Transactions Letters

Fast Motion Estimation Robust to Random Motions Based on a Distance Prediction

Yun-Gu Lee and Jong Beom Ra, *Senior Member, IEEE*

***Abstract*—For fast motion estimation, a gradient descent search is widely used due to its high efficiency. However, since it does not examine all possible candidates within a search area, it suffers from PSNR degradation for sequences having fast and/or random motions. To alleviate this problem, we propose a hybrid search scheme wherein a hierarchical search scheme is selectively combined with an existing gradient descent search. For the selective combination, we introduce a measure estimating the distance between the current search point and the optimal point. Since this measure greatly reduces the need to perform hierarchical searches, their computational burden is not noticeable in the overall motion estimation while their contribution to the PSNR improvement is considerable. Using the estimated distance, we can also noticeably improve the early termination performance in a local search. Experimental results show that the proposed algorithm outperforms the other popular fast motion estimation algorithms in terms of both PSNR and search speed, especially for sequences having fast or random motions.**

***Index Terms*—Block matching, diamond search, fast motion estimation, hierarchical search, random motion vector.**

I. Introduction

IN VIDEO CODING standards, such as H.263 and MPEG-1/2/4 [1]–[4], the block matching algorithm (BMA) is usually adopted to reduce temporal redundancy. In the BMA, a current frame to be encoded is divided into nonoverlapped blocks and the best matching block of each current block is found within the previous frame. Since the full search block matching algorithm (FSBMA) examines all possible candidates within a search area to find a motion vector (MV), it provides optimal PSNR performance, but requires a heavy computational burden. Hence, over a period of many years, fast motion estimation (ME) algorithms have been developed to reduce the computational complexity. These algorithms can be divided into several categories according to the employed approach. Among them, a fast search by limiting the candidates of MVs under the unimodal error-surface assumption is usually adopted due to its high speed-up ratio [5]–[14].

Manuscript received July 20, 2004; revised November 5, 2005. This work was supported in the part by the University IT research center program of the Government of Korea. This paper was recommended by Associate Editor C. W. Chen.

The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: yglee@issserver.kaist.ac.kr; jbra@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TCSVT.2006.877149

In these algorithms, since different shapes and sizes of search patterns significantly affect the search speed and PSNR in ME, appropriate selection of search patterns is critical. The large and sparse search pattern used in the three-step search (TSS) often leads the search center in a wrong direction so that TSS may fail to find the optimum point [6]. Meanwhile, the small search pattern of a size of 3×3 in the block-based gradient descent search (BBGDS) can be easily trapped into a local minimum for sequences having fast or random motions [8]. To overcome this problem, the diamond search (DS) first finds a MV using a large diamond search pattern (LDSP), and then adopts a small diamond search pattern (SDSP) to refine the search result [9]. Similar approaches, the hexagon-based search (HEXBS) [10] and the cross-diamond search (CDS) [12] have also been proposed. Compared to DS, they somewhat improve the search speed while maintaining PSNR performance.

To improve the search performance, BBGDS, DS, HEXBS, and CDS focus on search patterns with an initial search center of (0, 0). Meanwhile, the proper selection of an initial search center is addressed by considering the neighborhood information [13], [15]. In addition, the motion vector field adaptive search technique (MVFAST) attempts to reduce the computational load by introducing a thresholding technique for terminating the search early [13]. By adopting a better initial search center and an adaptive thresholding technique, the predictive MVFAST (PMVFAST) is an improved version of MVFAST. Here, it should be noted that all the algorithms mentioned above, i.e., TSS, BBGDS, DS, HEXBS, CDS, MVFAST, and PMVFAST, do not examine all possible candidates within a search area. Hence, their PSNR performances tend to be degraded for sequences having fast or random motions compared to the FSBMA or a hierarchical approach [16]–[18], which attempts to examine the overall search area by sacrificing the search speed.

In this paper, we propose a new hybrid search to provide a fast search speed and smaller distortion errors. The proposed algorithm properly combines an existing gradient descent search (GDS) and a hierarchical search, by using a newly defined distance between a current search point and the optimal point. It also adaptively terminates the search according to the measured distance. Thereby, it is especially proper for the sequences having fast or random motions compared to the existing GDS-based algorithms. In Section II, we describe the proposed algorithm. Experimental results are given in Section III. Finally, conclusions are presented in Section IV.

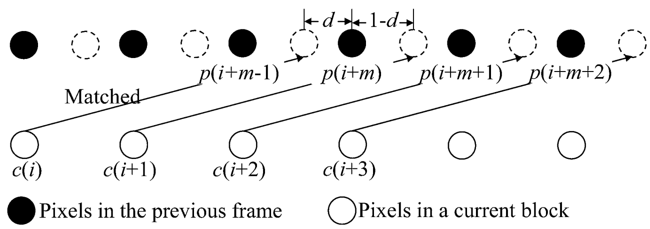


Fig. 1. Matching positions of current block pixels in the previous frame.

II. Proposed Algorithm

In this section, we first introduce a scheme to estimate the distance between a search point and the optimal point, and describe a novel decision procedure for early termination based on the estimated distance. We then explain the adopted hierarchical scheme for a random MV search. Finally, we describe the proposed ME algorithm.

A. Distance Estimation Scheme

To simplify distance estimation, we consider this problem as a one-dimensional one. We assume that there are only translation motions between the previous and current frames, and that all of the pixels within a block have the same motion. We also assume that the pixels in the previous frame can be reconstructed by the linear interpolation of pixels in the current block. Under these assumptions, the distance between the search point and the optimal point can be estimated. Fig. 1 shows the matched pixel positions in the previous frame, $p(i+m-d)$, of the current block pixels, $c(i)$, where m and d denote an integer MV and a sub-pixel MV, respectively. So, $(m-d)$ is the true MV of the current block, and the corresponding dotted circles represent the exactly matched points in the previous frame. From Fig. 1, we can describe that the following relation:

$$p(i+m) = c(i)(1-d) + c(i+1)d, i \in \text{block}. \quad (1)$$

In the integer-pixel ME, the minimum sum of absolute differences (SAD) value at MV m is given as

$$\text{SAD}(m) = \sum_{i \in \text{block}} |p(i+m) - c(i)|. \quad (2)$$

From (1) and (2), we obtain

$$\begin{aligned}\text{SAD}(m) &= \sum_{i \in \text{block}} |c(i)(1-d) + c(i+1)d - c(i)| \\ &= d \sum_{i \in \text{block}} |c(i+1) - c(i)| = d \times \text{SADNP} \quad (3)\end{aligned}$$

where SADNP denotes the sum of absolute differences between neighboring pixels and equals $\sum_{i \in \text{block}} |c(i+1) - c(i)|$.

Therefore, the distance can be estimated as

$$d = \frac{\text{SAD}(m)}{\text{SADNP}}. \quad (4)$$

It should be noted that (4) holds true if the distance between the current search point and the optimal MV is less than 1 pixel. However, if we assume that the SAD value monotonously increases starting from the optimal MV, we can still know from the estimated distance d, which is larger than 1 pixel, that a current search point locates more than 1 pixel off from the optimal one. Note here that d is not quantitatively meaningful aside from the information above.

Since ME for video coding is performed in two-dimensional (2-D) space, the derivation above should be extended into the two-dimensional case. In the 2-D space, the distance between a current search point and the optimal point is denoted as (d_x, d_y). And the horizontal SADNP and the vertical SADNP are given as follows:

$$\begin{aligned}\text{SADNP}_x &= \sum_{(i,j) \in \text{block}} |c(i+1,j) - c(i,j)|, \\ \text{SADNP}_y &= \sum_{(i,j) \in \text{block}} |c(i,j+1) - c(i,j)| \quad (5)\end{aligned}$$

where $c(i,j)$ denotes a pixel at the current block. Then, each component of the distance is independently predicted as follows:

$$d_x = \frac{\text{SAD}(m,n)}{\text{SADNP}_x} \quad \text{and} \quad d_y = \frac{\text{SAD}(m,n)}{\text{SADNP}_y} \quad (6)$$

where (m,n) denotes the current search point. According to (6), $\text{SAD}(m,n)$ should be zero if d_x (or d_y) is zero. However, even though d_x (or d_y) is zero, or the x (or y) component of the current search point is matched to that of the optimal point, $\text{SAD}(m,n)$ may not be zero if the $y(\text{or}x)$ component of the current search point is not matched to that of the optimal point. Therefore, the predicted d_x (or d_y) could be larger than the real distance.

B. Adaptive Thresholding Techniques

In MVFAST, if the block SAD value at the search point (0, 0) is smaller than a fixed value of 512, a search is stopped without additionally examining search points. However, this fixed threshold value can be too large or too small depending on the block characteristic. In particular, most of the minimum SAD values for the blocks having fast or random motion are usually larger than 512. Hence, a thresholding technique adopted in MVFAST does not improve the search speed for those sequences. To solve this problem, an adaptive thresholding technique is proposed in PMVFAST. In this algorithm, the threshold value of a current block is determined among the minimum SAD values of the three adjacent (or three left, top, and top-right) blocks so that it may efficiently terminate a search for blocks having fast or random motions. However, we should notice that the thresholding value is determined by only considering the SAD values of neighboring blocks, without taking into account its own characteristic of the current block. Hence, the determined value may not be suitable for every block. In particular, if its characteristic is different from those of their neighboring blocks, the improper thresholding value may cause PSNR degradation or limit the speedup performance.

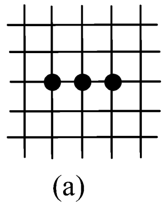


Fig. 2. One-dimensional search patterns: (a) horizontal and (b) vertical.

To alleviate this problem, we propose a new adaptive thresholding scheme based on estimated distances described in the previous subsection. In this scheme, the displacement d_x and d_y are predicted using (5) and (6). Then, if d_x (or d_y) is smaller than the $1/2$ pixel size, we regard the x (or y) component of the current search point as that of the optimal point having an integer-pixel accuracy, and a search along the x (or y) direction is not performed. Therefore, if both d_x and d_y are smaller than the $1/2$ pixel size, a search is terminated and a current search point becomes the final integer-pixel MV. However, if only d_x (or d_y) is smaller than a $1/2$ pixel, we perform the one-dimensional search along the y (or x) direction, using the search pattern shown in Fig. 2. After finding the component, the search is terminated. Meanwhile, if both d_x and d_y are larger than a $1/2$ pixel size, we proceed to the next search step based on a search range.

In thresholding, the SAD value of the current search point and the SADNP value of a current block are required. Since the SAD value of the current search point must always be examined, it does not increase the total computational burden. However, the computational load for calculating SADNP is additional. Since it can be reused during ME, SADNP for each block is calculated once. In order to further reduce the computation load of SADNP, we approximate SADNP in (5) as follows:

$$\mathrm{SADNP}'_x = 4\sum_{i=0}^{N/2}\sum_{j=0}^{N/2} |c(2i+1,2j) - c(2i,2j)|$$
$$\mathrm{SADNP}'_y = 4\sum_{i=0}^{N/2}\sum_{j=0}^{N/2} |c(2i,2j+1) - c(2i,2j)|. \quad (7)$$

Then, it is experimentally found that the approximated SADNP' rarely affects thresholding results, and the additional computational load of SADNP' for a block becomes a half of the one-point SAD calculation.

C. The Random MV Search

Since the shape and size of search patterns significantly affect the search speed and matching error in ME, we have performed several experiments to examine their effects on ME. The results are given in Fig. 3. In the figure, NSP denotes the number of search points required for finding a MV. Besides existing fast algorithms, DS, CDS, BBGDS, and HEXBS, we intentionally made a simplified version of DS, $\mathrm{DS_S}$, by utilizing only SDSP rather than using both SDSP and LDSP as in the algorithm DS (see Fig. 4). Experimental conditions for these data are given in detail in Section III-A. The difference among the compared algorithms is only the adopted search pattern. It is noted that DS provides the best PSNR performance among them, while it requires a larger NSP than others. Meanwhile, $\mathrm{DS_S}$ requires a small NSP (or provides a fast search speed) due to its simple search pattern. But its PSNR performance is the worst, since it has more chances to be trapped into a local minimum.

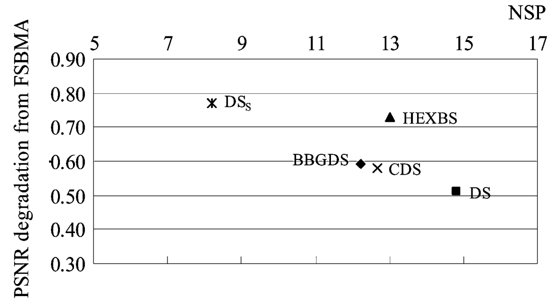


Fig. 3. PSNR degradation from FSBMA versus the number of search points for various fast search algorithms.

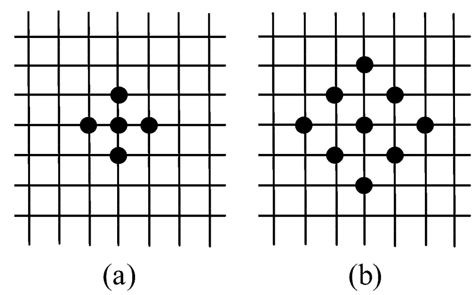


Fig. 4. Diamond search patterns: (a) SDSP and (b) LDSP.

To compare the probabilities to be trapped into a local minimum in $\mathrm{DS_S}$ and DS, we examine the consistency of MVs of $\mathrm{DS_S}$, DS and FSBMA under the same experimental conditions, as in Fig. 3. The consistency of MVs between $\mathrm{DS_S}$ and FSBMA is 86.01%, and 89.43% between DS and FSBMA. Hence, the consistency difference between DS and $\mathrm{DS_S}$ is only 3.42% on average. This means that most of the MVs can be correctly found using SDSP, and LDSP is not always needed to find a correct MV. Therefore, the proposed algorithm first finds a MV using SDSP. Then, if it is found that the searched MV corresponds to a local minimum or the block has a random motion, a random MV searching procedure is performed to find a better MV instead of a large pattern based search.

Since neighboring blocks are usually highly correlated, the SAD value of a current block is expected to be similar to those of the neighboring blocks. Thus, the existence of a random MV can be examined by comparing the SAD value of the current block, which is calculated from a MV using SDSP, with those of neighboring blocks. If the SAD value from the MV using SDSP, $\mathrm{SAD_{SDSP}}$, is greater than $\mathrm{SAD_{MAX}}$, the MV is considered incorrect and a random MV search is performed. Here, $\mathrm{SAD_{MAX}} = \mathrm{MAX}(\mathrm{SAD_{NMV1}}, \mathrm{SAD_{NMV2}}, \mathrm{SAD_{NMV3}}, \mathrm{SAD_P})$, and $\mathrm{SAD_{NMV1}}, \mathrm{SAD_{NMV2}}$, and $\mathrm{SAD_{NMV3}}$ denote the minimum SAD values at the left, top, and top-right neighboring blocks of the current block, respectively, and $\mathrm{SAD_P}$ is the minimum SAD value of the block at the same position in the previous

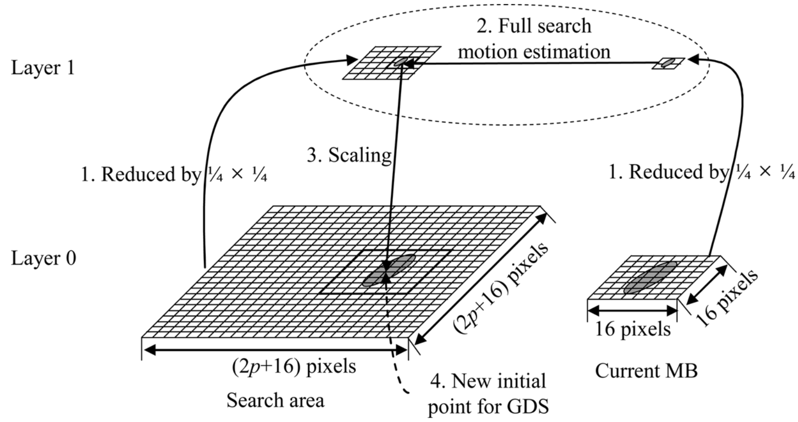


Fig. 5. Procedure proposed for obtaining a new initial point.

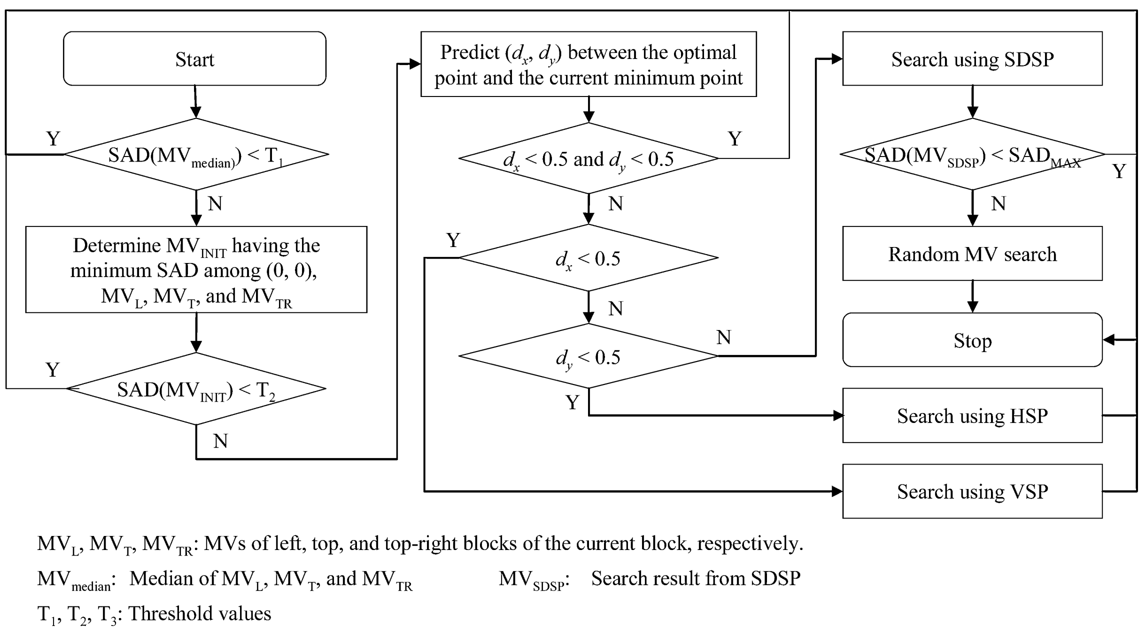


Fig. 6. Overall structure of the proposed ME algorithm. (a) Original image. (b) Full search algorithm. (c) DS, (d) MVFAST, (e) PMVFAST. (f) Proposed algorithm.

frame. In most cases, $\mathrm{SAD_{SDSP}}$ is smaller than $\mathrm{SAD_{MAX}}$, and the random MV search is usually not performed.

To efficiently perform the random MV search, we adopt a hierarchical search scheme [18], and modify it for the proposed algorithm as shown in Fig. 5. The modified scheme has two layers, layer 1 of the image reduced by 1/4 and layer 0 of the original image. In layer 1, we first obtain a new initial search center having the minimum SAD based on FSBMA. In this layer, the search area reduced by 1/4 × 1/4, $s_1(i,j)$ is examined by using a current block reduced size, $c_1(i,j)$. Then, a new initial search center can be found as

$$(m_1, n_1) = \arg\min_{(m_1,n_1)} \sum_{i=0}^{3}\sum_{j=0}^{3} |s_1(i+m_1, j+n_1) - c_1(i,j)| \qquad - \lfloor p/4 \rfloor \le m_1, n_1 \le \lfloor p/4 \rfloor \tag{8}$$

where p denotes the search range of the original image, and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

The initial search center obtained above is scaled up to the original size and considered as an initial search center in layer 0 (see Fig. 5). Then, a local search is performed around the initial center, to find the random MV. Since the new initial search center is obtained by examining all search points in layer 1, it is expected to be close to the global minimum. Thus, a SDSP will be considered adequate in the search in layer 0. Finally, if the minimum SAD obtained from the random MV search is smaller than $\mathrm{SAD_{SDSP}}$, the corresponding MV becomes the final one of the current block.

Since the random MV search is performed only when $\mathrm{SAD_{SDSP}}$ is greater than $\mathrm{SAD_{MAX}}$ and the full search is performed only in the reduced image, the computational complexity for a random MV search is not significant compared with the total complexity. Note that the practical computational burden for a candidate examination in the reduced image is only $1/16$ times that of the original image. In addition, this refinement process is rarely performed.

D. Proposed ME Algorithm

The proposed algorithm is based on the adaptive thresholding techniques and the random MV search scheme described above. Fig. 6 illustrates its overall structure. The first procedure is similar to that of PMVFAST. It predicts $\mathrm{MV_{median}}$ using MVs of the left, top, and top-right blocks of a current block, and if the

TABLE I
COMPARISON OF THE PROPOSED ALGORITHM WITH EXISTING ALGORITHMS. HERE, NSP DENOTES THE NUMBER OF SEARCH POINTS

			FSBMA	BBGDS	DS	HEXBS	CDS	SA	MVFAST	PMVFAST	Proposed
CP	CIF	PSNR	34.07	33.79	33.71	33.40	33.65	33.58	33.75	33.74	33.71
		NSP	869.33	12.29	15.32	13.48	13.17	8.14	5.44	5.57	3.67
FM	CIF	PSNR	33.41	33.17	33.07	32.58	32.98	32.84	33.03	33.04	33.13
		NSP	869.33	14.08	16.34	14.10	14.33	9.49	6.13	5.60	4.17
	QCIF	PSNR	32.55	32.40	32.31	32.05	32.19	32.16	32.17	32.21	32.23
		NSP	782.21	10.46	13.31	11.45	10.56	6.72	5.05	4.50	3.52
TT	SIF	PSNR	29.61	28.28	28.73	28.52	28.66	28.10	28.65	28.83	**29.28**
		NSP	859.45	12.58	15.47	13.28	13.75	8.49	5.73	5.50	4.42
	QCIF	PSNR	27.76	26.64	26.91	26.69	26.82	26.57	26.70	27.01	**27.50**
		NSP	782.21	10.88	13.89	12.56	11.82	7.44	6.71	4.97	4.52
FB	SIF	PSNR	22.98	21.99	22.17	22.12	22.07	21.76	22.32	22.42	**22.68**
		NSP	859.45	14.64	16.75	14.21	15.00	10.06	8.21	7.34	7.76
MC	SIF	PSNR	22.59	22.57	22.55	22.50	22.57	22.56	22.56	22.56	22.57
		NSP	859.45	10.71	12.57	11.96	10.13	7.13	4.98	4.66	3.66

SAD at $\mathrm{MV}_{\mathrm{median}}$ is smaller than T_1, the search stops. Otherwise, the SAD values of (0, 0), left, top, and top-right blocks are examined, and the point having the minimum SAD is considered $\mathrm{MV}_{\mathrm{INIT}}$. If the SAD at $\mathrm{MV}_{\mathrm{INIT}}$ is smaller than T_2, $\mathrm{MV}_{\mathrm{INIT}}$ is considered as the final MV, and the search stops. Otherwise, the distance (d_x, d_y) between the $\mathrm{MV}_{\mathrm{INIT}}$ and $\mathrm{MV}_{\mathrm{OPTIMAL}}$ is predicted using (6). If both d_x and d_y are smaller than $1/2$ pixel, the search stops. If only one of d_x and d_y is smaller than $1/2$ pixel, a search using VSP or HSP, given in Fig. 2(a) and (b), is performed. However, only if both d_x and d_y are larger than $1/2$ pixel, a search using SDSP is performed. According to the value of the SAD of $\mathrm{MV}_{\mathrm{SDSP}}$, the random MV search is selectively performed, as described in Section II-C. Note that the random MV search is not performed if any one of d_x and d_y is smaller than $1/2$ pixel. This is because the optimal MV may not be the random MV in this case. Note that any of d_x and d_y are smaller than $1/2$ pixel in many cases.

III. SIMULATION RESULTS

A. Experimental Environment

For the experiment, we use two video sequences of CIF format, *Carphone* (CP) and *Foreman* (FM), and three video sequences of SIF format, *Football* (FB), *Table tennis* (TT), and *Mobile and calendar* (MC). Also, two video sequences of QCIF format (176×144), *Foreman* and *Table tennis*, are additionally used. Each sequence consists of 100 frames with a frame rate of 30 Hz. In the simulation, the block size is set to 16×16 and the search range is to ± 15. Note that the number of search points for the proposed algorithm in the tables includes all computational burdens including the thresholding procedure and the random MV search. In the simulation, T_1 and T_2 for the proposed algorithm are set to 512 and 768, respectively.

B. Experimental Results

Table I illustrates the average PSNR and average number of search points per block for various ME algorithms. The thresholding values, T_1 and T_2, used in the proposed algorithm are a trade-off between the search speed and PSNR, and the results given in Table I are obtained using thresholding values of $T_1 = 512$ and $T_2 = 768$. Therefore, by adjusting the threshold values, the algorithm can achieve a higher speed-up ratio by sacrificing PSNR performance, or vice versa. Nevertheless, the results demonstrate that the proposed algorithm requires the smallest number of search points among BBGDS, HEXBS, CDS, DS, MVFAST, and PMVFAST algorithms and provides the best PSNR performance. Its average speed is improved by 20.24% with better PSNRs compared to PMVFAST. Hence, the proposed algorithm will provide even better PNSR performance than other algorithms if we adjust the thresholding values so as to provide the same search speed. Also, note that the algorithm significantly improves the PSNR performance for the sequences *Table tennis* and *Football*. This owes to the efficient random MV search in the proposed algorithm.

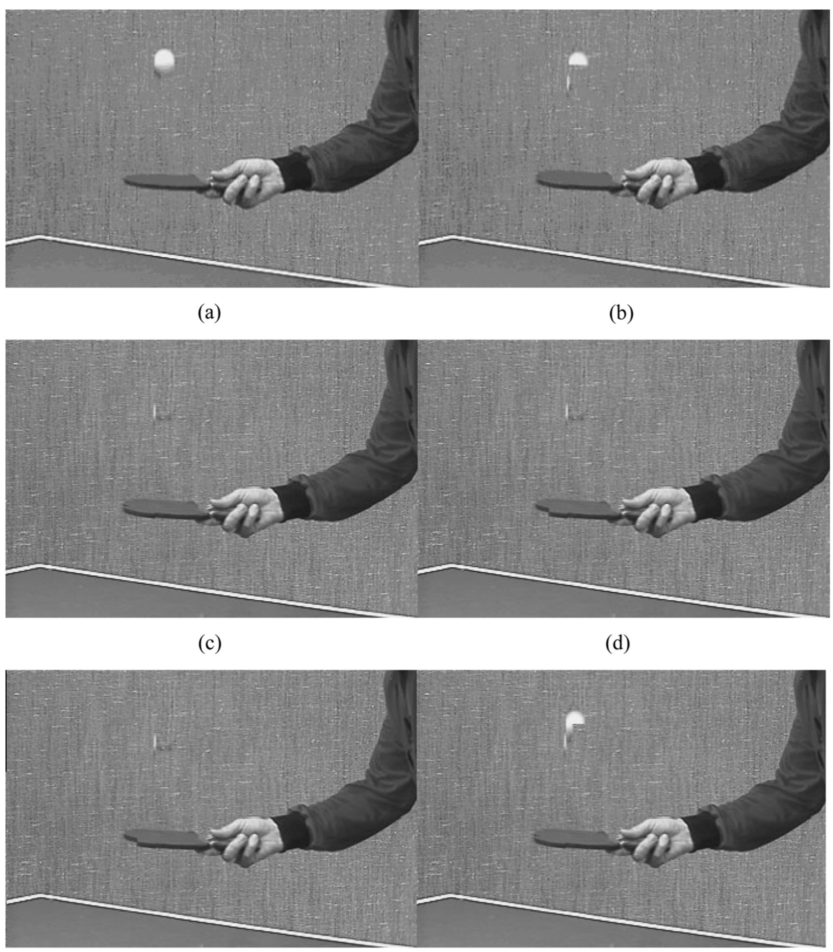
Computational burdens for thresholding in the proposed algorithm are given in Table II. In the table, we first examine the "Probability of calculating SADNP" because SADNP needs not to be calculated if SAD at $\mathrm{MV}_{\mathrm{INIT}}$ is less than T_2. Then, "Equivalent NSP" is obtained by multiplying the probability with the computational complexity for SADNP. Note here that the complexity is represented as the number of search points. As shown in the table, the additional computational load for SADNP is not significant compared with the total computational burden of ME. (Note that NSP in Table I includes this computational burden.) Table III illustrates the computational complexity for obtaining the initial search center in the random MV search. Here, NSP $_1$ denotes the number of search points in layer 1, and "Equivalent NSP" is obtained by multiplying the number of search points at layer 0 which corresponds to NSP_1,

TABLE II
COMPUTATIONAL COMPLEXITY REQUIRED FOR CALCULATING SADNP FOR THRESHOLDING

	CIF		SIF			QCIF	
	CP	FM	TT	FB	MC	FM	TT
Probability of calculating SADNP (%)	29.58	35.53	52.39	85.27	93.56	45.23	51.66
Equivalent NSP	0.14	0.18	0.26	0.43	0.47	0.23	0.26

TABLE III
COMPUTATIONAL COMPLEXITY REQUIRED FOR OBTAINING AN INITIAL SEARCH CENTER IN A RANDOM MV SEARCH

	CIF		SIF			QCIF	
	CP	FM	TT	FB	MC	FM	TT
Probability of performing a random MV search (%)	4.62	5.25	5.50	10.67	4.10	5.37	6.27
NSP_1	46.84	46.43	43.75	44.93	43.17	44.46	38.15
Equivalent NSP	0.14	0.15	0.15	0.30	0.11	0.15	0.15



(a) (b) (c) (d) (e) (f)

Fig. 7. The 14th image of the table tennis sequence reconstructed by using various motion estimation algorithms.

with "Probability of performing a random MV search." In Table III, it is noted that the probability of executing a random MV search is approximately 6% on average. However, this small portion of additional refinement significantly improves the PSNR performance, as shown in Table I.

Fig. 7 presents the 14th reconstructed image for various ME algorithms. The original image is given in (a), and (b)-(f) show the motion compensated images by using FSBMA, DS, MVFAST, PMVFAST, and the proposed algorithm, respectively. As shown in the figure, DS, MVFAST, and PMVFAST fail to find the ball within a search area while FSBMA and the proposed algorithm successfully find it. This demonstrates that the supplement scheme adopted in the proposed algorithm is working properly in terms of finding random MVs.

IV. Conclusion

In this paper, we introduce a newly defined measure to estimate the distance between the current search point and the optimal point in the middle of a search. If the real distance between them is smaller than 1 pixel, the predicted distance is close to the real one. If the real distance is larger than 1 pixel, it may not be correct but provides the information that the real one is larger than 1 pixel. By predicting fast or random MVs based on this estimated distance, we properly combine the existing GDS with the hierarchical search, which requires some computational burden but improves the ME performance for random MVs. We also propose a hybrid search scheme. The proposed scheme performs a hierarchical search only if a fast or random MV may exist, and fast or random MVs are infrequently found in video sequences. Therefore, the additional computational complexity due to the hierarchical search is not considerable. Also, the fairly well estimated distance improves the early termination performance, and thereby the overall search speed. Simulation results show that the proposed algorithm is faster than the existing GDS-based scheme while providing superior PSNR performance, especially for sequences having fast or random MVs.

References

[1] *Video Codec for Low Bitrate Communication*, ITU-T Recommendation H.263, International Telecommunication Union, Mar. 1996.

[2] *ISO/IEC CD 11172: Information Technology*, ISO/IEC JTC1/SC29/WG11, MPEG 1 Committee Draft, Dec. 1991.

[3] *ISO/IEC CD 13818: Information Technology*, ISO/IEC JTC1/SC29/WG11, MEPG 2 Committee Draft, Dec. 1993.

[4] *MPEG-4 Video Verification Model version 14.2*, ISO/IEC JTC1/SC29/WG11, MPEG99/5477, Dec. 1999.

[5] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799–1808, Dec. 1981.

[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video-conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Dec. 1981, pp. G5.3.1–G5.3.5.

[7] 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. 438–442, Aug. 1994.

[8] L. 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.

[9] 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.

[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. Zhu, X. Lin, L. Chau, and L. Po, "Enhanced hexagonal search for fast block motion estimation," *IEEE Trans. Circuits Syst Video Technol.*, vol. 14, no. 10, Oct. 2004.

[12] C. 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.

[13] P. I. Hosur and K. K. Ma, "Motion vector field adaptive fast motion estimation," in *Proc. 2nd Int. Conf. Information Communications and Signal Processing (ICICS '99)*, Singapore, Dec. 7–10, 1999.

[14] 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.

[15] K. R. Namuduri, "Motion estimation using spatio-temporal contextual information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 8, pp. 1111–1115, Aug. 2005.

[16] M. Bierling, "Displacement estimation by hierarchical block matching," *Proc. SPIE*, vol. 1001, pp. 942–951, 1988.

[17] K. M. Nam, J. S. Kim, R. H. Park, and Y. S. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 4, pp. 344–351, Aug. 1995.

[18] J. H. Lee, K. W. Lim, B. C. Song, and J. B. Ra, "A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1289–1301, Dec. 2001.