

A Fast Motion Estimation Algorithm Based on Diamond and Simplified Square Search Patterns

Yun Cheng^{1,2}, Kui Dai², Zhiying Wang², and Jianjun Guo²

¹Department of Information Engineering, Hunan Institute of Humanities,
Science and Technology, 417000 Loudi, China
{chengyun, daikui}@chiplight.com.cn

²College of Computer, National University of Defense Technology,
410073 Changsha, China

Abstract. Based on the directional characteristic of SAD(Sum of Absolute Difference) distribution and the center-biased characteristic of motion vectors, a fast BMA(block-matching motion estimation algorithm), DSSS(Diamond and Simplified Square Search), is proposed in this paper. DSSS employs line search pattern(LP), triangle search pattern(TP), or square pattern(SP) adaptively according to the distance between the MBD(Minimum Block Distortion) and SMBD(Second MBD) points to locate the best matching block with large motion vector, and diamond search pattern(DP) to refine the motion vector. Although the proposed DSSS may also be trapped in local minima, the experimental results show that it is faster than DS(Diamond Search) and DTS(Diamond and Triangle Search), while its encoding efficiency is better than DS and it is almost the same as that of DTS.

1 Introduction

Motion Compensated Predictive Coding can improve the encoding efficiency greatly by eliminating the temporal redundancy between successive frames and it was adopted by many video coding standards such as MPEG-1/2/4, H.261, H.263 and H.264/AVC[1], etc. The basic algorithm for motion compensated predictive coding is the block-matching motion estimation(BMME), and the most basic BMA(BMME Algorithm) is the full search (FS). Although FS can find out the best matching block by exhaustively testing all the candidate blocks within the search window, its computation is too heavy, for example, experimental results show that the time of the BMME consumed by FS in H.264/AVC is about 80% of the total. In order to speed up the BMME in the process of video encoding, many researchers have been working hard and have proposed many kinds of fast BMAs.

Most of the fast BMAs find the best matching block (or point) by using some special search patters. For example, TDLs(Two-Dimensional Logarithmic Search) uses “+” search pattern[2]; CSA (Cross-Search Algorithm)[3] and DSWA(Dynamic Search-Window Adjustment) [4] adopt “X” and “+” search patterns; TSS(Three-Step Search), NTSS (New TSS) [5], 4SS (Four-Step Search) [6], and BBGDS(Block-Based Gradient Descent Search)[7] employ square search pattern; DS(Diamond

Search) exploits diamond search pattern [8]; HEXBS(Hexagon-Based Search) adopts hexagon search pattern [9]; OCTBS uses octagon search pattern [10]; etc. Some improved fast BMAs usually use two different search patterns in the searching procedure of motion vector, for example, CBHS adopts “X” and diamond patterns [11]; CDS and its improved algorithms use “+” and diamond pattern [12,13,14]; DTS employs diamond and triangle search patterns[15]; etc.

Based on the directional characteristic of SAD distribution and the center-biased characteristic of motion vectors, we proposed a fast BMA, DSSS(Diamond Simplified Square Search), in this paper.

The rest of this paper is organized as follows. Section 2 introduces our previous work briefly. The proposed DSSS is described in Section 3. Experimental results are presented in Section 4. Finally, conclusions are given in Section 5.

2 The Previous Work

In [15] we proposed a fast motion estimation algorithm based on diamond and triangle search patterns(DTS). The DTS algorithm adopts two search patterns adaptively in the process of motion search. The first pattern, called DP(Diamond Pattern, as shown in Fig.1(a)), comprises five checking points from which four points surround the center one to compose a diamond shape. The second pattern consisting of three checking points and covering the MBD point obtained in the previous search step(as shown in Fig.1(b)) forms a triangle shape, called TP(Triangle Pattern). In the process of motion search, DP is used to refine the motion vectors and it is necessary no matter how the motion vector being small or big, while TP is used to locate the best matching block with large motion vector approximately and it can be disused if the motion vector is ‘0’.

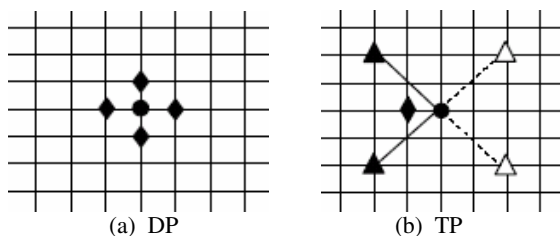


Fig. 1. Two search patterns employed in the proposed DTS algorithm

The DTS algorithm has the following technical characteristics. Firstly, the initial search center is formed according to the predicted motion vector of the current block by the adjacent blocks. Secondly, DP and TP are adaptively employed according to the motion extents of macro blocks.

By analyzing the DTS algorithm, we found that its speed of encoding is not so high for some video sequences, so we developed the proposed DSSS in the following.

3 The Proposed Diamond and Simplified Square Search Algorithm

3.1 DSSS Patterns

The proposed DSSS algorithm employs four search patterns adaptively in the process of motion search. The first pattern is called DP(as shown in Fig.2(a)). The second pattern consisting of two checking point forms a line shape, called LP(Line search Pattern) if the MBD(Minimum Block Distortion) and the SMBD(Second MBD) points obtained in the previous search step are located in different directions(as shown in Fig.2(b)). The third pattern consisting of three checking points and covering the MBD point forms a triangle shape, called TP(Triangle search Pattern) if the MBD and the SMBD points obtained in the previous search step are located in the same direction and the distance of the two points is only one pixel(as shown in Fig.2(c)). The fourth pattern consisting of four checking points forms a square shape, called SP(Square search Pattern) if the MBD and the SMBD points obtained in the previous search step are located in the same direction and the distance of the two points equals two pixels(as shown in Fig.2(d)). In the searching process of motion estimation, DP is

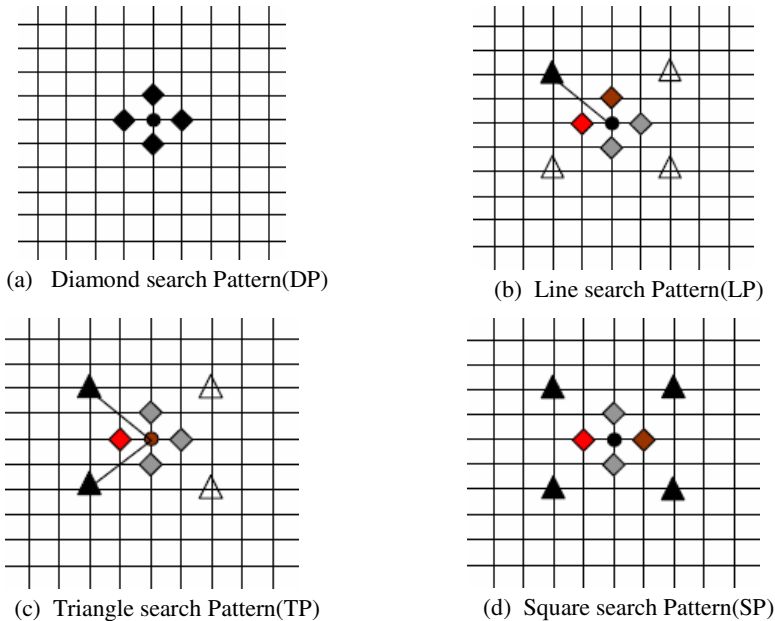


Fig. 2. Four search patterns employed in the proposed DSSS algorithm, only the solid black icons are the new checking points where the computational of block-distortion measurement is required, while the blank triangles are the skipped points, where '◆' and '◆' or '●' represent the MBD and SMBD point found in the previous search step respectively

used to refine the motion vectors and it is necessary no matter how the motion vector being small or big, while LP, TP or SP is used to locate the best matching block with large motion approximately and it can be discarded if the motion vector is '0'.

3.2 Description of the Proposed DSSS Algorithm

The proposed DSSS algorithm mainly comprises 4 stages. In the first stage, in order to reduce the search points for the best matching block with large motion, we use the median motion value of the adjacent blocks (as shown in Fig.3) to predict the motion vector of the current block. The median prediction is expressed as Eq. (1).

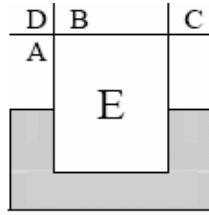


Fig. 3. Reference block location for predicting motion vector

$$pred_mv = median(mv_A, mv_B, mv_C) \quad (1)$$

In the second stage, in order to find the best matching block with zero or small motion vector efficiently, DP is selected as the search pattern. If the MBD point is located at the search center, then the motion search process terminates immediately and the best matching motion vector is equal to the predicted one. Assume that $P_0(x_0, y_0)$, $P_1(x_1, y_1)$ are the MBD and SMBD (Second MBD) points found in the current search step respectively, the searching pattern for the next search step can be decided by the distance from P_1 to P_0 , which is defined by Eq.(2).

$$P_1P_0 = (\Delta x_1, \Delta y_1) = (x_0 - x_1, y_0 - y_1) \quad (2)$$

In the third stage, the proposed DSSS algorithm selects a search pattern from LP, TP, and SP adaptively according to the distance between P_1 and P_0 . In the fourth stage, the proposed DSSS algorithm uses DP repeatedly until the new MBD point occurs at the center of DP or DP and LP/TP/SP alternately according to the position of the new MBD point found in the previous search step.

The block diagram of the proposed algorithm is depicted in Fig.4, and the proposed algorithm is summarized as follows.

Step1: Eq. (1) is used to predict the initial motion vector of the current block, and the initial search center point is set according to the predicted value.

Step2: DP is disposed at the search center, and the 5 checking points of DP(as shown in Fig.2(a)) are tested. If the minimum block distortion (MBD) point calculated is located at the center position of DP, then it is the final solution of the motion vector, goto **step5**. Otherwise, the new MBD point is re-positioned as the search center point, goto **step3**.

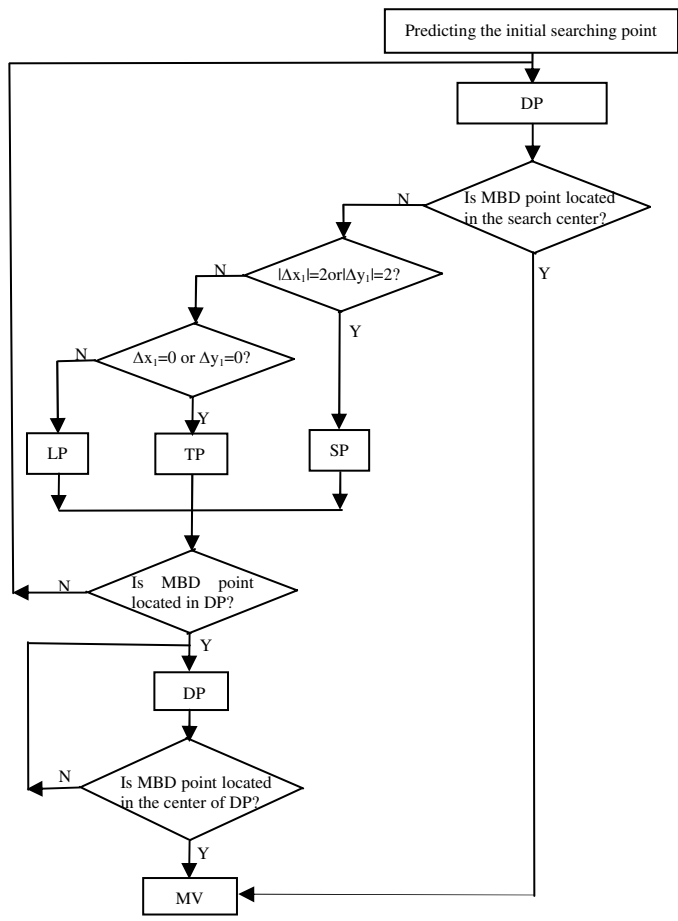


Fig. 4. The block diagram of the proposed algorithm

Step3: If($(|\Delta x_1|=2)$ or $(|\Delta y_1|=2)$)

- {
- SP is disposed at the new search center, and 4 checking points of SP (the 4 black squares as shown in Fig.2 (d)) are tested.
- }
- Else if($(\Delta x_1=0)$ or $(\Delta y_1=0)$)
- {
- TP is disposed at the new search center, and 2 checking points of TP (the 2 black squares as shown in Fig.2 (c)) are tested.
- }
- Else

```

{
  LP is disposed at the new search center, and 1 checking points of LP
  (the 1 black squares as shown in Fig.2 (b)) are tested.
}

```

If the MBD point is refreshed, the new MBD point is re-positioned as the search center, goto **step2**, otherwise, goto **step4**.

Step4: DP is disposed at the search center, and the 3 checking points of DP are tested according to the position of the MBD point. If the MBD point calculated is located at the center position, goto step5, otherwise, recursively repeat this step.

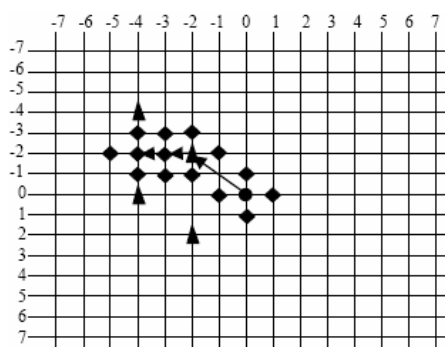
Step5: Stop searching. The center point is the final solution of the motion vector which points to the best matching block.

3.3 Analysis of the Proposed DSSS Algorithm

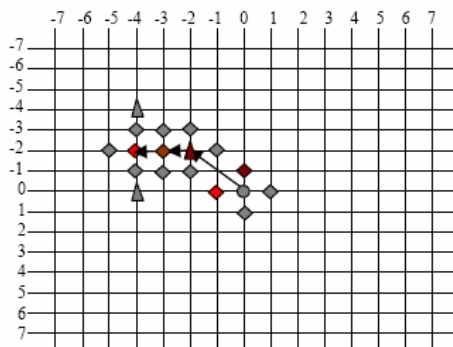
For BMME, computational complexity could be measured by average number of search points required for each motion vector estimation. According to the statistical distribution

Table 1. Comparison of least search points near the initial search center for DS,DTS and DSSS

	center	the best matching point is located in the		
		Circular area with a Radius of 1 pixel	Circular area with a Radius of $\sqrt{2}$ pixels	Circular area with a Radius of 2 pixels
DS	13	13	16	18
DTS	5	10	12	13
DSSS	5	9	11	12



(a) DTS uses six search steps—two times of TP and four times of DP. There are 19 search points in total—taking five, two, four, two, three, and three search points at each step, sequentially.



(b) DSSS uses six search steps—one time of LP, one time of TP, and four times of DP. There are 18 search points in total—taking five, one, four, two, three, and three search points at each step, sequentially.

Fig. 5. Search path example which leads to the motion vector (-4,-2) for DTS and DSSS

law of motion vectors in different images sequences, assume that the best matching point is located in a circle area with a radius of 2 pixels around the initial search point, the least search points needed for DS, DTS, and DSSS are listed in Table.1.

From Table.1 we observe that the least search points needed for DSSS is always less than that of DS, and the reduced search points is always 3~8. If the MBD point is not located in the initial search center, DSSS can reduce one search point comparing with that of DTS.

If the best matching point is located outside the circular area with a radius of 2 pixels, the least search points needed for DSSS is still less than that of DTS. This could be seen from the practical search path. Fig.5 gives a search path example which leads to the motion vector (-4,-2) for DTS and DSSS.

4 Experimental Results

Our proposed DSSS algorithm was integrated within version 7.6 of the H.264 software [16], and it is compared versus FS, DS, and DTS. Even though many image sequences are tested in the experiment, only four of them are selected out to be compared. The CABAC(Context-Adaptive Binary Arithmetic Coding) entropy coder[17] was used for all of our tests, with quantization parameter (QP) values of 28, 32, 36, and 40, a search range of ± 16 , and 2 references.

Table 2. The Average number of Search Points per macro-block

		FS	DS	DTS	DSSS
akiyo	QP=28	1089	13.04	5.14	5.12
	QP=32	1089	13.05	5.18	5.16
	QP=36	1089	13.10	5.22	5.20
	QP=40	1089	13.18	5.37	5.32
foreman	QP=28	1089	15.07	8.34	7.96
	QP=32	1089	15.12	8.40	8.00
	QP=36	1089	15.09	8.44	8.04
	QP=40	1089	14.99	8.42	8.01
coastguard	QP=28	1089	14.73	8.81	8.67
	QP=32	1089	14.72	8.54	8.39
	QP=36	1089	14.68	8.13	7.96
	QP=40	1089	14.51	7.80	7.65
mobile	QP=28	1089	14.57	8.81	8.59
	QP=32	1089	14.46	8.72	8.54
	QP=36	1089	14.56	8.77	8.51
	QP=40	1089	14.68	8.94	8.68

The four selected sequences are akiyo(Quarter Common Intermediate Format, QCIF), foreman(QCIF), coastguard(QCIF), and mobile(CIF). The former 100 frames of every sequence are tested, and only the first frame was encoded as I-frame, while the remainders are encoded as P-frames. Although H.264 provides seven different block-sizes for inter-frame coding, we have only used the 16×16 mode so as to compare the speed of motion search accurately. To simplify our comparison, we have used ASP(Average number of Search Points per macro-block) and RD(Rate Distortion) performance plot as shown in Table 2 and Fig.6 respectively.

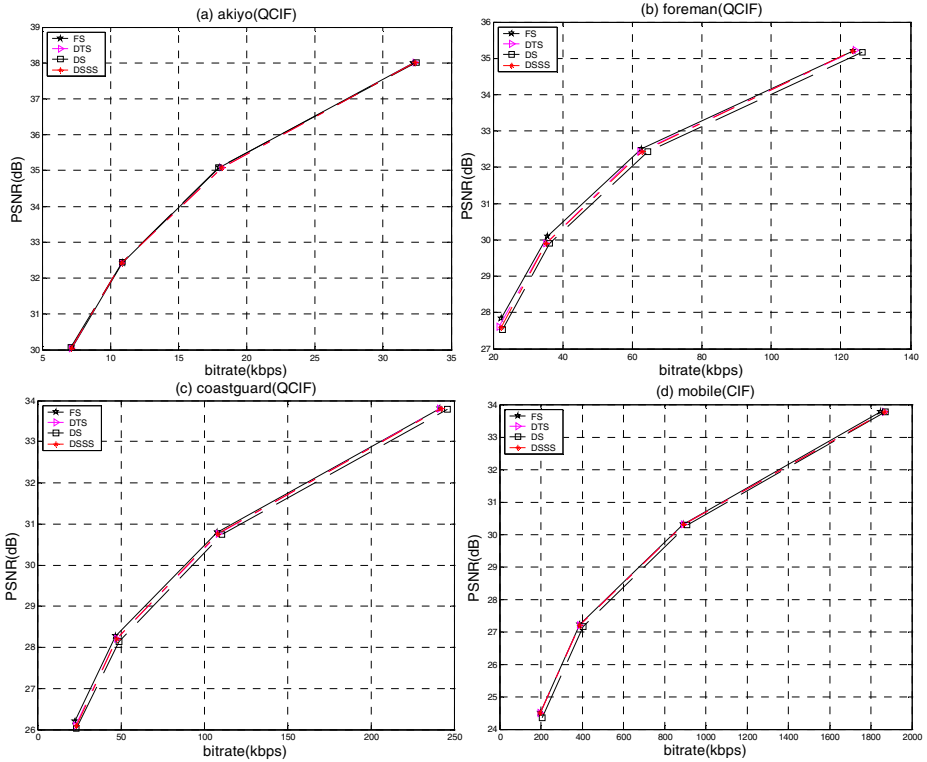


Fig. 6. RD performance plot for sequences (a) akiyo, (b) foreman, (c) coastguard, and (d) mobile

From Table 2 we can observe that the average number of search points per macro-block needed for DS, DTS, and DSSS are 13.04~15.12, 5.14~8.94, and 5.12~8.68 respectively. It's obvious that DSSS is faster than DS and DTS. From Fig.5 we can observe that the encoding efficiency of DSSS is better than DS and it is almost the same as that of DTS.

5 Conclusions

Based on the directional characteristic of SAD distribution and the center-biased characteristic of motion vectors, a fast BMA, DSSS, is proposed in this paper. DSSS

employs DP to refine the motion vectors, and LP, TP or SP adaptively according to the distance between the MBD and SMD point so as to locate the best matching block with large motion vector approximately. Although the proposed DSSS may also be trapped in local minima, experimental results show that it is faster than DS and DTS, while its encoding efficiency is better than DS and it is almost the same as that of DTS.

Acknowledgements. This work was supported by Grant No.04B055 from the Scientific Research Fund of Hunan Provincial Education Department and Grant No. 04JJ6032 from the Provincial Natural Science Foundation of Hunan.

References

1. T. Wiegand and G. Sullivan.: ITU-T Rec. H.264|ISO/IEC 14496-10 AVC, Final Draft, Document JVT-G050, 7th Meeting: Pattaya, Thailand, March 2003
2. J.Jain and A. Jain.: Displacement measurement and its application in interframe image coding. *IEEE Transaction on Communication*, vol. 29, pp.1799-1808, 1981
3. M. Ghanbari.: The cross-search algorithm for motion estimation. *IEEE Transaction on Communication*, vol. 38, pp. 950-953, July 1990
4. L. W. Lee, J. F. Wang, et al.: Dynamic search-window adjustment and interlaced search for block-matching algorithm. *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 3, pp.85-87, 1993
5. R.Li, B. Zeng, et al.: A new three-step search algorithm for block motion estimation. *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 4, pp. 438-442, Aug. 1994
6. L. M. Po and W. C. Ma.: A novel four-step search algorithm for fast block motion estimation. *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 6, pp. 313-317, June 1996
7. L. K. Liu and E. Feig.: A block-based gradient descent search algorithm for block motion estimation in video coding. *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 6, pp. 419-423, Aug. 1996
8. S. Zhu and K. K. Ma.: A new diamond search algorithm for fast block matching motion estimation. *IEEE Transaction on Image Processing*, vol. 9, pp.287-290, Feb. 2000
9. C. Zhu, X. Lin, and L.P. Chau.: Hexagon-based search pattern for fast block motion estimation. *IEEE Transaction on Circuits and Systems for Video Technology*, vol.12, pp.349-355. May 2002
10. L. P. Chau and C. Zhu.: A fast octagon-based search algorithm for motion estimation. *Signal Processing*, vol. 83, pp.671-675, 2003
11. Sung-Chul Shin, Hyunki Baik, et al.: A center-biased hybrid search method using plus search pattern for block motion estimation. *IEEE International Symposium on Circuits and systems*, Geneva, Switzerland, vol. IV, pp. 309-312, May, 2000
12. Chun-Ho Cheung and Lai-Man Po.: A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Trans. Circuit syst. video technol*, vol. 12, pp. 1168-1177, Dec. 2002.
13. Chun-Ho Cheung and Lai-Man Po.: A novel small-cross-diamond search algorithm for fast video coding and videoconferencing applications. *IEEE ICIP*, vol. I, pp.681-684, 2002.

14. Chi-Wai Lam, Lai-Man Po, et al.: A new cross-diamond search algorithm for fast block matching motion estimation. IEEE Int. Conf. Neural Networks & Signal Processing, pp. 1262–1265, Nanjing, China, 2003.
15. Y. Cheng, Z.Y. Wang, et al.: A fast motion estimation algorithm based on diamond and triangle search patterns. IbPRIA 2005, LNCS 3522, pp.419-426, 2005
16. http://bs.hhi.de/~suehring/ttml/download/old_jm/jm7.6, June 2004
17. D. Marpe, H. Schwarz and T. Wiegand.: Context-Based Adaptive Binary Arithmetic Coding in H.264/AVC Video Compression Standard. IEEE Transaction on Circuits and Systems for Video Technology, vol.13, pp. 620–636, July 2003