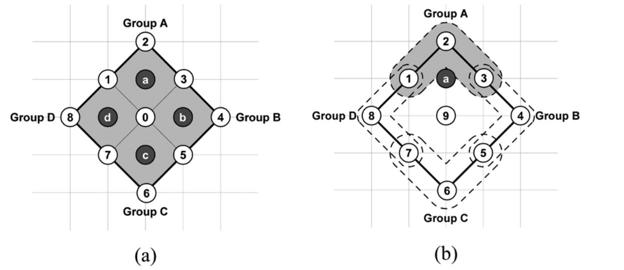

Fig. 8. (a) Configuration of large diamond search pattern (0–8) and its inner search points (a-d). The grouping is based on the four corners with exactly the same MIDs. (b) Point "a" is searched finally as group A has the smallest group distortion.

distortion computations and comparisons is similar to the 6-side-based inner search, thus it should be negligible. With the use of the proposed point-oriented fast inner search, the hexagon-based search can achieve faster convergence, and so it is named enhanced hexagonal-based search using point-oriented inner search (EHS-POIS). The algorithm is summarized in three steps, as follows.

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 seven checking points of the hexagon with the center at the previous minimum distortion point. If the new minimum distortion occurs at the center of the hexagon, go to Step 3; otherwise this step is repeated again.

Step 3 : Compute the NGDs of the hexagon and find out the minimum NGDs for Set-1 and Set-2. Based on the minimum NGDs, compute the distortions of the two additional search points and then identify the new minimum distortion point, which is the final motion vector.

B. Enhanced Diamond Search

Due to the regularity and symmetric shape of the LDSP, the proposed point-oriented strategy is very suitable for enhancing the original DS. It is because the search points of a LDSP could be easily divided into four corner groups with four points per group, as shown in Fig. 8(a). Each inner point is surrounded by four nearest evaluated points with exactly the same MID to other groups. Similarly, the NGD is calculated and used for group comparisons. However, as all the internal distance is equal to one and all four groups include the central point 0, the normalization does not take effect and we do not need to count point 0 for practical implementation. As a result, the inner point within the smallest distortion corner group should be searched as shown in Fig. 8(b). The selected inner point has a much higher chance to be the final minimum point as compared with the other three points. This is a very efficient inner search for DS as it only requires one search point in the final step. In addition, this inner search pattern is much more regular and simple in terms of the internal distance as compared with the 6-side-based inner search, which requires two or three additional search points with six group distortion comparisons. With the use of the proposed 4-corner-based fast inner search, the enhanced diamond search (EDS) is summarized as follows.

TABLE I
INNER MOTION VECTOR DISTRIBUTION FOR HS AND DS

Position	0	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*
Motion vector distribution inside hexagon search pattern									
Akiyo	96.27	0.97	0.23	1.57	0.27	0.22	0.17	0.17	0.11
Foreman	35.03	11.93	12.65	5.54	11.54	4.09	9.50	2.68	7.05
Mobile	37.12	9.47	7.37	7.40	34.77	2.21	0.62	0.21	0.82
Sean	87.40	3.81	2.02	3.16	1.60	0.51	0.39	0.61	0.50
Silent	87.73	2.51	2.76	2.08	2.55	0.78	0.38	0.82	0.40
Stefan	58.34	4.42	11.72	5.68	14.33	1.65	1.29	1.36	1.20
Average	66.98	5.52	6.13	4.24	10.84	1.58	2.06	0.98	1.68

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.

C. Early Termination

Besides the new inner search methods, an early termination method [14], [15] is also employed. It is reasonable to assume if the current sub-optimal distortion is small enough, it is not necessary to spend further effort to search another optimum. The motion vector distributions of the inner search area for HS are summarized in Table I. There are around 35%–96% motion vectors located in the position of point 0 among the eight inner search points inside the LHBSP. This locally center-biased characteristic is particularly obvious for the sequences containing large amount of static background e.g., Akiyo, Sean, and Silent. The proposed 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 SAD block distortion measure. The threshold is set to keep the picture quality unchanged, but further lower the minimum number of inner checking points to zero.

V. EXPERIMENTAL RESULTS

To demonstrate the performance of the proposed algorithms, simulations are done to evaluate the computational complexity and prediction accuracy. The simulations are performed on six representative CIF sequences ("Akiyo", "Container", "Foreman", "News", "Silent", and "Stefan"). The simulation settings are: 100 frames, 16×16 block size, ± 16 search window, and SAD block distortion measure. The results are tabulated by three testing criteria—average PSNR per frame, average number of search point per block, and speed improvement rate (SIR). The number of search points directly reflects the absolute speed of different algorithms, while the

TABLE II
SIMULATION RESULTS OF DIFFERENT ALGORITHMS ARE SUMMARIZED IN THREE CATEGORIES RELATIVE TO THE VALUES OF HS

	Akiyo	Container	Foreman	News	Silent	Stefan
Average PSNR per frame relative to HS						
HS	**42.556**	**38.331**	**32.340**	**37.038**	**35.714**	**23.717**
HS+	0.227	0.000	0.530	0.068	0.085	0.075
EHS	-0.242	-0.002	0.326	-0.041	0.033	0.006
EHS-POIS^	-0.245	-0.001	0.360	-0.047	0.041	0.005
EHS-POIS	-0.066	-0.001	0.378	-0.003	0.059	0.042
EHS-POIS+	-0.066	-0.001	0.376	-0.012	0.059	0.042
Average number of search points per block relative to HS						
HS	**10.346**	**10.419**	**12.659**	**10.588**	**10.958**	**13.758**
HS+	3.606	3.612	3.645	3.605	3.603	3.630
EHS	-0.973	-1.407	-1.313	-1.090	-1.208	-1.405
EHS-POIS^	-1.626	-1.633	-1.652	-1.625	-1.623	-1.649
EHS-POIS	-1.626	-1.633	-1.652	-1.625	-1.623	-1.649
EHS-POIS+	-3.553	-3.349	-1.915	-3.443	-3.216	-2.155
Speed improvement Rate (%) over HS						
HS+	-34.85	-34.67	-28.79	-34.05	-32.88	-26.39
EHS	9.41	13.50	10.37	10.29	11.02	10.21
EHS-POIS^	15.72	15.67	13.05	15.35	14.81	11.98
EHS-POIS	15.72	15.67	13.05	15.35	14.81	11.98
EHS-POIS+	34.34	32.14	15.13	32.51	29.35	15.67

TABLE III
SIMULATION RESULTS OF DIFFERENT ALGORITHMS ARE SUMMARIZED IN THREE CATEGORIES RELATIVE TO THE VALUES OF DS

	Akiyo	Container	Foreman	News	Silent	Stefan
Average PSNR per frame relative to DS						
DS	**42.817**	**38.331**	**33.088**	**37.189**	**35.783**	**23.837**
EDS	-0.026	-0.002	-0.021	-0.013	-0.004	-0.002
EDS+	-0.027	-0.002	-0.022	-0.022	-0.004	-0.002
Average number of search points per block relative to DS						
DS	**12.284**	**12.375**	**16.310**	**12.670**	**13.350**	**17.794**
EDS	-2.424	-2.436	-2.482	-2.425	-2.420	-2.477
EDS+	-3.671	-3.554	-2.667	-3.608	-3.478	-2.825
Speed improvement Rate (%) over DS						
EDS	19.73	19.69	15.22	19.14	18.13	13.92
EDS+	29.88	28.72	16.35	28.48	26.05	15.88

SIR reflects the speedup percentage relative to the original search. The SIR of method 1 over method 2 is defined by $\mathrm{SIR} = (N_2 - N_1)/N_2 \times 100\%$, where N_1 is the number of search point used by method 1 and N_2 is that of method 2.

A. Evaluation of EHS-POIS

In this subsection, we will emphasize the improvement of the proposed fast inner search algorithm over HS. The measured value of HS is listed as a reference, and the other algorithms will be provided with relative values. That means a positive value indicates an increase over HS and vice versa. The compared five algorithms are HS+ (8-point full inner search), EHS (6-side-based inner search), EHS-POIS (point-oriented inner search), EHS-POIS^ (no NGD normalization), and EHS-POIS+ (plus early termination). From Table II, the simulation result reveals that HS+ has a PSNR increase over HS from 0.07 to 0.53 dB consistently due to its exhaustive inner full search. Moreover, the PSNR change of EHS is around -0.42 to 0.33 dB, while that of EHS-POIS is around -0.15 to 0.38 dB. We can see that EHS-POIS always yields higher picture quality than the original EHS for all the tested sequences. For the other two variations, EHS-POIS^ and EHS-POIS+, the changes are around -0.25 to 0.36 dB and -0.15 to 0.38 dB, respectively. The results prove two things. First, the NGD is always a better prediction than the non-normalized one. Second, the early termination does not ruin the picture quality at all, as the maximum difference between EHS-POIS and EHS- POIS+ is only 0.009 dB (News). The above data reveals that a negligible degradation may be caused by the fast inner search algorithms. However, a slightly increase of PSNR may sometimes be shown as the original HS only checks a small cross (four points) around the center. In short, the proposed 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 compare the speed performance, HS+ always needs about 3.6 search points more. This is actually a tradeoff for the gain in the PSNR. On the other hand, the EHS can save around 1.0 to 1.4 points. For EHS-POIS and EHS-POIS^, they constantly save the same number of search points that are around 1.6, while the improved EHS-POIS+ can save around 1.7 to 3.6 search points. It is found that all the simulation results are a bit less than the theoretical value, since the implementation enables additional inner checking points if the corresponding groups are out of the frame boundary.

Finally, the speed improvement rates over the HS are shown. It is expected that HS+ uses about one third more computations. The SIR of EHS is around 9% to 14%, while that of EHS-POIS and EHS$-$ POIS+ are around 12% to 16% and 14% to 34%, respectively, which are significantly higher than that of the 6-side-based methods. The experimental results justify the ideas that the new grouping method is more suitable and efficient for the hexagonal search pattern. The superior performance on both accuracy and speed is also demonstrated for the new fast inner search.

B. Evaluation of EDS

Similarly, we use the same method to interpret the simulation results for DS and EDS. Since EDS does not require normalization, only the variation, EDS+ (plus early termination), is listed together. All the results are shown as a relative value to the DS. To compare the picture quality first, Table III shows that the PSNR changes of EDS and EDS+ against DS are from -0.027 dB to 0.002 dB, which are surely negligible. The difference between EDS and EDS+ is smaller than 0.009 dB (News). Furthermore, EDS saves around 2.5 search points on average, while EDS+ saves around 2.5 to 3.7 search points. Due to the same reason mentioned above, it is also less than the theoretical value, and the early termination can efficiently reduce one more search point for most low to medium motion sequences. Again, the speed improvement rates of the two algorithms over DS are tabulated in the bottom of the table. The SIR of EDS and EDS+ over DS are around 14% to 20% and 16% to 30%, respectively. The same set of principles for developing fast inner search algorithms is applied to both HS and DS. To make a comparison between them, we only take the results of EHS-POIS and EDS into account, as the early termination is related to the motion vector distribution rather than the pattern structure. For both the gains of speedup and prediction, EDS outperforms EHS-POIS moderately. This is evidence to support the fact that the regular structure of the DS pattern is most suitable and efficient for the proposed kind of fast inner search.

TABLE IV
PERFORMANCE COMPARISONS BETWEEN THE PROPOSED FAST INNER SEARCH ALGORITHMS AND THE H.264 EMBEDDED FAST ALGORITHMS

	Akiyo	Container	Foreman	News	Silent	Stefan
Average PSNR per frame relative to FFS						
FFS	**38.26**	**36.11**	**35.70**	**36.71**	**35.87**	**34.35**
EDS+	0.02	0.01	-0.02	0.00	0.00	-0.03
EHS-POIS+	0.00	0.01	-0.04	-0.01	-0.02	-0.05
UMHexagonS	0.02	0.00	-0.02	-0.01	0.01	-0.01
Percentage change of bitrate over FFS (%)						
EDS+	-0.54	-0.53	1.35	0.09	0.54	1.05
EHS-POIS+	0.20	0.06	1.47	0.66	1.18	1.59
UMHexagonS	0.10	-0.80	-0.52	-0.48	0.16	-0.01
Speedup ratio over FFS						
EDS+	8.71	9.09	7.42	8.57	7.91	6.97
EHS-POIS+	10.00	9.97	8.32	9.64	9.10	8.07
UMHexagonS	9.65	7.72	3.58	6.97	5.39	2.88

Lastly, we compare the proposed algorithms with the H.264 embedded fast full search (FFS) and the unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [17] algorithm using the H.264/AVC reference software JM9.6 [16] with quantization parameter $(\mathrm{QP}) = 28$ for the simplest baseline profile. Basically, the FFS speeds up the coding process by reusing the pre-calculated block distortion. The UMHexagonS is a hybrid fast motion-estimation algorithm, which combines many advance techniques such as motion vector prediction, early termination, and search patterns of cross, hexagon, and diamond. Table IV shows the comparison of EDS+, EHS-POIS+ and UMHexagonS against FFS in terms of PSNR, bitrate, and speedup ratio. The speedup ratios are calculated by counting the number of CPU cycles involved in motion estimation. From the results of Table IV, we find that the motion estimation quality of these three fast algorithms is very close to FFS. The change of PSNR ranges between -0.05 and 0.02 dB, and the bitrate varies from -0.80% to 1.59%. While the quality is kept significantly similar to FFS, our proposed algorithms can achieve seven to ten times the speedup over FFS. The speedup ratio is particularly higher than UMHexagonS for more complex and vigorous sequences like “Foreman” and “Stefan”.

VI. CONCLUSIONS

In this paper, new grouping principles are proposed based on an analyzed statistic of the inner area. It is found that the mean internal distance is a very good measurement for the correlation between the inner points and coarse points. Then, two fast motion-estimation algorithms, EHS-POIS and EDS, are proposed to improve the efficiency of the original HS and DS algorithms using new point-oriented fast inner search techniques. As a result, the two new fast algorithms significantly reduce the complexity of their original search methods by up to 34% and 30%, respectively. At the same time, a negligible PSNR loss is kept below 0.15 dB. As compared with other previous works, the proposed EHS-POIS is also proven to out-perform the EHS on both accuracy and speed.

REFERENCES

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing ,” in *Proc. Nat. Telecommunications Conf.*, New Orleans, L.A, Nov.-Dec. 1981, pp. G5.3.1–G.5.3.5.

[2] 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.

[3] R. Srinivasan and K. R. Rao, “Predictive coding based on efficient motion estimation,” *IEEE Trans. Commun.*, vol. COM-33, no. 8, pp. 888–896, Aug. 1985.

[4] 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, no. 4, pp. 438–443, Aug. 1994.

[5] 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, no. 3, pp. 313–317, Jun. 1996.

[6] L. K. Liu and E. Feig, “A block-based gradient descent search algorithm for block motion estimation in video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419–422, Aug. 1996.

[7] 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.

[8] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[9] C. H. Cheung and L. M. Po, “A novel cross-diamond search algorithm for fast block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec. 2002.

[10] 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, no. 5, pp. 349–355, May 2002.

[11] C. H. Cheung and L. M. Po, “Novel cross-diamond-hexagonal search algorithms for fast block motion estimation,” *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 16–22, Mar. 2005.

[12] X. Jing and L. P. Chau, “An efficient three-step search algorithm for block motion estimation,” *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 435–438, Jun. 2004.

[13] C. Zhu, X. Lin, L. P. Chau, and L. M. Po, “Enhanced hexagonal search for fast block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 10, pp. 1210–1214, Oct. 2004.

[14] Y. Nie and K. K. Ma, “Adaptive rood pattern search for fast block-matching motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1442–1449, Dec. 2002.

[15] A. M. Tourapis, O. C. Au, and M. L. Liou, “Highly efficient predictive zonal algorithms for fast block-matching motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 934–947, Oct. 2002.

[16] JVT H.264/AVC Reference Software Joint Model 9.6 (JM 9.6) [Online]. Available: http://iphome.hhi.de/suehring/tml/

[17] Z. Chen, Y. He, and J. Xu, “Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H.264,” in *Proc. Picture Coding Symp.*, Saint Malo, Apr. 2003, pp. 17–22.

Lai-Man Po, photograph and biography not available at the time of publication.

Chi-Wang Tin, photograph and biography not available at the time of publication.

Ka-Man Won, photograph and biography not available at the time of publication.

Ka-Ho Ng, photograph and biography not available at the time of publication.