AN EXTENDED SMALL DIAMOND SEARCH ALGORITHM FOR FAST BLOCK MOTION ESTIMATION

Chang-Uk Jeong, Takeshi Ikenaga, and Satoshi Goto

Graduate School of Information, Production and Systems
Waseda University, Kitakyushu, 808-0135, Japan
Email: jcu@ruri.waseda.jp, {ikenaga, goto}@waseda.jp

ABSTRACT

In this paper, we propose a fast motion estimation algorithm that is suitable for searching both center-biased motion and large motion by applying the small diamond pattern used in some block-matching algorithms (BMA). The search will be terminated by a small diamond search (SDS) method after performing a large search based on a modified three-step search (3SS) strategy. The results of the experiment show an increase about 220% in the search speed compared to that of diamond search (DS) and efficient three-step search (E3SS) in a sequence that represent small motions in objects. Experimental results also demonstrate reasonable search points in the estimation of rough motions in high-resolution images, maintaining a performance better than other fast BMAs in terms of PSNR.

***Index Terms*—**Block-matching algorithm, fast search algorithm, motion estimation, small diamond search, video compression

1. INTRODUCTION

Motion estimation (ME) method is typically used to effectively remove the temporal redundancy. In addition, the block-matching motion estimation (BMME) is the most significant part in today's video coding techniques and standards. The BMME has also been adopted by various international video compression standards, such as ISO/IEC MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, H.263 [1], and recently published H.264 [2]. The block-matching algorithm (BMA) has been widely applied due to the efficiency of prediction and accuracy of estimation.

In the last few years, various fast BMAs were proposed to replace the full search (FS) algorithm with high computational complexity. The proposed BMAs use different search strategies to simultaneously satisfy the accuracy and search speed in motion estimation. In particular, the size and shape of search patterns largely affect the efficiency of algorithms. A typical algorithm of fast BMAs is the three-step search (3SS) [3]. The 3SS shows a logarithmical decrease in the size of rectangular search patterns in each search step. Even though the 3SS algorithm performs well in large search, there exists severe redundancy in calculation because it always uses a constant number of search points (SPT). Also, it overlooks large part of the center-biased motion vector (MV) distribution of real-world image sequences [4]. Some later published algorithms consider the estimation of center-biased motions and thus are more appropriate for searching real-world sequences. They are new three-step search (N3SS) [4], four-step search (4SS) [5], and block-based gradient descent search (BBGDS) [6] algorithms. In addition, most of search patterns are restricted as a rectangular shape before the diamond search (DS) [7]–[9] and the hexagon-based search (HEXBS) [10] algorithms are proposed. The DS and HEXBS can obtain higher prediction quality even though they use fewer search points. They perform the search by applying unrestricted search steps under the diamond and hexagon-shaped search patterns. However, a weakness in local minimum block distortion measure (BDM) points deteriorates the performance of large search because these algorithms depend on size and shape restricted patterns. Nevertheless, new methods such as the point-oriented grouping strategy [11] continue to be proposed for speedup them. The efficient three-step search (E3SS) algorithm [12] proposed in recent years uses the three-step search strategy employed in the 3SS algorithm to perform large search and can be corresponded to the center of searching area by performing a small diamond search (SDS) process based on the unrestricted search step method. Meanwhile, the unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [13] algorithm produces an excellent search quality, but is much slower than other fast BMAs.

In this paper, we propose an extended small diamond search (ESDS) algorithm by employing the small diamond pattern used in the early searching step in the cross-diamond-hexagonal search (CDHS) algorithm [14] as a type of minimum-size patterns to find the zero motion vector (ZMV). To perform large search, the ESDS is based on the phased reduction of the large rectangular pattern and new modified search strategy of the 3SS.

2. EXTENDED SMALL DIAMOND SEARCH

The proposed ESDS algorithm arranges a 3×3 small diamond-shaped pattern (SDSP) at the center of a search window in the early searching step. The SDSP is selected as a minimum-size pattern to find the ZMV. The N3SS, DS, and E3SS require the minimum number of SPTs as 17, 13, and 13, respectively, to find the ZMV, whereas the ESDS requires only 5 SPTs. If the minimum BDM point is determined as the center of the SDSP, the search is terminated. The second searching step is implemented when the minimum BDM point is determined as the contour of the SDSP.

The second step in search uses a rectangular pattern that is a half of the search window. In a 15×15 small search window (maximum displacement $W=\pm7$), a 9×9 rectangle that is a half of the search window becomes the search pattern. As shown in Fig. 1, 8 SPTs marked at the outside of the SDPS using dots are the elements of the large rectangle-shaped pattern (LRSP) arranged at the center of a search window. If the minimum BDM point on the SDSP used in the early searching step becomes the minimum BDM point in the second searching step again, it skips the third and fourth searching step and performs the SDS algorithm of the final step in order to search center-biased motions. The second searching step is critical in the decision that selects either large search or central search.

In the third step in search, the LRSP used for the large search in

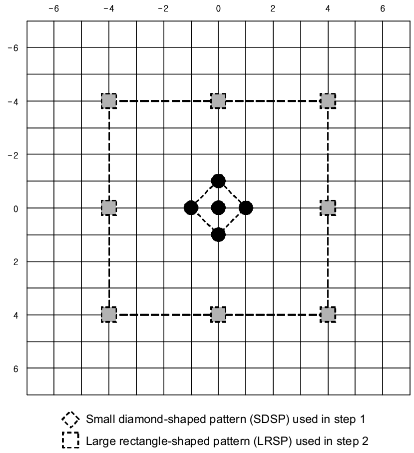


Fig. 1. The SDSP used in the first step and the LRSP used in the second step of the ESDS. The ESDS uses only 5 SPTs to find the ZMV. If the SPTs are not recognized as the ZMV, it adds 8 SPTs on the LRSP in the next step.

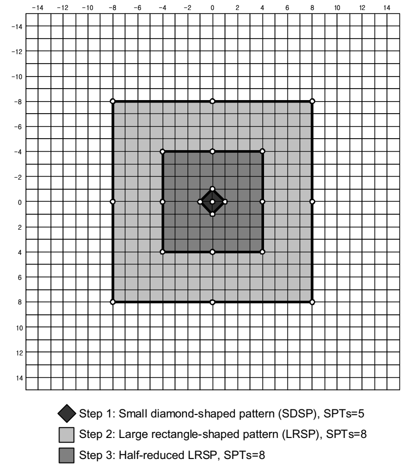


Fig. 2. The SDSP and LRSPs used up to the third step of the ESDS for a search window, W=±15.

a search window, which represents more than W=±15 (31×31), shows a logarithmical decrease in its size up to 9×9. Thus, in a small search window (W=±7), it skips this process and then comes to the fourth step. The LRSP reduced by half are arranged at the center of a search window as pattern center in which each reduced pattern shows a constant number of SPTs as 8. Because the distance between the SDSP used in the early searching step and the LRSP used in the second searching step becomes more distant in proportion as the size of a search window increases, it is important to consider the third search process as a countermeasure against the existence of a global minimum BDM point in the area between two patterns. The performance of the third searching step not only enhances the internal search of the LRSP, but also modifies the search direction. Fig. 2 shows the search points on the SDSP and LRSPs used up to the third searching step for W=±15.

The proposed algorithm implements the fourth step in search that applies a large search strategy employed in the 3SS after completing the internal reducing process of rectangular patterns. First, it is necessary to reduce the size of the LRSP, which corresponds to the minimum BDM point in the previous step, to half of the original size. Because the point always exists on the 9×9 rectangular pattern for W=±7, the size becomes 5×5. In the case of W=±15, because the size determined in the second search process is reduced from 15×15 to 9×9, the LRSP will be reduced by 9×9 if one of the search points on the 15×15 LRSP is the minimum BDM point. In addition, if the minimum BDM point in the previous step is a point of the search points on the 9×9 LRSP, the LRSP will be reduced by 5×5. Then, the half-reduced LRSP of the objective pattern is arranged with the minimum BDM point in the previous step as the center and is able to be searched and compared. In addition, this process is repeatedly performed until the size of the LRSP is reduced to 5×5, different from that of the 3SS. Fig. 3 illustrates certain search patterns and SPTs in the fourth searching step for the search window of W=±15. The number of SPTs on the reduced and arranged LRSP can be determined as 5, 7, and 8 according to the location of the minimum BDM point in the previous step as illustrated in Fig. 3.

In the final step of the ESDS, it performs the SDS algorithm (Fig. 4). Because the SDS is performed as a manner of the unrestricted search step, the number of SPTs is always changing. In addition, it

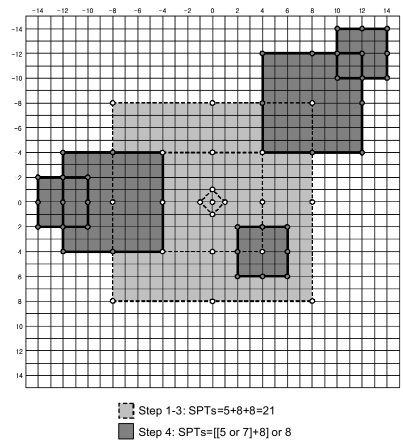


Fig. 3. Examples of the search patterns and search points used in the fourth step for W=±15.

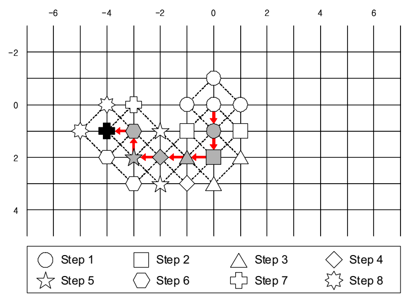


Fig. 4. An example of the MV (-4, 1) search path in the SDS algorithm.

uses a few number of search points and is simple in implementation. Fig. 5 illustrates examples in the search using the ESDS algorithm for the search window size W=±15. Each search step in the ESDS can be summarized as follows:

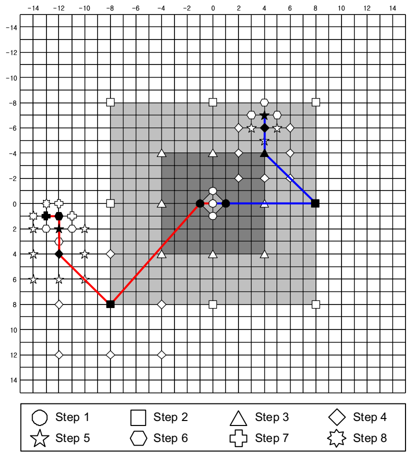


Fig. 5. Two examples of search paths in the ESDS for W=±15.

Step (1): Search of SPTs on the SDSP

5 searching points on the SDSP are arranged at the center of a search window. If a minimum BDM point is determined as the center of the search window, the search will be terminated (MV=ZMV). Otherwise, it implements Step (2).

Step (2): Search of SPTs on the LRSP

The LRSP here is a half of the search window. The center point of the LRSP is set at the center of the search window. 8 SPTs on the pattern are compared with the minimum BDM point in the previous step. If a SPT in Step (1) is determined as the minimum BDM point, the process moves to Step (5). Otherwise, it implements the next step, Step (3).

Step (3): Reduction of the LRSP

It reduces the LRSP to half size. If the reduced size of the LRSP is smaller than 9×9, the process moves to Step (4). Otherwise, it arranges the reduced LRSP at the center of the search window and compares 8 SPTs on the pattern with the minimum BDM point in the previous step. Then, it repeats Step (3).

Step (4): Large search of the LRSP

It reduces the size of the LRSP that corresponds to the minimum BDM point in the previous step as its half size. If the reduced size is determined to be smaller than 5×5, the process moves to Step (5). Otherwise, the reduced LRSP is adopted with the minimum BDM point as its center, and SPTs are searched. Then, it repeats Step (4).

Step (5): Implementation of the SDS algorithm

It moves the center of the SDSP to the minimum BDM point in the previous step. Then, it searches the other SPTs on the SDSP. If the minimum BDM point is the center of the SDSP, it terminates the search. Otherwise, it repeats Step (5).

3. EXPERIMENTAL RESULTS

The search speed can be verified by the average number of search points per block (Avg. SPT) and the speed up ratio with respect to the FS (Speedup). The sum of absolute difference (SAD) algorithm is used to measure the BDM. The basic block size used in simulations is 16×16 and search window sizes are configured as W=±7, W=±15, and W=±31, respectively. The average peak signal-to-noise ratio (PSNR) per frame shows the search quality. Test sequences used in this experiment are “Claire” (CIF 360×288, 100 frames), “Football” (CIF 360×240, 200 frames), “Coastguard” (CIF 352×288, 200 frames), “Stefan” (CCIR601 720×480, 200 frames), “Susie” (CCIR601 720×486, 90 frames), and “Garden” (CCIR601 720×486, 90 frames). Tables 1–6 compare the performance between the proposed ESDS algorithm and other BMAs, such as the FS, 3SS, N3SS, DS, extended hexagon-based search (EHS) [13], and E3SS. From Table 1, in the “Claire” sequence that shows a tendency of quasi-stationary motions, it is evident that the ESDS shows an increase in the search speed more than 220% by searching about 6 SPTs, which are 7 SPTs fewer than that of the DS and E3SS. However, the prediction accuracy can be maintained as an equivalent level for other algorithms. In the case of the CIF sequences of “Football” and “Coastguard”, which include large-scale motions, it demonstrates the highest PSNR and a good search speed as shown in Tables 2 and 3. The performance on the “Stefan” sequence including tough and rough sports scene with the resolution of 720×480 is shown in Table 4. The proposed ESDS shows a remarkable performance in the search quality for the “Stefan” sequence compared to that of the N3SS, DS, and EHS. Also, it shows that 2–3 SPTs are determined as a gain even though it shows a similar aspect of the PSNR compared to that of the E3SS. In addition, the proposed algorithm achieves the highest PSNR compared to other fast algorithms for the “Susie” and “Garden” sequences as shown in Tables 5 and 6. Fig. 6 illustrates the average PSNR per frame and the average number of search points per block for CCIR601 “Stefan” sequence with W=±31. The figure shows that the proposed algorithm demonstrates faster search speed than other fast BMAs except for the DS and EHS in most frames. We can also see the proposed ESDS maintains an outstanding search quality compared to other fast BMAs.

4. CONCLUSIONS

This paper proposes an extended small diamond search (ESDS) algorithm using the SDSP and the LRSP. The proposed ESDS algorithm logarithmically decreases the size of the LRSP in a large search window and arranges the reduced LRSP at the center. Then, it uses a modified large search of the 3SS. In the last search step, it finishes the search using the SDS algorithm. The results of the experiment show that the proposed algorithm represents better search quality compared with other fast BMAs while it shows more computing gains than that of the DS and E3SS. Moreover, the proposed algorithm provides stronger flexibility than other BMAs because it demonstrates satisfying performances in both quasi-stationary and rough motions.

ACKNOWLEDGMENT

This research was supported by CREST, JST.

REFERENCES

[1] K. R. Rao and J. J Hwang, *Techniques and Standards for Image, Video and Audio Coding*. Englewood Cliffs, NJ: Prentice Hall, 1996.

[2] “Draft ITU-T Rec. and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264-ISO/IEC 14 496-10 AVC),” Joint Video Team (JVT) of ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1, 2003.

[3] 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, Nov. 29–Dec. 3 1981, pp. G5.3.1–G5.3.5.

[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, pp. 438–442, 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, pp. 313–317, June. 1996.

Table 1. Performance comparisons for CIF "Claire"

BMA	Max. Displacement W=±7			Max. Displacement W=±15		
	Avg. SPT	Speedup	PSNR	Avg. SPT	Speedup	PSNR
FS	225.000	1.000	41.184	961.000	1.000	41.186
3SS	25.000	9.000	41.092	33.000	29.121	41.061
N3SS	17.405	12.927	41.167	17.396	55.243	41.129
DS	13.225	17.013	41.161	13.226	72.660	41.162
EHS	11.330	19.859	41.093	11.330	84.819	41.094
E3SS	13.425	16.760	41.137	13.422	71.599	41.136
ESDS	6.030	37.313	41.133	6.047	158.922	41.134

Table 2. Performance comparisons for CIF "Football"

BMA	Max. Displacement W=±7			Max. Displacement W=±15		
	Avg. SPT	Speedup	PSNR	Avg. SPT	Speedup	PSNR
FS	225.000	1.000	23.922	961.000	1.000	25.333
3SS	25.000	9.000	23.608	33.000	29.121	24.337
N3SS	25.552	8.806	23.662	28.229	34.043	24.263
DS	21.793	10.324	23.506	23.898	40.213	24.241
EHS	17.869	12.592	23.479	19.080	50.367	24.200
E3SS	22.150	10.158	23.673	25.476	37.722	24.498
ESDS	18.703	12.030	23.624	23.561	40.788	24.607

Table 3. Performance comparisons for CIF "Coastguard"

BMA	Max. Displacement W=±7			Max. Displacement W=±15		
	Avg. SPT	Speedup	PSNR	Avg. SPT	Speedup	PSNR
FS	225.000	1.000	29.687	961.000	1.000	29.811
3SS	25.000	9.000	29.366	33.000	29.121	28.953
N3SS	21.937	10.257	29.529	20.640	46.560	28.950
DS	17.648	12.749	29.425	17.692	54.318	29.483
EHS	14.832	15.170	29.387	14.838	64.766	29.436
E3SS	19.228	11.702	29.533	18.764	51.215	29.535
ESDS	17.999	12.501	29.508	17.907	53.666	29.618

Table 4. Performance comparisons for CCIR601 "Stefan"

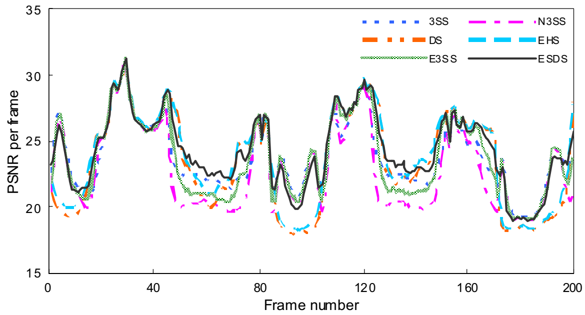
BMA	Max. Displacement W=±15			Max. Displacement W=±31		
	Avg. SPT	Speedup	PSNR	Avg. SPT	Speedup	PSNR
FS	961.000	1.000	24.434	3969.000	1.000	25.998
3SS	33.000	29.121	23.277	41.000	96.805	23.057
N3SS	27.100	35.461	22.900	27.199	145.924	21.930
DS	21.504	44.689	21.944	21.702	182.886	21.981
EHS	17.683	54.346	22.249	17.903	221.695	22.360
E3SS	24.460	39.289	23.289	25.773	153.998	22.776
ESDS	21.051	45.651	23.122	23.411	169.536	23.411

Table 5. Performance comparisons for CCIR601 "Susie"

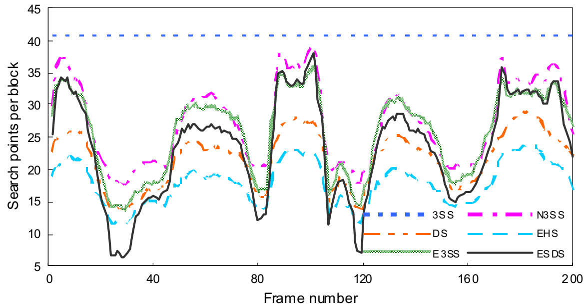
BMA	Max. Displacement W=±15			Max. Displacement W=±31		
	Avg. SPT	Speedup	PSNR	Avg. SPT	Speedup	PSNR
FS	961.000	1.000	35.837	3969.000	1.000	36.278
3SS	33.000	29.121	34.502	41.000	96.805	34.115
N3SS	23.706	40.538	34.754	23.771	166.968	34.307
DS	20.464	46.961	34.659	20.626	192.427	34.842
EHS	16.590	57.926	34.886	16.718	237.409	35.156
E3SS	20.822	46.153	34.788	21.237	186.891	34.585
ESDS	19.529	49.209	34.936	20.402	194.540	35.047

Table 6. Performance comparisons for CCIR601 "Garden"

BMA	Max. Displacement W=±15			Max. Displacement W=±31		
	Avg. SPT	Speedup	PSNR	Avg. SPT	Speedup	PSNR
FS	961.000	1.000	28.171	3969.000	1.000	28.262
3SS	33.000	29.121	25.250	41.000	96.805	22.188
N3SS	27.486	34.963	24.486	26.463	149.983	21.268
DS	22.329	43.038	26.252	22.339	177.671	26.255
EHS	17.606	54.584	26.155	17.612	225.358	26.158
E3SS	26.098	36.823	26.983	26.739	148.435	24.059
ESDS	25.907	37.094	27.532	26.898	147.557	26.887



(a)



(b)

Fig. 6. Frame by frame performance comparisons for CCIR601 "Stefan" sequence with W=±31. (a) The average PSNR per frame. (b) The average number of search points per block.

[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, pp. 419–423, Aug. 1996.

[7] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," in *Proc. 1997 Int. Conf. Information, Communications and Signal Processing (ICICS)*, vol. 1, Sep. 9–12, 1997, pp. 292–296.

[8] 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.

[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, pp. 349–355, May. 2002.

[11] L. M. Po, C. W. Ting, K. M. Wong, and K. H. Ng, "Novel point-oriented inner searches for fast block motion estimation," *IEEE Trans. Multimedia.*, vol. 9, no. 1, pp. 9–15, Jan. 2007.

[12] 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.

[13] 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.

[14] 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, Feb. 2005.