Journal of Information Hiding and Multimedia Signal Processing ©2010 ISSN 2073-4212
Ubiquitous International Volume **1**, Number **2**, April **2010**

Multipath Flatted-Hexagon Search for Block Motion Estimation

Chao-Ho Chen
Department of Electronic Engineering
National Kaohsiung University of Applied Sciences
415, Chien Kung Rd., Kaohsiung 807, Taiwan (R.O.C.)
thouho@cc.kuas.edu.tw

Tsong-Yi Chen
Department of Electronic Engineering
National Kaohsiung University of Applied Sciences
415, StreetChien Kung Rd., Kaohsiung 807, Taiwan (R.O.C.)
chentso@cc.kuas.edu.tw

Da-Jinn Wang
Department of Information Management
National Kaohsiung Marine University
No.142, Haijhuan Rd., Nanzih District, Kaohsiung City 81143, Taiwan (R.O.C.)
wangdaj@mail.nkmu.edu.tw

Yi-Fan Li
Department of Electronic Engineering
National Kaohsiung University of Applied Sciences
415, StreetChien Kung Rd., Kaohsiung 807, Taiwan (R.O.C.)

Received August 2009; revised March 2010

Abstract. *This paper proposes a novel and simple multipath search with atted-hexagon or diamond pattern for block motion estimation to achieve adjustable speed/accuracy in block-matching algorithm (BMA). To improve the accuracy of the fast BMA near to that of full search (FS), the inherent problem of being trapped at the local minimum block distortion measure (BDM) should be overcome substantially. In the proposed method, a threshold of BDM is introduced to determine the possible-optimal search directions in order to escape from being trapped into a local minimum BDM, followed by a atted-hexagon or diamond search performed in these directions with a BDM below a threshold. Then, the estimated motion vector will be re*

ned at each search step until the searching process is stopped. The BDM threshold will be adjustable for the purpose of adjusting the search speed and search accuracy speci

ed in the certain applications. Experimental results show that the proposed multipath search algorithm can achieve an average match- ing probability up to 98% near to that of FS and about 10 times of checking points faster than FS in most of real-world sequences.
Keywords: Motion Estimation, Block-matching Algorithm, Multipath Search, Flatted-hexagon Search

1. Introduction.

Motion estimation can make the interframe coding to achieve a very high compression ratio, when compared to the intraframe coding, by exploiting the heavy temporary redundancy between successive frames. Among various motion estimation techniques, the block-matching algorithm (BMA) is the most attractive method for the current international video compression standards including H.261, H.263, MPEG-1, MPEG-2 and MPEG-4 [1]-[6], because of its effectiveness and simplicity for implementations [7]. However, the matching process of finding the optimal still involves a large amount of calculations, e.g. the full search (FS) method (i.e., the most accurate approach), in which all candidate blocks require to be evaluated. To reduce the intensive computational complexity with a tolerable distortion, many fast block-matching algorithms were developed [8]-[21].

Among the above suboptimal methods, both the search patterns attributes and initial searching range always directs the developmental processes of these algorithms. By taking advantage of the characteristics of the center-biased motion vector distribution existed in most real-world image sequences, the new three-step search (NTSS) [9], four-step search (4SS) [10] and block-based gradient descent search (BBGDS) [11] perform better than the three-step search (3SS) [8], where these four search patterns are square-shaped. Based on a practical compact-shaped pattern with fewer candidate search points per block, a diamond-search (DS) algorithm [12][13] can not only improve the searching speed but also reduce the chances of being trapped in local minimum block distortion measure (BDM) points, when compared to those four algorithms. To improve the local-minimum trapping problem in the 3SS algorithm for the estimation of small motions, an efficient three-step search (E3SS) algorithm [14] employs a small diamond pattern in the first step and the unrestricted search step is used to search the center area. It performs better than DS in terms of MSE with fewer or comparable number of search points for the sequences that contain medium to large motion, but is inferior in speed performance to DS when searching small motion vectors. The hexagon-based search (HEXBS) algorithm [15] utilized a hexagon-shaped pattern with only 7 checking points in the initial search and 3 checking points in the following searches to achieve substantial speed improvement over the DS algorithm with similar distortion performance for most high-resolution (e.g. 720×480) image sequences. Nevertheless, the matching-probability (i.e., the probability of finding the true motion vector) will degenerate with the decreasing resolution of the video format. By introducing a fast inner search into the interior of hexagonal pattern, an enhanced hexagonal search algorithm [16] is proposed to improve HEXBS in search accuracy. The introduction of flatted-hexagon search (FHS) pattern will make the FHS algorithm [17] to provide a better speed-probability product than the above fast BMAs, when both speed performance and matching probability need to be considered. The basic idea behind the FHS algorithm is that the covering range of a search pattern should be enlarged as horizontal as possible to find the optimal motion vector quickly because the occurrence probability of horizontal-biased motions is larger than that of vertical-biased motions in most of real-world image sequences. To obtain a faster searching speed than the DS algorithm while maintaining similar search quality, the cross-diamond search (CDS) algorithm [18] and cross-diamond-hexagonal search (CDHS) algorithm [19] employed a cross-shaped pattern at the initial step to exploit the characteristics of the center-biased motion vector distribution very efficient, followed by the halfway-stop technique, and the large/small diamond or hexagon search patterns in the subsequent steps. Based on inter-block correlations, an adaptive rood pattern search (ARPS) [20] dynamically determines the size of the search pattern in the initial search stage in order to find a good starting point for each macroblock. In addition, zero-motion prejudgment is incorporated to further speed up the search, particularly beneficial to the sequences containing small motions. By vector quantization technologies [22][23], a new approach of using predictive fine granularity successive elimination for fast optimal block matching motion estimation is proposed in [21]. Nevertheless, various search patterns and/or extra processes at searching steps will make those algorithms [18]-[21] to be complicated in realization, especially for VLSI implementation, because of considering the regularity [24]. Without loss of generality, a regular searching algorithm using a single search-pattern is always more interesting in realization cost than the search algorithm using the complicated search process or multiple various search-patterns.

Generally speaking, the common drawback of fast BMAs is that they cant almost approach FS in search accuracy for the real-world sequences, so these algorithms are also called suboptimal BMAs. In fact, the local-minimum trapping problem will be major factor and may occur when there are multiple local minima existed in the search window, especially for large motion blocks, in the real-world sequences. However, most of the previous fast BMAs are based on the assumption that the BDM increases monotonically as the search pattern moves away from the global minimum BDM point. Owing to using such a monotonous searching path, those BMAs frequently suffer the local-minimum trapping problem and hence cant have a matching probability near to FS. In theory, to cope with the local-minimum trapping problem, the multipath search approach will be the better method, especially for the case of multiple minimum BDM points existed. Besides, those algorithms cant also provide an adjustable search speed and picture quality for some specific applications. To reduce the local-minimum trapping problem, this paper develops a multipath search algorithm that uses the dynamic BDM threshold to derive possible directions of leading to the global minimum BDM point. The proposed search method is dedicated to achieving high search accuracy near to FS, i.e., people cant visually discriminate the difference of motion-compensated results using motion vectors estimated by the both search algorithms. In the proposed multipath search scheme, many well-known search patterns can be employed. In this paper, we propose two multipath search algorithms: multipath flatted-hexagon search (MFHS) and multipath diamond search (MDS), for low-resolution (e.g. CIF, 352x288 or SIF, 352x240) and high-resolution (e.g. CCIR601, 720x480)) image sequences, respectively. The searching scheme employs the flatted hexagonal pattern or diamond pattern to search for each possible optimal path and is further designed for adjusting the search speed and matching probability. It also points out that the flatted-hexagon pattern is more effective than others in the image sequences mainly containing horizontal-biased motions for low-resolution image sequences. For brevity, only the MFHS algorithm is discussed since MDS has the same search process as MFHS except that the search pattern used is different from that of MFHS. The following section describes the proposed multipath search algorithm including the analysis of search strategy, selection of search pattern, and the search process of MFHS. Section III discusses the simulation results of FHS, MFHS and MDS and comparisons with several reported fast BMAs, and conclusions are made in the final section.

2. **Multipath Search Algorithm.**

2.1. **Analysis of Search Strategy.**

Most conventional block motion estimation algorithms are explicitly or implicitly based on the assumption: BDM increases monotonically as the checking point moves away from the global minimum. Obviously, this assumption essentially requires that the error surface is unimodal over the search windows. Unfortunately, this is usually not true due to many reasons such as the aperture problem, the textured (periodical) local image content, the inconsistent block segmentation of moving object and background, the luminance change between frames, noises, and etc. As a consequence, this may make the search easy to be trapped into a local minimum.

Recently, some pre-existing fast BMAs [14][18]-[21] employed a cross-shape search pattern in the first step to possibly avoid being trapped at a local minimum BDM point. In those methods, even all fast BMAs, the search direction for the next step is oriented by someone search point that has the least BDM value among points checked in the current step. However, it may be not always true for the assumption that the search direction oriented by the minimal BDM point at each step will be toward the final position of the global minimum error. This may be explained by the fact that there is one local minimum BDM point existed in the neighborhood of the point with the least BDM value and this will lead the search to be trapped into that local minimum. Basically, the direction of the global minimum is always oriented by the search points of low BDM value. Hence, it implies that such low-BDM points should be considered to settle the searching direction for the next step in order to escape from being trapped into a local minimum. In other words, those low-BDM points will be the candidates of searching in the direction of the global minimum. As an example, Figure1 describes a case of misleading the search direction for the next step in the DS algorithm and thus it will be likely to be trapping into one of local minima around the point of BDM value 67, where the grayish point of BDM value 4 indicates the global minimum. In the figure, there are two smaller BDM points of values 67 and 73 among checking points searched in the current step. Obviously, to avoid misleading the search direction for the next step, these two points of BDM-value 67 and 73 seem to be required for being oriented as the search directions in the following search path to find the global minimum point. Based on the above discussions, it reveals that the multipath search in the direction of certain low-BDM points for the following step will be more effective to cope with the local-minimum trapping problem than the single-path search used in those famous fast BMAs.

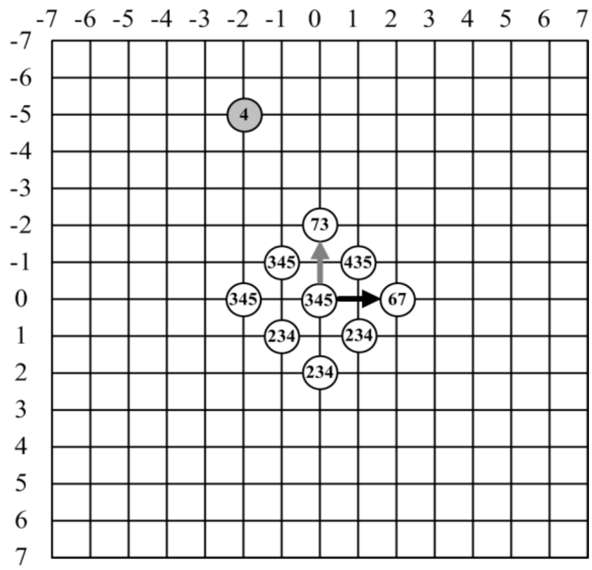
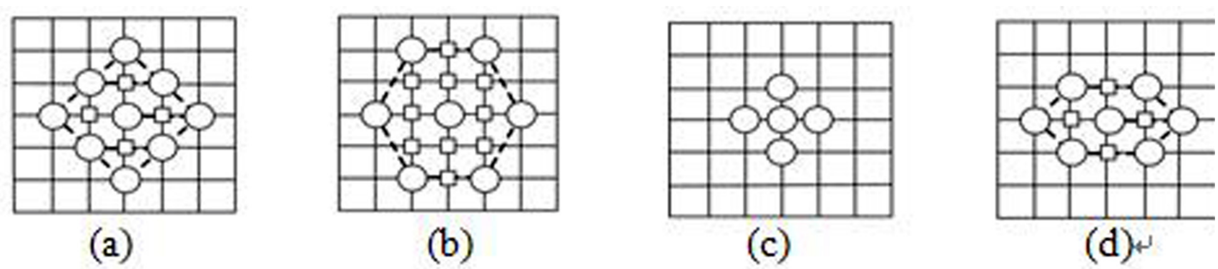


Figure 1. An example of false searching direction for DS.

2.2. **Selection of Search Pattern.** An in-depth examination of the previous researches [8]-[20] reveals that both shape and size of a search pattern can influence the searching speed and quality significantly. In respect to shape, the main merit of compactness is to consider all possible searching directions for tracking the optimal motion vector that has the least matching error. The hexagon is more compact than the diamond, as shown in Figure 2(a) and (b), which in turn is better than the square. The patterns size will affect the probability of the best match and also the moving speed of the search pattern. The moving speed of a search pattern within the searching window is directly proportional to the size of that pattern. The faster moving of a large search pattern will increase the speed of finding the large motion vector. As a result, a large search pattern is more suitable for the video with large motion contents than a small pattern. Small-size pattern usually causes the searching to be trapped into a local minimal-error point, especially for those image sequences with large motion contents which may also implies the high-resolution format. On the other hand, a large search pattern is most likely to result in misleading of the searching direction that may frequently either delay the searching time or even miss the optimal one, especially for the video with small-motion contents or low-resolution format. Besides, the quantity of checking points required at each step will also have the similar influence on the search speed and quality as the patterns shape and size. Basically speaking, fewer checking points needed in every step can speed up the search but suffer a larger distortion, and oppositely more checking points will reduce the search speed but can provide a better quality performance of block-matching. The matching rate of a search pattern at each step is mainly dependent on the quantity of the checked points within all candidate search points covered by that pattern. In other words, the matching probability will be inversely proportional to the hollowness degree, which is defined as the ratio c/n where c is the number of unchecked points and n is the number of total points within the search pattern, i.e., all candidate search points.



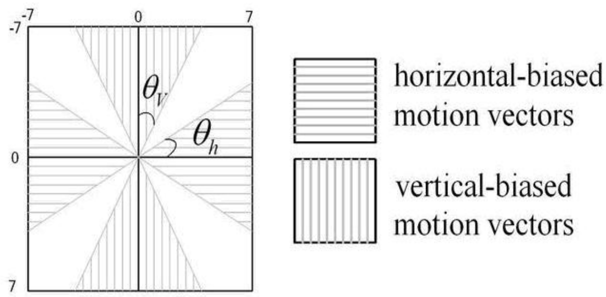
○: checking point; □: unchecked point

FIGURE 2. Shapes of various search patterns: (a) diamond pattern; (b) hexagon pattern; (c) 5-point cross pattern; (d) flatted-hexagon pattern.

Among fast BMAs with the center-biased search, the diamond-shaped search algorithm [12, 13] performs better than those square-shaped methods [8-11], because its medium-size compact-shape pattern with 9 checking points can find any-size motion vectors under a certain search speed and quality. Besides, the unrestricted search process minimizes the distortion caused by the local-optimal trapping problem. The hexagon pattern adopted in [15] has a more compact (i.e., circle-approximated) shape with larger size and less checking points (7 checking points) than the diamond pattern. By using such a search pattern, the HEXBS algorithm has a speed-up improvement over the DS algorithm mainly owing to the contribution of less checking points. But, the combination of fewer checking points and larger size for the search pattern will make HEXBS to suffer degradation on the probability of finding true motion vectors. Thus, the HEXBS algorithm can maintain a matching probability similar to that of the DS algorithm for most high-resolution image sequences (e.g. CCIR601) or some certain low-resolution image sequences (e.g. SIF or CIF) with a highly center-biased motion vector distribution. Basically speaking, HEXBS has a significant degradation of matching-probability for most low-resolution image sequences in comparison with DS. The best explanation of low matching-probability for HEXBS is its high degree of hollowness, 10/17, existed in the search pattern, while the diamond pattern used in DS has a lower hollowness degree of 4/13. Both DS and HEXBS adopt a cross-pattern of five points, as shown in Figure 2(c), to recheck whether the zero motion vector obtained by the initial search is the final solution. An initial search by using 5-point cross pattern will provide a very high search speed and matching probability for block motion estimation in an image sequence containing massive quasi-zero motion vectors. For an image sequence containing massive zero motion vectors, the HEXBS algorithm will have a similar matching-probability but a faster search speed compared to the DS algorithm because HEXBS can save 2 checking points than DS on the initial search.

Based on an advanced analysis on the distributions of motion vectors in the most real-world image sequences, it is clear that the occurrence possibility of horizontal-biased motions is significantly greater than that of vertical-biased motions. Table 1 lists the probability distributions of horizontal-biased and vertical-biased motions in seven well-known image sequences with various motion contents for a search window ± 7. In the table, the horizontal-biased and vertical-biased motion vectors are defined as a vector in which the angle between the motion vector and the horizontal and vertical axis, respectively, is equal or small than 30X. For the video-conferencing sequence, the Salesman sequence bears a very high center-biased motion vector distribution, i.e., containing a large quantity of small motions, with a low H/V (horizontal/vertical) probability ratio of 17.94/14.17. Belonging to the medium-motion sequence, Foreman has a medium H/V radio of 30.17/22.62, but the Coastguard sequence has a very high H/V radio of 81.85/3.53. Involving complicated large-motion contents, the Football sequence and Tennis sequence have medium H/V ratios of 27.14/17.05 and 26.07/20.50, respectively, but the Garden sequence captured by panning the camera with translation has an extremely high H/V ratio of 92.78/2.12.

The above analysis reveals that the covering range of a search pattern may need to be flatted horizontally in order to find the optimal motion vector quickly. This implies that both speed and probability performances in block-matching process will be improved for the most of real-world sequences if the shape of a search pattern is flatted horizontally. A hexagon will be the most attractive search-pattern to be flatted, since it has a more compact form than other search patterns reported previously. Based on both horizontallyVflatted and hexagonal characteristics, a flatted-hexagon search pattern used in the block-matching algorithm for motion-vector estimation is proposed [17]. The flatted-hexagon pattern can be viewed as that a hexagon pattern is flatted horizontally or that both top and bottom checking points of a diamond pattern are removed. Besides, similar to the hexagon search pattern, the flatted-hexagon search pattern is composed of seven checking points with the center surrounded by six endpoints of the flatted hexagon, as described in Figure 2(d). Because of its small shape, the flatted hexagon pattern has a lower hollowness degree of 4/11 than that of the hexagon pattern used in the HEXBS algorithm. Both features of low hollowness and horizontal-biased shape will significantly improve the matching probability over HEXBS, and a quantity of only seven checking points required in the pattern will provide a faster search speed than DS.



On the other hand, pre-determining an initial search point through evaluating certain highly reliable predictor sets can improve the searching efficiency substantially. The

TABLE 1. Probability Distributions of Horizontal- And Vertical-Biased Motions.

Sequence	Format	Horizontal Probability(%)	Vertical Probability(%)
Salesman	CIF,80 frames	17.94	14.17
Foreman	CIF,300 frames	30.17	22.62
Coastguard	CIF,300 frames	81.85	3.53
Garden	CIF,115 frames	92.78	2.12
Tennis	CIF,67 frames	26.07	20.50
Football	CIF,125 frames	27.14	17.05

*Horizontal-biased and vertical-biased motion vectors are defined as the vectors with $\theta_v \leq 30^\circ \, and \, \theta_h \leq 30^\circ$, respectively.

search using various search pattern in different search steps [16][18][19] will also improve the search speed and matching probability. However, in point of fact, those methods cant completely solve the local-optimal trapping problem, especially for the situation of multiple local-optimal points existed. Besides, using a single search pattern with a simple monotonous searching strategy is more suitable for being employed into the multipath search than the complicated or multi-pattern search because they will largely complicate the realization when they are used in the multipath searching process.

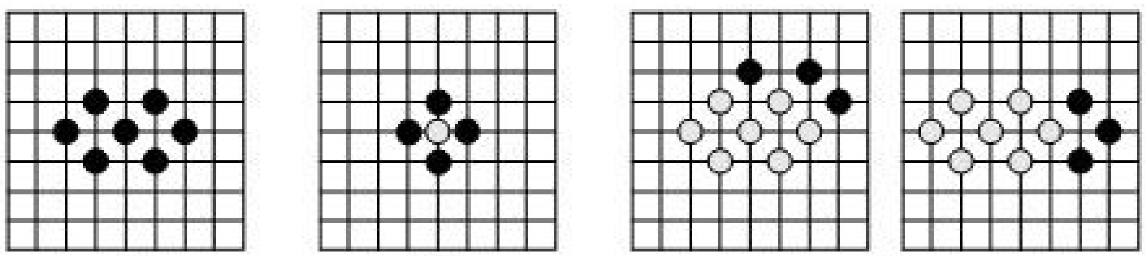
Theoretically, a multi-path search can increase the matching probability by escaping being trapped into a local minimal point, but the speed performance will be substantially reduced owing to searching multiple paths, compared to the single-path search. So, the search pattern used in the multi-path search should be designed with less checking points at each step to avoid reduction in speed performance. The flatted hexagonal pattern has only seven checking points and performs better than the above well-known search patterns when both search speed and matching probability need to be considered concurrently. Therefore, the flatted hexagonal search pattern will be the most suitable for the proposed multipath search approach.

2.3. **MFHS Algorithm.** The basic search strategy of the flatted hexagon search pattern is to keep advancing with the center moving to any of the six endpoints and whichever endpoint the center of the search pattern moves to. Thus, there are always three new endpoints introduced and the other three endpoints and the original center point are overlapped, as shown in Figure 3. A 5-point cross pattern, as plotted in Fig. 3 (b) that is also used in those famous BMAs, is finally used in the focused inner search. Firstly, a minimum block distortion measure is obtained by calculating the 7 search-points of the flatted-hexagon pattern which is located at the center of the search window, as shown in Figure 3 (a). If the minimum BDM is found at the central checking point, the search will switch to use the 5-point cross pattern which introduces new 4 search points around that center for ending the search, as depicted in Figure 3 (b). Then, one with minimum BDM among these 5 checking points will be selected as the optimal solution for motion vector estimation. Otherwise, the flatted-hexagon pattern moves toward one endpoint with a minimum BDM and then the search continues with the same flatted-hexagon pattern centered at that minimum BDM point in two normal forms of Figure. 3 (c).

In essence, each search path of MFHS adopts the search process of FHS, exclusive of determining the next search-paths. To judge which paths are required for approaching the optimal solution, a dynamic threshold is introduced to determine the local minimum points that may be in the direction of the global minimum. The block matching error is based on the measurement of SAD (sum of absolute differences) and SAD_{min} denotes the minimum SAD value among all checking points in a search step. For every search step, if the absolute difference between SAD value of someone point and SAD_{min} is smaller than or equal to a threshold T, such a point will be regarded as a local minimum. That is,

$$if\ |SAD_p - SAD_{min}| \leqq T,\ p\ is\ a\ local\ minimum \tag{1}$$

Then, the center of a new FHS needs to be moved to the local minimum point. The local-minimum decision rule (1) is used to find the next local minimum point from those checking points of multiple new FHSs searching patterns. Such a process will be continued until there is no new FHS excited. Thus, the motion vector is estimated at the location of the latest SAD_{min}. On the other hand, if the local minimum is located at the center of FHS, a 5-point cross search is executed for ending this search-path. But, in the first step, a 5-point cross pattern will end the MFHS process if there is only one local-minimum point located at (0, 0). The above discussion of the proposed MFHS algorithm is summarized in Figure 4. In Figure 5, we illustrate a search process with T = 25 for finding a motion vector (4, -1) by using 24 checking points. In the step 1, three local-minimum points with SAD = 50, 70, and 75 are firstly derived by the local-minimum decision rule (1) and then the SAD_{min} is set to 50. Thus, the following step performs three search paths based on those three local-minimum points, in which there are two new searches of FHS centered at points of SAD = 50 and 70 and one new 5-point cross search centered at the point of SAD = 75. Then, the value of SAD_{min} is updated by 20. In the step 3, only one local-minimum point of SAD = 40 is found, so the search process is reduced to one search-path and SAD_{min} remains as 20. Because the local-minimum of SAD = 40 is located at the center of an FHS, a 5-point cross search is executed for ending this search-path, as described in the step 4. Then, the value of SAD_{min} is updated by 10 and such a point of SAD_{min} is used to estimate a motion vector as (4, -1) with 24 (= 7+10+3+4) checking points.



(a) (b) (c)

FIGURE 3. Various forms of the flatted-hexagon search pattern: (a) starting search-points; (b) ending search-points; (c) search points of two normal searches. (● : the required checking point.)

In the proposed algorithm, a fixed T may result in different searching results for image sequences of various contents. Therefore, an adjustable scheme is introduced to provide an appropriate T for various image sequences, as shown in the following equation (2).

$$T = SAD_{min} \times \beta \tag{2}$$

In the equation, β is a parameter of local-minimum decision and its value ranges from 0 to 1 for most of the real world sequences, and the required number of checking points is

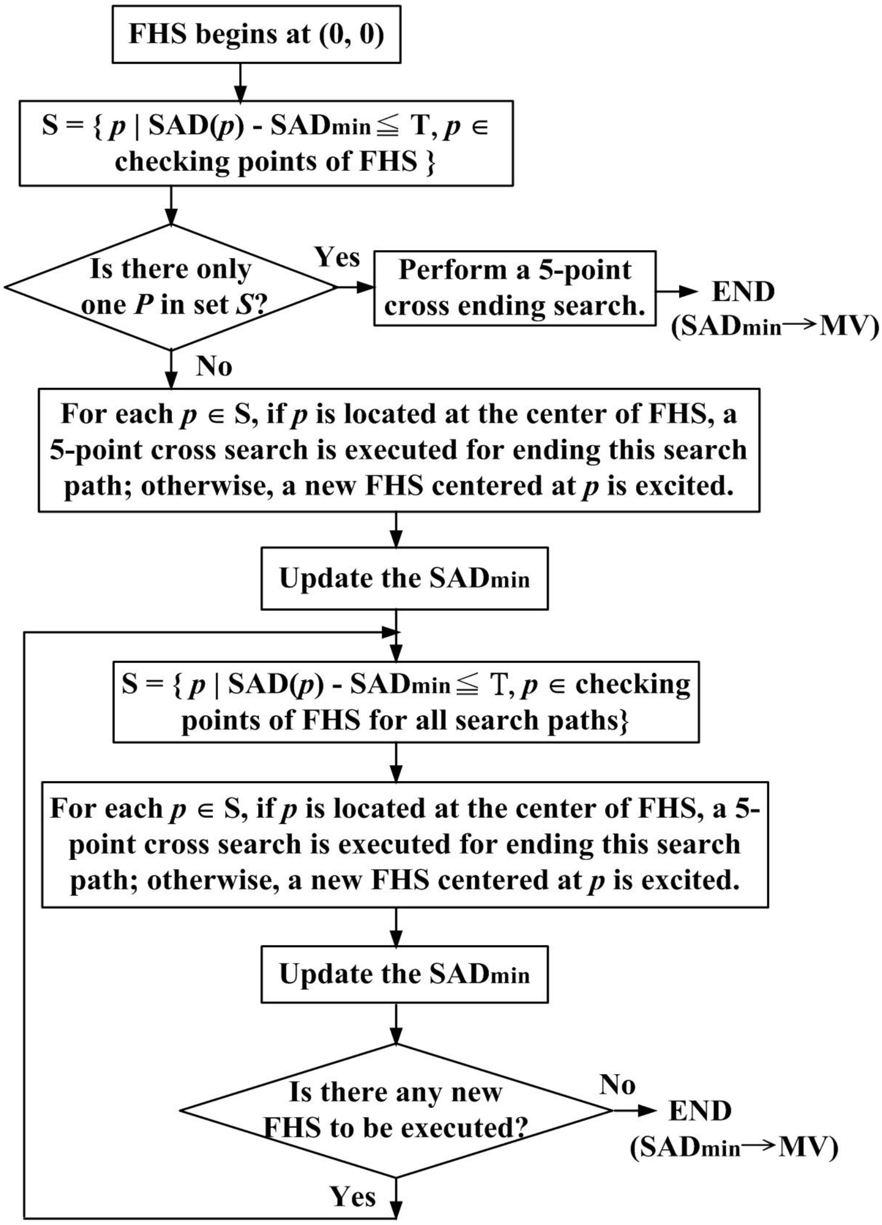


FIGURE 4. The MFHS algorithm

likely linearly proportional to β. If $\beta = 0$, T will reduce to zero and this means that only one local-minimum point is regarded as being in the direction of the global minimum at each step, i.e., MFHS will return to FHS. If $\beta = 1$, the matching probability will be near to 100%, i.e., the quality performance of MFHS will approach that of FS. Generally, we cant visually distinguish the motion-compensated result using motion vectors estimated by MFHS from that of FS when β is above 0.5, so the value of β is usually set below 0.5. It is pointed out that the noise interference and certain motion contents will affect the matching probability because these factors will result in more local minimum points to be verified. For the sequence containing complicated motions or zooming-captured motions, it needs a larger β to provide a better performance in search accuracy. Thus, by introduction of β, the value of T will be adjustable in order to make both search speed and matching probability to be adjustable.

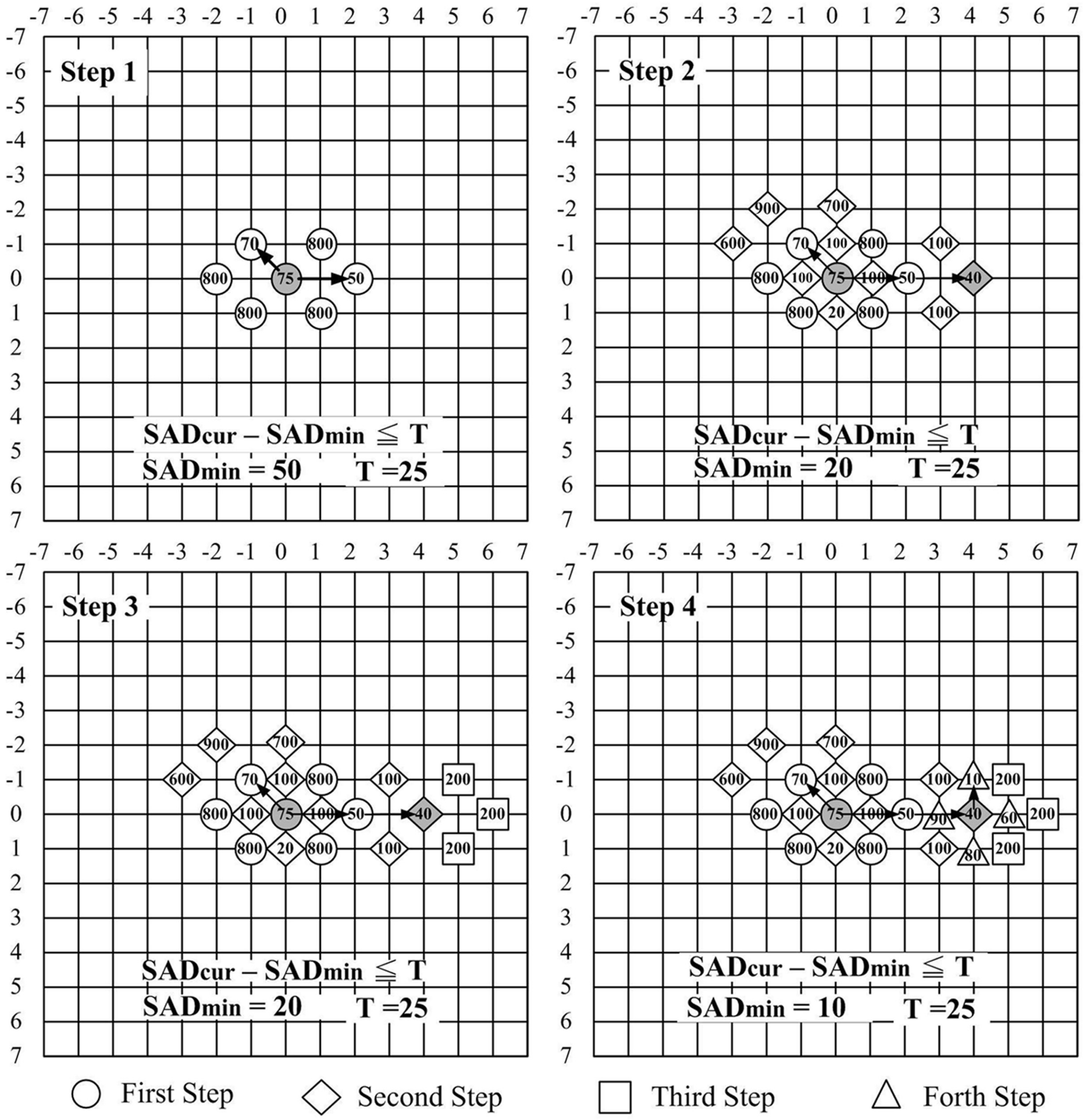


Figure 5. An example of MFHS,estimating a motion vector (4, -1) by using 24 checking points.

3. **Experimental Results and Discussions.** A theoretical analysis about FHS and has been given in the above section, but the implementation with several representative sequences of various motion contents can provide a realistic and interesting evaluation. The following subsections will demonstrate the simulation results of FHS, MFHS, MDS and their comparisons with other reported search algorithms.

3.1. **Simulation Results of FHS.** To manifest the flatted hexagonal search pattern to be appropriate for proposed multipath search algorithm, the simulation and comparison with other well-known search patterns are made in this subsection. For the purpose of comparison, three previous fast BMAs including DS, HEXBS and CDS and the proposed FHS algorithm are simulated by using the luminance component of six popular sequences: "Salesman" (CIF, 499 frames), "Foreman" (CIF, 300 frames), "Coastguard" (CIF, 300 frames), "Garden" (SIF, 115 frames), "Tennis" (SIF, 67 frames) and "Football" (SIF, 125 frames), with various motion types. For each sequence, the search is performed at a block size of 16 × 16 within a window of size ±7. Evaluation with such six sequences in terms of MAD (mean absolute distortion) used as the BDM, matching probability and number of search points is described in Table 2. In addition to the above frequently-used evaluation terms, a new measure of tradeoff between search speed and matching probability, i.e., a product of both speed and probability, is introduced. For making a reasonable comparison of those search algorithms, the speed-probability (SP) product is defined as follows:

$$SP = (N_{FS}/N_{way}) \times (P_{way}/P_{FS}) \tag{3}$$

Where NFS and Nway denote the number of search points required for the FS algorithm and the compared search way, respectively, and PFS and Pway mean the matching probabilities of FS and the compared search way, respectively. Both ratios of NFS/Nway and Pway/PFS imply the rates of speed and probability enhancement compared to the FS algorithm, respectively. From equation (3), it reveals that the larger the SP value is, the more the improvement over FS can be achieved.

From Table 2, it can be observed that the FHS algorithm is superior to other three fast BMAs, i.e., DS, HEXBS and CDS, in terms of SP improvement over FS on average. However, in the "Salesman" sequence with a highly center-biased motion vector distribution in which there is about 96% of the motion vectors found in the central 5-point cross region, CDS has the lowest MAD and highest matching probability and even fewer search points than other three fast BMAs. The major reason is that the CDS algorithm uses the halfway-stop technique for searching within the 5-point cross area. Though, the search speed of CDS will be largely degraded if there is no a high motion-vector distribution over the 5-point cross region in the sequence. For image sequences with a moderate or high probability of horizontal-biased motions, such as "Foreman", "Tennis", "Football", "Coastguard" and "Garden", the proposed FHS algorithm always gives a larger SP value than others.

For a fair comparison, only DS, HEXBS and FHS are considered, excluding CDS, because these three algorithms all adopt a simple monotonous searching strategy with the unitary searching-pattern and the identical ending-pattern during the block-matching process. In respect of application, a monotonous search algorithm is always interesting and wide-applicable owing to its easy realization while a complicated algorithm may be not attractive since it likely needs a high-cost realization, especially for chip implementation. From Table 2, Table 3 particularly tabulates the average SIR (speed improvement rate), average MAD changed, average MP (matching probability) increment and average SP increment in percentage of the proposed FHS over DS and HEXBS. As shown in Table II and Table 3, the proposed FHS algorithm has the better SP value and always achieves substantial probability improvement of up to 18% over HEXBS with a slight speed degradation of less than 8% and a significant speed improvement of at least 21% over DS with a little probability decrement of less than 7%. The FHS algorithm can provide the best SP increase, about 26% and 12%, of FHS over DS and HEXBS, respectively, for the "Garden" sequence. This points out that the SP improvement will be enhanced with the increasing degree of horizontal-biased motions.

Figure 6(a) and (b) plot the average number of search points per block and the average MAD difference per pixel from subtracting MAD of FS, respectively, in a way of frame-by-frame comparison for the "Garden" sequence. These curves demonstrate that the

Table 2. Simulation results of FHS and other BMAs

Salesman 352288 449frames

BMAs	MAD	Probability(%)	Points(%)	SP
FS	2.773	100.000	204.283	1
DS	2.813	95.227	12.890	15.090
HEXBS	2.825	94.287	10.564	18.232
CDS	2.782	97.578	9.414	21.173
FHS	**2.784**	**96.589**	**10.637**	**18.548**

Foreman 352288 300frames

BMAs	MAD	Probability(%)	Points(%)	SP
FS	4.144	100.000	204.283	1
DS	4.372	87.349	17.237	10.352
HEXBS	4.617	69.329	12.956	10.931
CDS	4.389	86.138	15.929	11.046
FHS	**4.552**	**81.890**	**14.036**	**11.918**

Coastguard 352288 300frames

BMAs	MAD	Probability(%)	Points(%)	SP
FS	4.798	100.000	204.283	1
DS	4.852	98.825	16.384	12.322
HEXBS	4.900	96.601	12.822	15.389
CDS	4.854	98.771	15.713	12.841
FHS	**4.884**	**98.359**	**12.994**	**15.462**

Garden 352240 115frames

BMAs	MAD	Probability(%)	Points(%)	SP
FS	8.413	100.000	202.048	1
DS	8.656	92.894	16.712	11.230
HEXBS	9.322	81.470	13.069	12.594
CDS	8.598	93.923	15.021	12.633
FHS	**8.595**	**93.460**	**13.333**	**14.162**

Tennis 352240 67frames

BMAs	MAD	Probability(%)	Points(%)	SP
FS	4.809	100.000	202.048	1
DS	5.045	91.129	16.309	11.290
HEXBS	5.438	75.238	12.891	11.792
CDS	5.102	89.141	15.516	11.607
FHS	**5.338**	**84.954**	**13.415**	**12.794**

Football 352240 125frames

BMAs	MAD	Probability(%)	Points(%)	SP
FS	10.065	100.000	202.048	1
DS	10.513	91.307	15.968	11.552
HEXBS	10.822	79.029	12.387	12.890
CDS	10.599	89.836	14.319	2.676
FHS	**10.688**	**87.878**	**13.151**	**13.501**

Table 3. Average SIR, MAD changed and SP increment of FHS over (a) DS and (b) HEXBS

		Salesman	Foreman	Coastguard	Garden	Tennis	Football
Avg. SIR(%)	(a)	21.18	22.81	26.09	25.34	21.57	21.42
	(b)	-0.69	-7.69	-1.32	-1.98	-3.91	-5.81
Avg. MAD Changed(%)	(a)	-1.03	4.12	0.66	-0.70	5.81	1.66
	(b)	-1.45	-1.41	-0.33	-7.80	-1.84	-1.24
Avg. MP Increase(%)	(a)	1.43	-6.25	-0.47	0.61	-6.78	-3.76
	(b)	2.44	18.12	1.82	14.72	12.91	11.20
Avg. SP Increase(%)	(a)	22.92	15.13	25.48	26.11	13.32	16.87
	(b)	1.73	9.03	0.47	12.45	8.50	4.74

proposed FHS is substantially superior to DS and CDS but similar to HEXBS in terms of number of search points and gives a similar distortion error as both DS and CDS but an evident improvement over HEXBS.

The above experimental results manifest the superiority of the proposed FHS algorithm to other BMAs, when both speed and quality are considered simultaneously, for the most real-world sequences. This reveals that the flatted hexagonal pattern is more suitable to be a search pattern employed in the proposed multipath search algorithm.

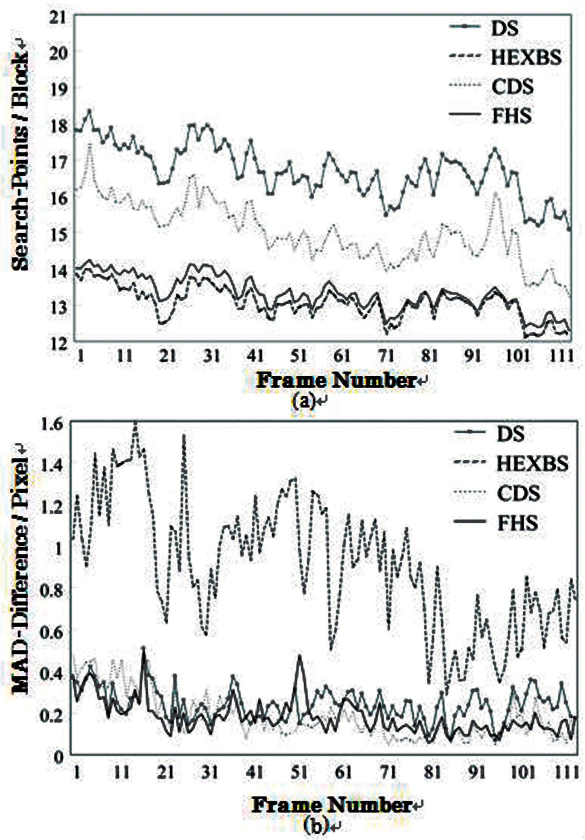


Figure 6. Comparison of DS, HEXBS, CDS and the proposed FHS for SIF sequence "Garden" in terms of: (a) the average number of search points per block and (b) the average MAD difference per pixel from subtracting MAD of FS.

3.2. **Simulation Results of MFHS.** By employing the flatted hexagonal search pattern in the multipath search process, MFHS will cope with the local-minimal trapping problem to achieve a high matching probability with a moderate number of search points required. For comparison, FS, five previous fast BMAs including 3SS, 4SS, DS, HEXBS and CDS, and the proposed MFHS algorithm with various values of β are simulated by using the luminance component of five popular sequences mentioned in the above subsection: "Salesman", "Coastguard", "Garden", "Tennis", and "Football". Evaluation using these such five sequences in terms of MSE (mean square error) used as the BDM, matching probability and number of search points is shown in Table 4, in which the search is performed at a block size of 16×16 within a window of size ± 7.

Obviously, Table 4 shows that the MFHS algorithm can provide a higher matching probability than other fast BMAs with only a slight increase in checking points and approaches FS in matching-probability but requires very less checking points, when compared to FS. In the four sequences of "Salesman", "Coastguard", "Garden" and "Football", containing various degrees of motions, MFHS provides a higher matching probability than those fast BMAs by using $\beta = 0.05$. For the "Tennis" sequence, involving a stronger zooming capture, the matching-probability of MFHS is greater than other fast BMAs by increasing β to 0.1.

Figure 7(a) and (b) plot the average number of checking points per block and the average matching probability per block, respectively, using various values of β in the proposed MFHS for those sequences discussed in Table 4. These curves demonstrate that the number of checking points increases about linearly with the increasing value of β and the increase of the matching probability will be saturated at $\beta = 1$ or so. In $\beta = 1$, the matching probability will be very near to that of FS for all sequences except "Tennis". Owing to zooming capture in the "Tennis" sequence, it requires a greater number of checking point but achieves a lower matching probability when comparing with the other sequences. This is because that zooming will lead to a radiate distribution of local-minima, which needs more search-paths and also causes more mismatches. Besides, a sequence containing large motions will also require a greater quantity of checking points, as illustrated in the curve of "Football". However, another superiority of MFHS to other suboptimal BMAs is the adjustability in search speed and accuracy by changing β.

Figure 8 demonstrates a visual comparison of using various values of β in the 66th frame of the SIF sequence "Tennis" motion-compensated for: (a) the 66th frame of the original sequence (non-compensated); (b) FS, MSE = 155.7; (c) MFHS by $\beta = 0$ (i.e., FHS), MSE = 233.2; (d) MFHS by $\beta = 0.1$, MSE = 198.1; (e) MFHS by $\beta = 0.22$, MSE = 178.8; (f) MFHS by $\beta = 0.36$, MSE = 163.2. It can be observed that the picture quality of MFHS is almost identical to that of FS, when using $\beta = 0.36$. In such a case, from Table 4 it also shows that MFHS only needs about 29 checking points but 202 checking points are required by FS, with a matching probability of 96.7% compared to 100% assumed in FS. For averaging those simulation results in Table 4, it manifests that the proposed MFHS algorithm can achieve an average matching probability up to 98% near to that of FS and about 10 times of checking points faster than FS. It should be pointed out that the search scheme with pre-determination of an initial search point is not involved in the above comparison owing to fairness. However, the strategy of pre-determining an initial search point can be also introduced into the MFHS algorithm to increase the search efficiency.

Table 4. Simulation results of MFHS and other BMAs.

Salesman, CIF, 499 frames			
BMA	**MSE**	**Probability**	**Points**
FS	17.965	1.000	204.283
3SS	18.850	0.946	23.212
4SS	18.704	0.953	16.144
DS	18.654	0.952	12.890
HEXBS	18.749	0.943	10.564
CDS	18.232	0.976	9.414
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	17.971	0.998	23.914
β**=0.5**	17.975	0.991	14.413
β**=0.36**	18.011	0.987	12.789
β**=0.22**	18.044	0.984	11.912
β**=0.1**	18.102	0.980	11.417
β**=0.55**	18.152	0.977	11.078
β**=0**	18.227	0.966	10.637

Garden, SIF, 115 frames			
BMA	**MSE**	**Probability**	**Points**
FS	281.781	1.000	202.048
3SS	333.443	0.834	23.204
4SS	308.586	0.868	18.696
DS	295.254	0.929	16.712
HEXBS	324.485	0.815	13.069
CDS	294.841	0.939	15.021
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	281.888	0.997	38.988
β**=0.5**	282.325	0.991	26.175
β**=0.36**	283.039	0.987	22.583
β**=0.22**	285.064	0.980	18.651
β**=0.1**	287.398	0.968	15.253
β**=0.55**	289.598	0.956	14.115
β**=0**	292.974	0.935	13.333

Coastguard, CIF, 300 frames			
BMA	**MSE**	**Probability**	**Points**
FS	63.198	1.000	204.283
3SS	67.259	0.974	23.375
4SS	66.898	0.984	18.500
DS	66.111	0.988	16.509
HEXBS	68.135	0.966	12.909
CDS	66.123	0.988	15.917
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	281.888	0.999	28.473
β**=0.5**	282.325	0.998	18.026
β**=0.36**	283.039	0.997	16.441
β**=0.22**	285.064	0.995	15.047
β**=0.1**	287.398	0.992	13.972
β**=0.55**	289.598	0.989	13.512
β**=0**	292.974	0.984	13.075

Football, SIF, 125 frames			
BMA	**MSE**	**Probability**	**Points**
FS	372.692	1.000	202.048
3SS	411.197	0.886	23.104
4SS	413.459	0.897	18.097
DS	416.761	0.913	15.968
HEXBS	437.712	0.790	12.387
CDS	425.528	0.898	14.319
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	372.893	0.998	68.017
β**=0.5**	373.703	0.994	38.656
β**=0.36**	375.545	0.989	31.568
β**=0.22**	380.985	0.981	24.091
β**=0.1**	394.986	0.948	17.648
β**=0.55**	409.136	0.920	15.128
β**=0**	436.618	0.879	13.151

Tennis, SIF, 67 frames			
BMA	**MSE**	**Probability**	**Points**
FS	116.944	1.000	202.048
3SS	116.944	0.729	23.111
4SS	139.890	0.850	18.455
DS	132.398	0.911	16.309
HEXBS	150.450	0.752	12.891
CDS	134.068	0.891	15.516
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	117.446	0.981	69.430
β**=0.5**	118.993	0.967	36.904
β**=0.36**	121.438	0.960	29.262
β**=0.22**	126.981	0.944	22.340
β**=0.1**	130.967	0.910	16.610
β**=0.55**	135.831	0.883	14.660
β**=0**	141.779	0.850	13.415

Table 5. Simulation results of MDS and other BMAs.

Tennis (CCIR601)			
BMA	**MSE**	**Probability**	**Points**
FS	162.309	1	213.479
3SS	194.145	0.735	24.058
4SS	185.764	0.770	16.464
DS	197.392	0.711	12.853
HEXBS	191.045	0.744	14.243
CDS	199.122	0.708	13.174
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	162.349	0.997	98.020
β**=0.5**	162.758	0.988	63.384
β**=0.36**	163.681	0.976	51.376
β**=0.22**	167.035	0.947	38.077
β**=0.1**	173.094	0.893	24.616
β**=0.55**	178.263	0.840	19.701

Garden (CCIR601)			
BMA	**MSE**	**Probability**	**Points**
FS	142.496	1	213.479
3SS	178.941	0.763	24.161
4SS	171.145	0.860	20.582
DS	172.140	0.826	15.326
HEXBS	172.445	0.855	21.808
CDS	174.738	0.846	15.614
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	142.598	0.995	69.094
β**=0.5**	143.057	0.989	51.25
β**=0.36**	143.642	0.984	44.580
β**=0.22**	145.131	0.972	35.930
β**=0.1**	149.365	0.940	26.273
β**=0.55**	154.828	0.905	22.868

Mobile (CCIR601)			
BMA	**MSE**	**Probability**	**Points**
FS	39.008	1	213.479
3SS	48.668	0.827	24.065
4SS	39.610	0.931	16.431
DS	46.048	0.741	13.223
HEXBS	39.453	0.936	12.850
CDS	39.692	0.926	13.477
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	39.112	0.996	71.872
β**=0.5**	39.116	0.995	46.447
β**=0.36**	39.126	0.994	39.865
β**=0.22**	39.144	0.992	31.642
β**=0.1**	39.278	0.982	22.196
β**=0.55**	39.404	0.960	18.571

Football (CCIR601)			
BMA	**MSE**	**Probability**	**Points**
FS	305.439	1	213.479
3SS	332.185	0.830	24.080
4SS	339.107	0.845	18.582
DS	348.651	0.735	13.769
HEXBS	346.765	0.830	17.457
CDS	358.865	0.721	14.207
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	305.474	0.999	89.554
β**=0.5**	305.682	0.997	64.072
β**=0.36**	306.087	0.994	54.934
β**=0.22**	307.953	0.984	43.151
β**=0.1**	314.03	0.954	30.339
β**=0.55**	321.644	0.921	24.232

Susie (CCIR601)			
BMA	**MSE**	**Probability**	**Points**
FS	21.541	1	213.479
3SS	25.105	0.638	24.078
4SS	23.561	0.741	17.652
DS	24.332	0.595	13.501
HEXBS	23.814	0.731	16.056
CDS	24.112	0.650	14.001
MFHS(β)	**MSE**	**Probability**	**Points**
β**=1**	21.548	0.997	121.485
β**=0.5**	21.613	0.991	95.049
β**=0.36**	21.674	0.985	84.495
β**=0.22**	21.829	0.971	65.528
β**=0.1**	22.220	0.92	35.398
β**=0.55**	22.646	0.851	24.186

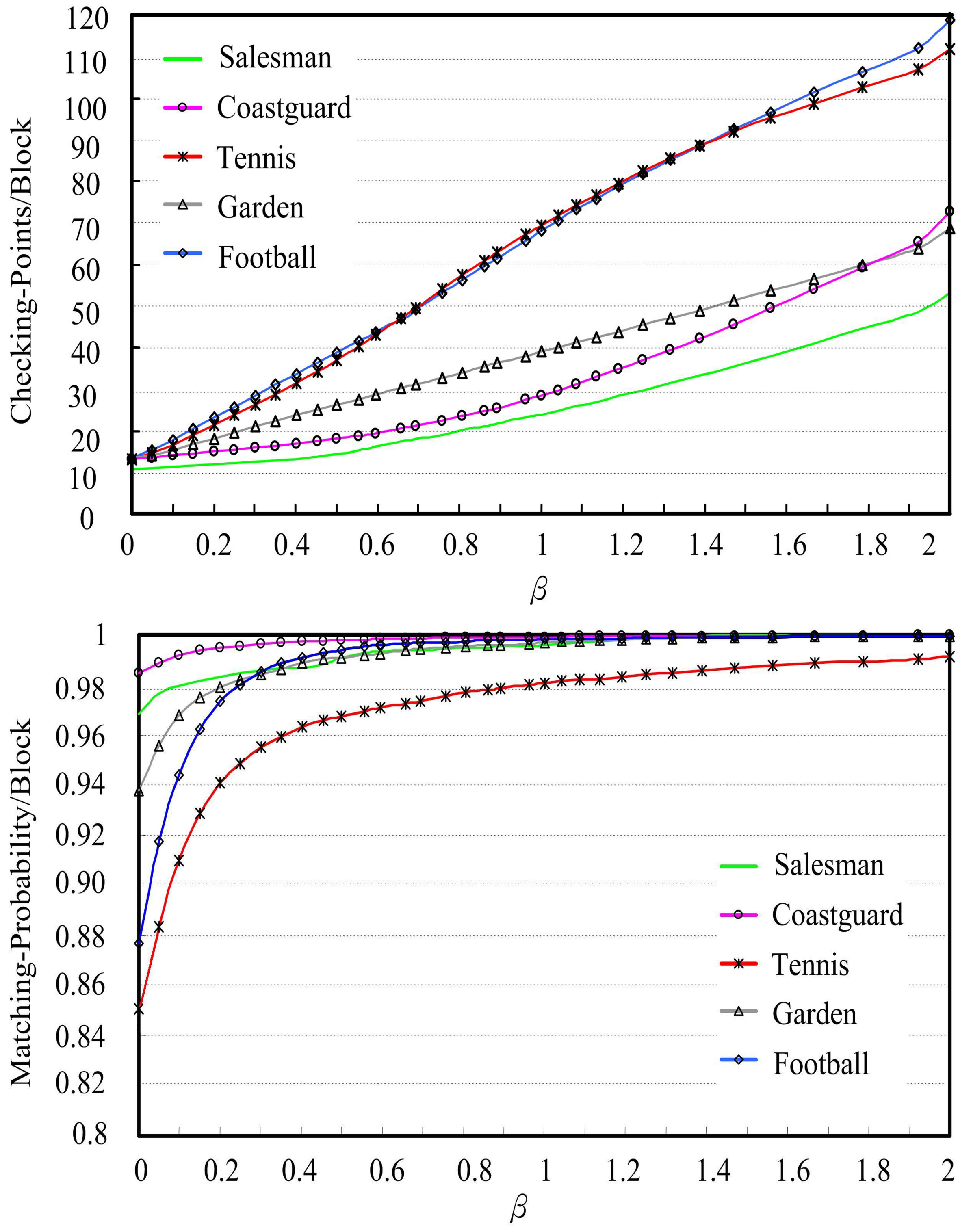


FIGURE 7. The searchspeed/accuracy performance of MFHS in various values of β for SIF sequences. (a) Average number of checking points per block. (b) Average matching probability per block

(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 8. A visual comparison of using various values of β in the SIF sequence "Tennis" motion-compensated for : (a) the 66th frame of the original sequence (non-compensated) ; (b) FS, MSE = 155.7; (c) MFHS,β=0 , MSE = 233.2; (d) MFHS,β0.1 , MSE = 198.1; (e) MFHS,β=0.22 , MSE = 178.8; (f) MFHS,β=0.36 , MSE= 163.2.

3.3. **Simulation Results of MDS.** If the frame size becomes larger, like CCIR601 format (i.e., 720×480) or so, the search performance of MFHS will be degraded owing to its small-size search pattern. Using such a frame size, the diamond will be the best search pattern for the multipath search algorithm. The search accuracy of MDS is better than MFHS but the search speed is only slightly smaller than that of MFHS in most real-world sequences of CCIR601 format. The major reason is that the search-pattern of MDS is bigger than that of MFHS, where a larger search-pattern is more suitable for searching in a higher-resolution sequence.

Table 5 shows evaluations of MDS by comparing with six BMAs including FS, 3SS, DS, HEXBS, CDS, and FHS described previously through using five sequences of CCIR601-format: "Susie", "Mobile", "Tennis", "Garden" and "Football", where these BMAs search the optimal point using a block size of 16x16 within a window of size ± 7. In the sequence "Susie", the monotonousness degree of error surface is slight and thus it results in many local optimal points, besides, it also has a low zero motion vector distribution of about 9.22%. For this reason, using such a sequence will generate lower matching probabilities for those fast BMAs than other sequences. Also, this makes MDS to need more checking points because there are more candidates of the optimal path to search. Considering the sequence "Mobile", there exists a large quantity of small horizontal motions, i.e., a high distribution of motion vectors (± 1, 0), about 64.36%. Hence, those fast BMAs with a search pattern of small c/n ratio, such as DS, CDS and FHS, achieve a higher matching probability than using the other sequences. For the other three sequences of "Tennis", "Garden", and "Football", the reduction in search accuracy of FHS when using the CCIR601 sequence is more significant than that of other fast BMAs, compared to using the previous two sequences of "Susie" and "Mobile". This is because that FHS employs a small-size search pattern. On the other hand, DS has the best search accuracy among those fast BMAs compared in Table 5 for all sequences and thereby the diamond pattern is selected as the search pattern adopted in the multipath search algorithm for using the CCIR-601 sequences. For those five sequences in Table 5, MDS provides a greater matching probability than other fast BMAs by using $\beta = 0.05$ and approach FS in matching probability, i.e., above 0.98, by increasing β to 0.36. Generally, people can't almost visually discriminate the difference between two motion-compensated results of using motion vectors searched by FS and MDS when the matching probability is above 0.96.

Figure 9 describes the search speed/accuracy performance of MDS in various values of β using five CCIR601 sequences, "Susie", "Mobile", "Tennis", "Garden" and "Football", involving different degrees of motion, where the search speed and search accuracy are evaluated by average number of checking points per block and average matching probability per block, respectively. Like MFHS, these curves of MDS in Figure 9 shows that the number of checking points increases about linearly with the increasing value of β, except the curve of "Susie", and the increase of the matching probability will be saturated at $\beta = 1$ or so. Owing to having only slight monotonousness of error surface in the "Susie" sequence, it makes MDS to require more checking points to search more possible optimal paths, i.e., the curve has a larger slope than other curves of sequences, when β is below 0.4. The matching probability is almost identical to that of FS, i.e., above 0.99, for all sequences when β is adjusted up to 1.

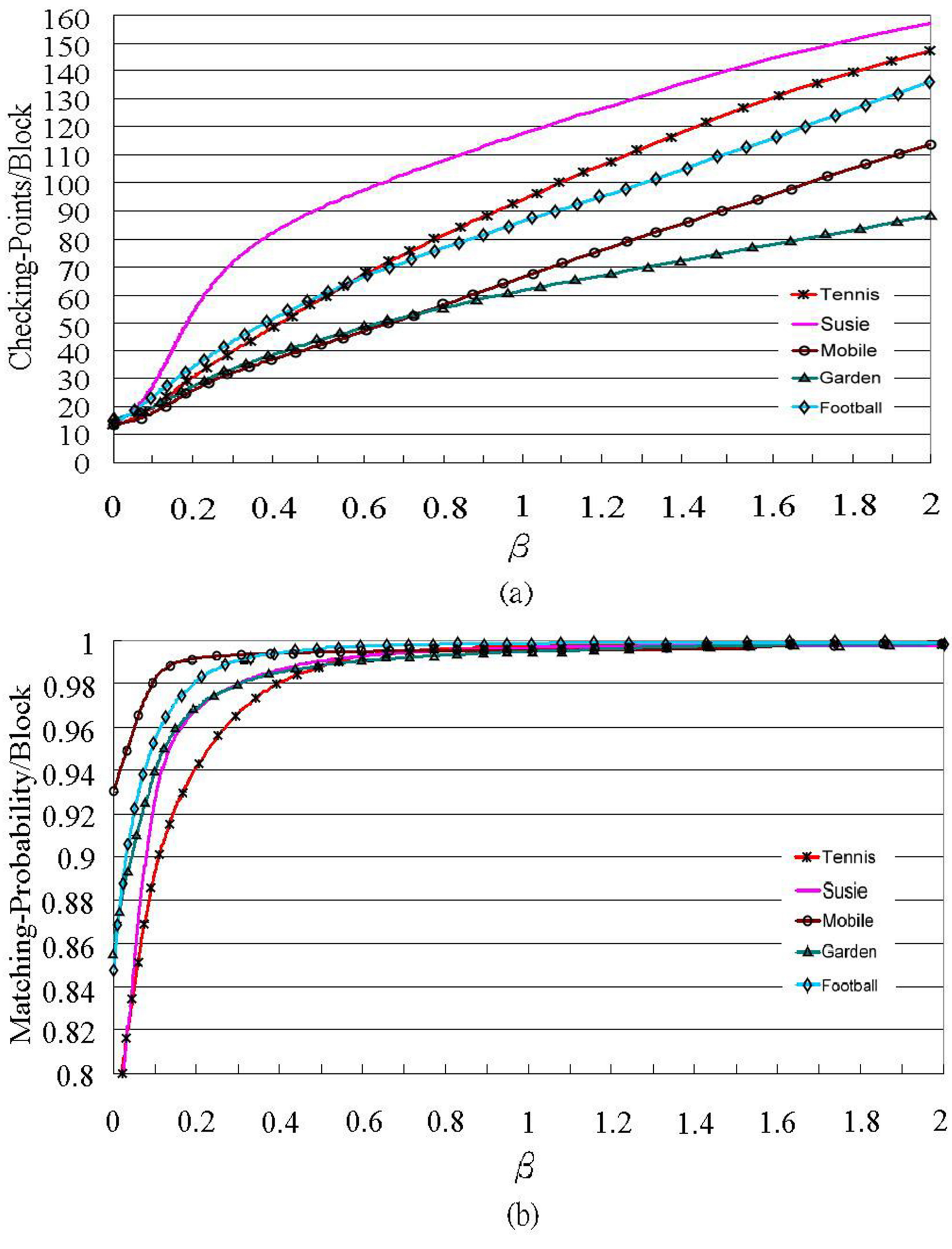


FIGURE 9. The search speed/accuracy performance of MDS in various values of β for CCIR601 sequences. (a) Average number of checking points per block. (b) Average matching probability per block.

3.4. **Discussions.** From the above experimental results, the proposed multipath search algorithm can achieve an average matching probability up to 98and about 10 times of checking points faster than FS in most of real-world sequences. Besides, the BDM threshold will be adjustable for the purpose of adjusting the search speed and search accuracy specified in the certain applications. However, the proposed multipath search algorithm will spend more search time when meeting a low-contrast image sequence, in which there are more local minimum BDM points needed to be checked. In a low-contrast image sequence, it will generate many quasi-zero motion vectors because the pixel difference is very slight. To cope with this problem, a zero-motion prejudgment [20] may be employed to initially find zero motion vectors to avoid further searching for the optimal point. This is because most video compression standards only require the motion vector that is acceptable, not the optimal.

4. **Conclusions.** In this paper, we have proposed a novel and simple speed/accuracy-adjustable block-matching algorithm based on multipath search scheme. By the multipath search strategy, it can substantially reduce the local-minimum trapping problem to achieve a high matching probability near to FS but still obtain about ten times of search speed faster than FS. It also provides adjustability in search speed/accuracy by a local-minimum decision parameter. Besides, the implementation is not complex because the number of search paths required is determined by a simple decision rule of BDM and each path is searched by use of the identical search pattern. Furthermore, in theory any search pattern can be also used in the proposed multipath search method for various-purpose applications. The experimental results imply that the multipath flatted-hexagon search (MFHS) algorithm and multipath diamond search (MDS) algorithm are suitable for low-resolution (e.g. CIF, 352x288 or SIF, 352x240) and high-resolution (e.g. CCIR601, 720x480)) image sequences, respectively. When the search accuracy (i.e., picture quality) is considered for approaching that of FS, the proposed multipath search algorithm will be more attractive than other suboptimal BMAs.

Acknowledgement. This work is supported by National Science Council under Grant NSC94-2213-E-151-008, Taiwan, R.O.C.. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] *Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s-Part 2: Video*, ISO/IEC 11 172-2 (MPEG-1 Video), 1993.
[2] *Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13 818-2 (MPEG-2 Video), 2000.
[3] *Information Technology-Coding of Audio Visual Objects - Part 2: Visual*, ISO/IEC 14 469-2 (MPEG-4 Visual), 1998.
[4] *Video Codec for Audiovisual Services at $p \times 64$ kbits/s*, ITU-T Recommendation H.261, Mar. 1993.
[5] *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation H.263, Feb. 1998.
[6] Advanced Video Coding, ITU-T Recommendation H.264/ISO/IEC 14496-10, Final Committee Draft, Document JVC-E022, September 2002.
[7] Hui Wang, Joyce Liang and C.-C. Jay Kuo, Overview of robust video streaming with network coding, *Journal of Information Hiding and Multimedia Signal Processing*, vo1. 1, no. 1, pp. 36-50, Jan. 2010.
[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, Motion compensated interframe coding for video conferencing, *Proc. of National Telecommunications Conf.*, pp. G5.3.1- G5.3.5, New Orleans, LA, Nov. 1981.
[9] 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.

[9] 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.

[10] 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.

[11] 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, pp. 419-423, Aug. 1996.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] T. H. Chen and Y. F. Li, A novel flatted hexagon search pattern for fast block motion estimation, *Proc. of IEEE 2004 International Conference on Image Processing (ICIP)*, Singapore, pp. 1477-1480, Oct. 2004.

[18] 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.

[19] C. H. Cheung and L. M. Po, A novel cross-diamond-hexagonal search algorithm for fast block motion estimation, *IEEE Trans. Multimedia*, vol. 7, no. 1, pp.16-22, Feb. 2005.

[20] Y. Nie and K. K. Ma, Adaptive rood pattern search for fast block-matching motion estimation, *IEEE Trans. Image Processing*, vol. 11, no. 12, pp. 1442-1449, Dec. 2002.

[21] C. Zhu, W. Qi, and W. Ser, Predictive fine granularity successive elimination for fast optimal block matching motion estimation, *IEEE Trans. Image Processing*, vol. 14, no. 2, pp. 213-221, Feb. 2005.

[22] C. Zhu and L. M. Po, Minimax partial distortion competitive learning for optimal codebook design, *IEEE Trans. Image Processing*, vol. 7, no. 10, pp. 1400-1409, Oct. 1998.

[23] C. Zhu, A new subsampling-based predictive vector quantization for image coding, *Proc. of Signal Processing: Image Communication (EURASIP)*, vol. 17, no. 6, pp. 477-484, July 2002.

[24] T. H. Chen, A cost-effective 3-step hierarchical search block-matching chip for motion estimation, *IEEE Journal of Solid-State Circuits*, vol. 33, no. 8, pp. 1253-1258, Aug. 1998.