An Efficient Three-Step Search Algorithm for Block Motion Estimation

Xuan Jing and Lap-Pui Chau, *Senior Member, IEEE*

***Abstract*—The three-step search algorithm has been widely used in block matching motion estimation due to its simplicity and effectiveness. The sparsely distributed checking points pattern in the first step is very suitable for searching large motion. However, for stationary or quasistationary blocks it will easily lead the search to be trapped into a local minimum. In this paper we propose a modification on the three-step search algorithm which employs a small diamond pattern in the first step, and the unrestricted search step is used to search the center area. Experimental results show that the new efficient three-step search performs better than new three-step search in terms of MSE and requires less computation by up to 15% on average.**

***Index Terms*—Block matching, motion estimation, video coding.**

I. Introduction

Motion estimation plays an important role in interframe predictive coding system. When the frame rate is sufficiently high, there is a great amount of similarity between successive frames. It is more efficient to code the difference between frames than the frames themselves. Motion estimation is aimed to reduce such strong temporal redundancy present in video sequence. So far, the most successful technique for motion estimation is perhaps the block-matching algorithm (BMA) which has been adopted by various video coding standards, such as ITU-T H.261, H.263, MPEG-1, MPEG-2, and MPEG-4. The full search (FS) block-matching algorithm is the simplest BMA which provides an optimal solution by exhaustively search all candidates within the search window. But its high computational complexity made it difficult to be implemented in real-time software-based applications. In order to reduce the computational complexity of FS, many fast block-matching algorithms have been developed such as 2D-logarithm search (LOGS) [1], three-step search (3SS) [2], new three-step search (N3SS) [3], four-step search (4SS) [4], block-based gradient descent search (BBGDS) [5], and diamond search (DS) [6], [7], etc. Among these fast BMAs, the three-step search (3SS) algorithm becomes the most popular because of its simplicity and effectiveness.Searching with a large pattern in the first step, 3SS is more efficient to find the globe minimum especially for those sequences with large motion. On the other hand, it becomes inefficient for the estimation of small motions since it will be trapped into a local minimum.

Manuscript received April 5, 2002; revised October 25, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ryoichi Komiya.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: PG01772596@ntu.edu.sg; elpchau@ntu.edu.sg).

Digital Object Identifier 10.1109/TMM.2004.827517

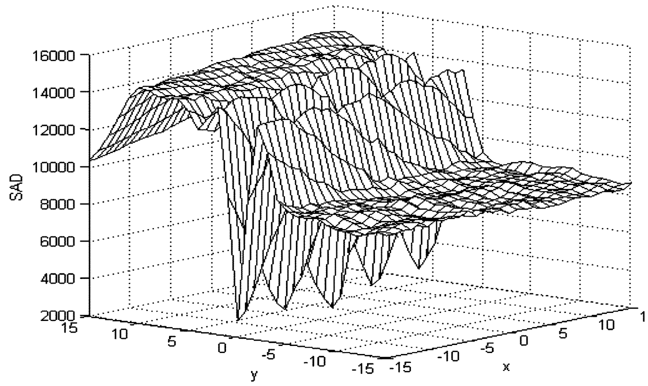


Fig. 1. Multiple local minima within a search window.

Actually, in real-world video sequences, the distribution of motion vectors is highly center-biased. In order to favor this characteristic, the N3SS modified the checking points pattern in the first step by searching additional central eight points. At the mean time, N3SS used a halfway-stop technique to speed up the (quasi-) stationary blocks matching. Experimental results show that the N3SS performs better than 3SS in terms of mean square error and the number of search points. Another BMA using center-biased search pattern is 4SS which utilizing a smaller 5×5 grid in the first step. As a result, 4SS requires four steps to reach the boundary checking points for the search window $w = 7$. The performance of 4SS is better than the 3SS and comparable to the N3SS with fewer search points. In fact, all of the fast BMAs are based on the assumption [1] that the block distortion measure (BDM) increases monotonically as the checking point moves away from the global minimum BDM point. However, this is not always true in the real world especially for large motion blocks where multiple local minima may exist in the search window. In Fig. 1, we depict an error surface of a block which contains multiple local minima within the search window. When a large search window is used e.g., $w = 15$, the initial step size for the above three algorithms will increase by a factor of 2. Therefore, the large 17×17 grid and 9×9 grid with the nine sparsely distributed checking points in the first step of 3SS and 4SS are more likely to mislead the search to a wrong direction. In addition, the halfway-stop technique in N3SS will lose its accuracy as well for it may miss the chance to find the globe minimum surrounding the central 5×5 area.

In this paper, we propose an efficient three-step search (E3SS) algorithm for fast block motion estimation. The E3SS employs a center-biased small diamond pattern in the first step and the unrestricted search step is used to search the center area. The

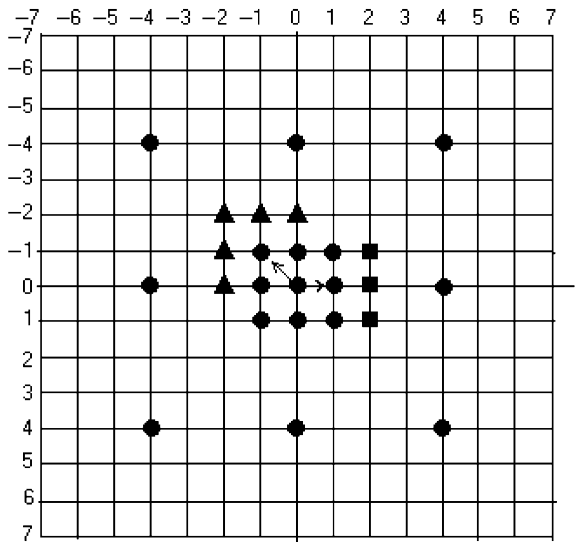


Fig. 2. Two different search paths for finding MV within 5 × 5 area in N3SS.

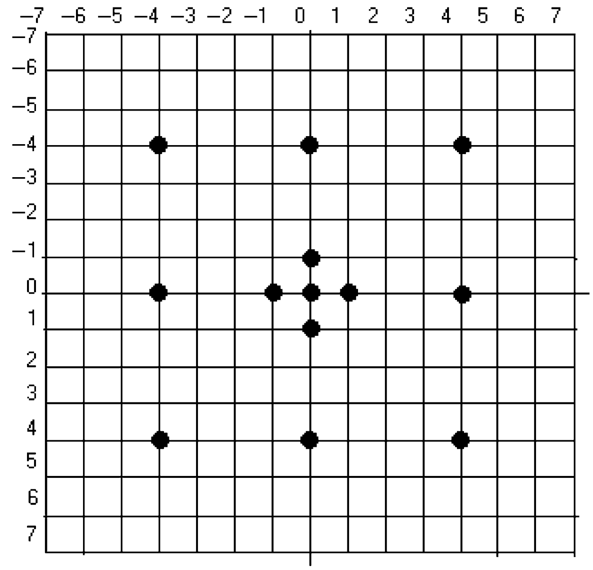


Fig. 3. Search pattern used in the first step of E3SS.

rest of the paper is organized as follows. Section II first reviews the N3SS algorithm. Then our E3SS is presented in Section III. Section IV presents some simulation results of our proposed method in comparison of the FS, 3SS, 4SS, N3SS, and DS. Finally, the conclusions are given in Section V.

II. New Three-Step Search Algorithm

For the search window $w = 7$, in the first step the N3SS algorithm utilizes a 3×3 grid search pattern at the center in addition to the larger 9×9 grid in the 3SS. If the minimum BDM point is found at the center of the search window, the search will stop. If the minimum BDM point is one of the eight points on the 3×3 grid, only additional five or three points will be checked, otherwise the search window center will be moved to the winning point and the following procedure is the same as in 3SS. The details are as follows.

Step 1) Totally $8 + 9 = 17$ points are checked including the central nine points on the 3×3 grid and the eight neighboring points on the 9×9 grid. If the minimum BDM point is the search window center, the search will be terminated; otherwise go to Step 2.

Step 2) If one of the central eight neighboring points on the 3×3 grid is found to be the minimum in the first step, go to Step 3; otherwise go to Step 4.

Step 3) Move the small 3×3 search window so that the window center is the winning point found in Step 1. Search additional five or three points according to the location of the previous winning point, then the search will stop.

Step 4) Reduce the large 9×9 search window size by half and move the center to the minimum BDM point in Step 1, follow the searching process of Step 2 and Step 3 in 3SS (refer to [2]).

Fig. 2 shows two different search paths for finding motion vector within 5×5 area. According to the halfway-stop technique, the N3SS needs 17 search points for stationary blocks and 20 or 22 points for small motion vectors within central 5×5 area. For the worst case, $25 + 8 = 33$ search points will be required compared to 25 points needed in 3SS.

III. An Efficient Three-Stap Search Algorithm

In order to exploit the center-biased characteristics of motion vector distribution in real-world video sequences, a small diamond search pattern is utilized in the search window center. Fig. 3 shows the search pattern used in the first step of our E3SS, we take the search window $w = 7$ as an example. Thus, in the first step, totally $9 + 4 = 13$ points will be checked which is four points more than 3SS and four points fewer than N3SS. If the winning point is the search window center the search will be stopped. If the minimum BDM point is one of the eight neighboring points on the large 9×9 grid, the following process will be the same as in 3SS. If the minimum is one of the four points on the small diamond, the small diamond center is set to the minimum point and another three points will be checked.The center of the small diamond is moved to the winning point each time until the minimum is found in the small diamond center. The E3SS differs from N3SS in that 1) a small diamond pattern is used instead of a square pattern in the central area and 2) unrestricted search step for the small diamond rather than a single movement for the small square. The E3SS can be summarized as follows: (taking search window $w = 7$ as an example).

Step 1) In addition to the nine checking points in 3SS, four points on a small diamond in the center are added. If the minimum BDM point is in the search window center, the search will be stopped. Otherwise, go to Step 2.

Step 2) If the minimum point in Step 1 is one of the eight neighboring points on the 9×9 grid, the rest search procedure is the same as in 3SS. Otherwise, go to Step 3.

Step 3) Move the small diamond so that its center is the previous winning point. Continue to search the other points on the small diamond.There is no restriction

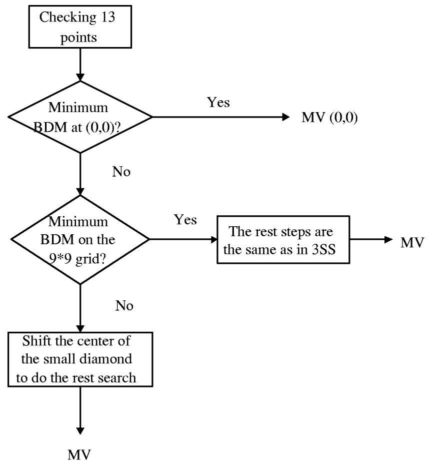


Fig. 4. Block diagram of the E3SS algorithm.

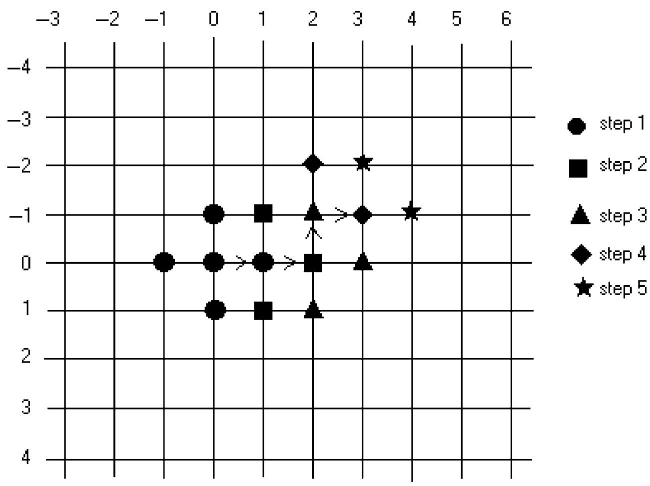


Fig. 5. Search path for finding MV $(3, -1)$.

on the search step unless the minimum is the small diamond center or the small diamond reaches the boundary of search window. More importantly, only three or two extra points are checked each time.

Fig. 4 shows the block diagram of the E3SS algorithm. In Fig. 5, we illustrate a search path for finding motion vector $(3, -1)$ when the winning point in the first step is (1, 0). It is obvious that when the minimum BDM point is found in one of the neighboring points around center (0, 0) in the first step, utilizing unrestricted search step can increase the probability to find the true motion vector which located near the central 5×5 area. In general, the E3SS needs 13 points for stationary blocks and 16 to 21 points for small motion vectors within central 5×5 area compared with 17, 20 or 22 points needed in N3SS respectively. For the worst case, $9+4+8+8 = 29$ points are required while 25, 33, and 27 points are needed in 3SS, N3SS, and 4SS, respectively.

IV. Simulation Results

The experiments are conducted by using the luminance component of five video sequences "Coastguard" (CIF 352×288, 100 frames) "Football" (CCIR601 720×486, 59 frames), "Susie"(CIF 352×288, 100 frames), "Mother & Daughter" (CIF 352×288, 100 frames), and "Salesman" (CIF 352×288, 100 frames). The first two sequences contain medium to large motion activity while the others consist of relatively small motion. The distortion measurement of mean absolute difference (MAD) is used as the BDM. The search window $\mathrm{w} = 7$ and $\mathrm{w} = 15$ are both used for a block size of 16×16.

Three criteria are used for comparing our E3SS against five other BMAs—FS, 3SS, 4SS, N3SS, and DS. They are

TABLE I
Performance Comparisons for CIF Sequence "Coastguard"

BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	79.87	225.00	1.00	76.07	961.00	1.00
3SS	89.33	25.00	9.00	95.69	33.00	29.12
4SS	90.83	19.81	11.36	89.05	28.25	34.02
N3SS	85.15	21.56	10.44	99.17	21.38	44.95
DS	88.45	17.80	12.64	86.48	17.89	53.72
E3SS	85.06	18.63	12.08	84.66	18.88	50.90

TABLE II
Performance Comparisons for CCIR601 Sequence "Football"

BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	257.83	225.00	1.00	145.06	961.00	1.00
3SS	276.22	25.00	9.00	186.04	33.00	29.12
4SS	289.72	20.56	10.94	193.54	29.80	32.25
N3SS	276.70	23.81	9.45	192.82	25.68	37.42
DS	289.30	20.72	10.86	217.97	22.75	42.24
E3SS	275.59	20.29	11.09	182.47	22.63	42.47

TABLE III
Performance Comparisons for CIF Sequence "Susie"

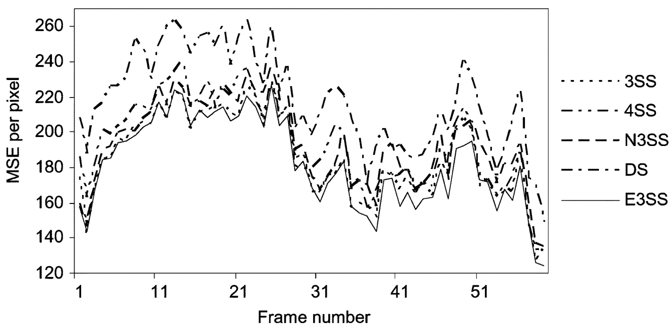
BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	11.89	225.00	1.00	11.82	961.00	1.00
3SS	12.52	25.00	9.00	12.71	33.00	29.12
4SS	12.29	17.37	12.95	12.37	25.51	37.67
N3SS	11.93	17.96	12.53	12.00	17.92	53.63
DS	12.05	13.65	16.48	12.03	13.65	70.40
E3SS	12.07	13.95	16.13	12.08	13.91	69.09

TABLE IV
Performance Comparisons for CIF Sequence "Mother & Daughter"

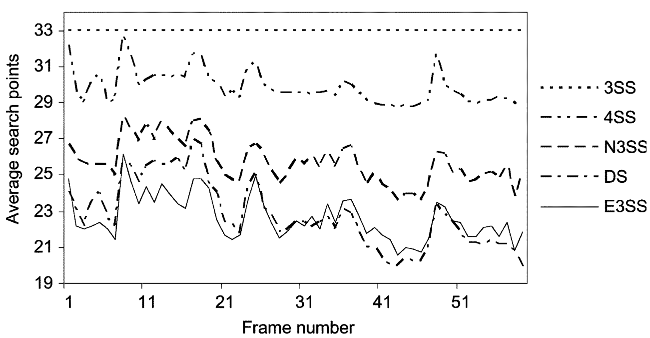
BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	17.63	225.00	1.00	16.02	961.00	1.00
3SS	18.45	25.00	9.00	17.70	33.00	29.12
4SS	18.31	17.97	12.52	17.24	26.29	36.55
N3SS	17.88	19.36	11.62	16.97	19.43	49.46
DS	18.04	14.45	15.57	16.99	14.51	66.23
E3SS	18.08	15.60	14.42	16.94	15.79	60.86

TABLE V
Performance Comparisons for CIF Sequence "Salesman"

BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	18.82	225.00	1.00	18.79	961.00	1.00
3SS	19.70	25.00	9.00	19.96	33.00	29.12
4SS	19.45	17.40	12.93	19.69	25.49	37.70
N3SS	18.94	18.19	12.37	19.25	18.15	52.95
DS	19.41	13.85	16.25	19.41	13.85	69.39
E3SS	18.95	14.23	15.81	19.00	14.23	67.53



(a)



(b)

Fig. 6. Frame by frame comparisons for "Football" sequence (720×486, 59 frames with search window $w = 15$. (a) MSE per frame. (b) Average search points per block.

mean square error (MSE) between the estimated frames and the original frame, the average number of search points per block (SearchPT), and the speed up ratio with respect to FS (SpeedUP).The experimental results for comparison are summarized in Tables I–V. It can be seen from Tables I and II that for sequences which contain medium to large motion and regardless of search window size the proposed E3SS gives the smallest MSE among the five fast algorithms while its computational complexity is less than 3SS, 4SS, N3SS, and comparable to DS. Tables III–V show that the E3SS is also suitable for searching small motion vectors. It achieves similar or smaller MSE than that of N3SS and DS while it needs one more search point compared with DS. On average the E3SS requires three to four points fewer than the N3SS thus a 15% speed up ratio against N3SS can be achieved. In addition, we can find that when using large search window, the search points needed for 3SS and 4SS increased dramatically due to the increasing search step while the computational requirement for E3SS, DS, and N3SS only changed slightly. Thus, the E3SS is more robust than the other fast BMAs. Fig. 6(a) and (b) illustrate the frame by frame comparison of E3SS, 3SS, 4SS, N3SS, and DS in terms of MSE and the average number of search points per block respectively for the "Football" sequence using search window $w = 15$.

V. Conclusion

In this paper, an efficient three-step search algorithm (E3SS) for fast block matching is proposed. By employing a small diamond checking pattern in the first step, the unrestricted search step is used to search the center-biased motion vectors. Experimental results show that for both large and small search window the proposed E3SS performs better than the 3SS, 4SS, N3SS, and DS in terms of MSE with fewer or comparable number of search points for the sequences which contain medium to large motion. It also has similar performance compared with N3SS and DS when searching small motion vectors. Hence, the E3SS is suitable for a wide range of video applications.

References

[1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1806, Dec. 1981.

[2] 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.

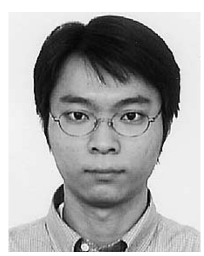
[3] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.

[4] 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.

[5] 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.

[6] 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.

[7] 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.



Xuan Jing received the B.E. degree from the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2001. He is currently pursuing the Ph.D. degree in the School of electrical and electronic engineering, Nanyang Technological University, Singapore.

His research interests include digital video compression and coding, signal processing and multimedia applications, etc.

Lap-Pui Chau (M'96–SM'03) received the B.Eng. degree with first class honor in electronic engineering from Oxford Brookes University, U.K., and the Ph.D. degree in electronic engineering from Hong Kong Polytechnic University in 1992 and 1997, respectively.

In 1996, he joined Tritech Microelectronics as a Senior Engineer. During 1997–2000, he was a Researcher at the Center for Signal Processing, Singapore. Since 2000, he has been an assistant professor with Nanyang Technological University, Singapore and is currently an Associate Professor. His research interests include streaming multimedia, image and video processing, and VLSI for signal processing.

Dr. Chau is involved in the organizational and technical committees with several international conferences. He is a member of the Technical Committee on Visual Signal Processing and Communications of the IEEE Circuits and Systems Society.