Journal of Zhejiang University SCIENCE A
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

A fast block-matching algorithm based on variable shape search

LIU Hao (刘 浩)[†1], ZHANG Wen-jun (张文军)[1], CAI Jun (蔡 骏)[2]

([1]*Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200030, China*)

([2]*Department of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta T2N 1N4, Canada*)

[†]E-mail: liu_hao@sjtu.edu.cn

Received Oct. 8, 2004; revision accepted Mar. 15, 2005

Abstract: Block-matching motion estimation plays an important role in video coding. The simple and efficient fast block-matching algorithm using Variable Shape Search (VSS) proposed in this paper is based on diamond search and hexagon search. The initial big diamond search is designed to fit the directional centre-biased characteristics of the real-world video sequence, and the directional hexagon search is designed to identify a small region where the best motion vector is expected to locate. Finally, the small diamond search is used to select the best motion vector in the located small region. Experimental results showed that the proposed VSS algorithm can significantly reduce the computational complexity, and provide competitive computational speedup with similar distortion performance as compared with the popular Diamond-based Search (DS) algorithm in the MPEG-4 Simple Profile.

Key words: Motion estimation, Block-matching, Variable shape search, MPEG-4 Simple Profile
doi:10.1631/jzus.2006.A0194 **Document code:** A **CLC number:** TN919.81

INTRODUCTION

Block-matching motion estimation is a key video coding technology that reduces the temporal redundancy between adjacent frames. The 3GPP/3GPP2 mandatory video codecs (MPEG-4 Simple Profile or H.263 Baseline) have adopted this technology extensively. However, block-matching motion estimation is quite computationally intensive if the Full Search (FS) algorithm is used which exhaustively checks all possible candidate motion vectors within the search window. Therefore, fast block-matching algorithms (BMAs) are useful for accelerating the process without serious distortion, and are especially suitable for the current 3G real-time video applications.

Most of the search shapes in the well-known fast BMAs are regular and symmetrical in horizontal or vertical directions, for example, New Three-Step Search (NTSS) (Li *et al*., 1994), Four-Step Search (4SS) (Po and Ma, 1996), Hexagon-based Search (HS) (Zhu *et al*., 2002) and Diamond-based Search (DS) (Tham *et al*., 1998; Zhu and Ma, 2000; Cheung and Po, 2002) as shown in Fig.1. At present in the MPEG-4 Simple Profile, the DS algorithm is a kind of relatively outstanding fast algorithm whose complexity and precision has been recommended by the MPEG-4 video Verification Model (VM).

In this paper, we propose a simple and efficient Variable Shape Search (VSS) algorithm that integrates some merits of both the diamond search and the hexagon search in the MPEG-4 Simple Profile. In Section 2, we will analyze the new VSS algorithm. The computational gains and the experimental results of the proposed VSS algorithm as compared with that of the existing NTSS, 4SS, and DS algorithms are presented in Section 3. The conclusions are given in Section 4.

VARIABLE SHAPE SEARCH

Fundamentally speaking, search methods with different shapes or sizes in the fast BMAs can de-

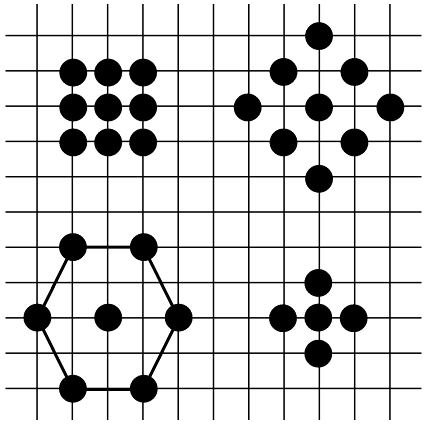
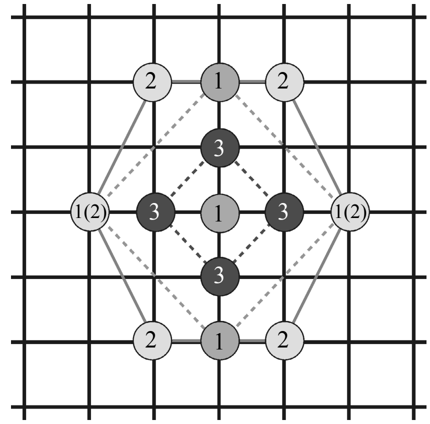


Fig.1 Some search methods used in the fast BMAs

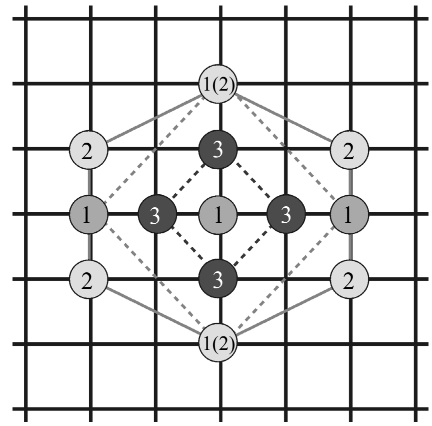
termine not only their search speeds but also resulted performance. Statistical experiments have showed that the motion field of real-world video sequences is usually stationary or quasi-stationary (Jain and Jain, 1981; Ghanbari, 1990), which results in a center-biased motion vector distribution instead of a uniform motion vector distribution. Furthermore, about 50% to 95% of the motion vectors are enclosed in a circular area with radius of two pixels and centered on the position of zero motion (Zhu and Ma, 2000). Therefore, our VSS algorithm will first check these points in the most possible region.

The geometry proved that the triangle, the diamond and the hexagon are only three kinds of polygons that can cover the overall search plane. If the distortion within a small neighborhood around the global minimum increases monotonically, a circle-shaped search method is effective for achieving the fastest search speed uniformly. The diamond or triangle shape is not approximate enough to a circle compared with the hexagonal shape (Zhu *et al*., 2002). However, the frequency spectrum of most real-world video sequences is similar to that of the diamond shape, and the block displacement of motion objects can be in any direction, but mainly in horizontal and vertical directions (Cheung and Po, 2002).

Considering the merits of the diamond search and hexagon search, our block-matching process mainly comprises three phases: (1) the big diamond search to fit the directional centre-biased characteristics of the real-world video sequence; (2) the directional hexagon search to identify a small region where the best motion vector is expected to locate; (3) the small diamond search to select the best motion vector in the located small region.



(a)



(b)

Fig.2 Two search patterns for the VSS algorithm
(a) Pattern 1 (1: big diamond; 2: horizontal hexagon; 3: small diamond); (b) Pattern 2 (1: big diamond; 2: vertical hexagon; 3: small diamond)

The proposed VSS algorithm employs two basic search patterns as illustrated in Fig.2: Pattern 1 and Pattern 2. For the two search patterns, the big diamond comprises five points marked as ‘1’ including the center point. Like the shrunk diamond, a small diamond covering four points marked as ‘3’ is inside the big diamond, which is finally applied to the focused search. For the hexagon search, we can use the six endpoints marked as ‘2’. Because the block displacement of real-world video sequences can be mainly in horizontal and vertical directions, the VSS algorithm adopts two kinds of asymmetrical hexagonal search shapes with different directionality separately, i.e., horizontal hexagon and vertical hexagon.

The Mean Absolute Difference (MAD), rather than Mean Square Error (MSE), is used as the matching criterion to reduce the block-matching computation in practice. Many correlated modules in a video codec may affect the final Peak Signal-to-Noise Ratio (PSNR) of the reconstructed video, and it is effective to use the MAD criterion to evaluate the performance of different fast BMAs.

With the switching strategy of variable shape search, we develop the following search method as depicted in Fig.3.

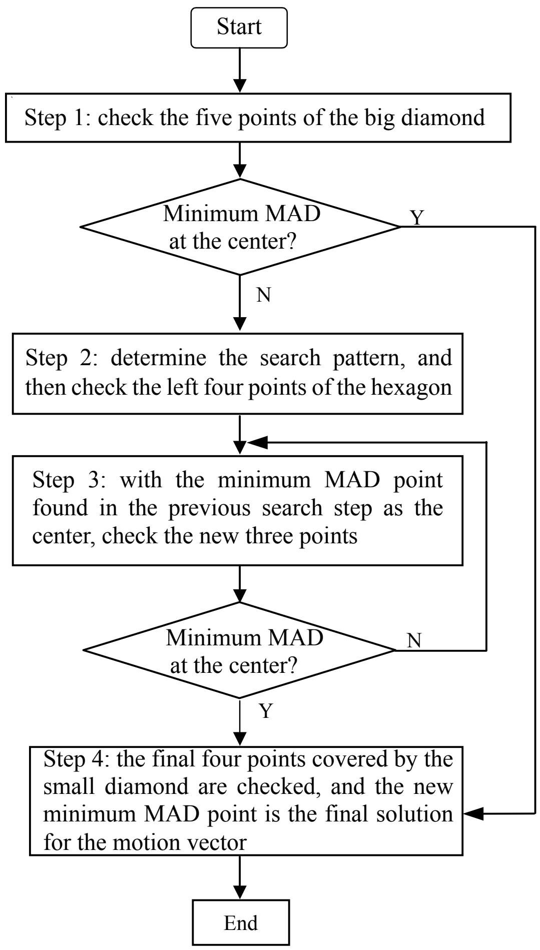


Fig.3 Flowchart of the VSS algorithm

Step 1: All the five points of the initial big diamond within a search window around the motion vector predictor is checked first. For each of the five points, the MAD is computed and compared. Here considering the regularity and simplicity of hardware-oriented features, we do not adopt the early termination technology which only depends on an uncertain and experiential threshold. If the minimum MAD point is found at the center of the big diamond, then go to Step 4; otherwise, go to Step 2.

Step 2: Considering the distribution characteristic of the motion vector and real shape of the hexagon, our algorithm needs to determine the directionality of the hexagon search for the subsequent steps. If the minimum MAD point in the previous search step is located at the horizontal corner of the big diamond, then check the left four points of the horizontal hexagon using Pattern 1. Otherwise, if located at the vertical corner of the big diamond, then check the left four points of the vertical hexagon using Pattern 2. Note that the determined search pattern is not changed until the end of the particular search.

Step 3: The minimum MAD point found in the previous search step is repositioned as the center point to form a new hexagon, and only three new non-overlapped points will be checked as candidates each time. If the minimum MAD point is still the center point of the newly formed hexagon, then go to Step 4 for the focused search; otherwise, maintain the hexagonal shape unchanged and repeat Step 3 recursively. Note that any candidate point outside the search window is ignored, and the VSS algorithm does not restrict the number of the search steps essentially.

Step 4: Switch the search shape from the hexagon to the small diamond. The final four points covered by the small diamond are checked to compare with the current minimum MAD point. The MAD point found in this step is the final solution of the motion vector which points to the best matching block.

In this way, the search process of the current block is completed, and the codec will proceed to Step 1 for the next block, if any.

EXPERIMENTAL RESULTS

In this section, we will examine the proposed VSS algorithm for comparison with the popular NTSS, 4SS and DS algorithms. For the fast BMAs, computational complexity can be measured by the Number of Search Points (NoSP) required for each block-matching. The experiments are performed based on the MPEG-4 video reference software (VM version 15.0). Several standard video sequences (CIF, 30 frames/s) including Salesman, Claire, Tennis, Gardener and Mobile&Calendar are used, which consist of different degrees and types of motion content. The MAD is used for the distortion measurement, and all fast BMAs are only used for the 16×16 integer-pixel search. The search window size is [–15,+15] and has no skipped frames. The average MAD values and average NoSP for the different video sequences are summarized in Table 1 and Table 2 for the different fast BMAs including the NTSS, 4SS, DS and our proposed VSS, respectively.

Table 1 Average MAD per pixel for different fast BMAs and video sequences

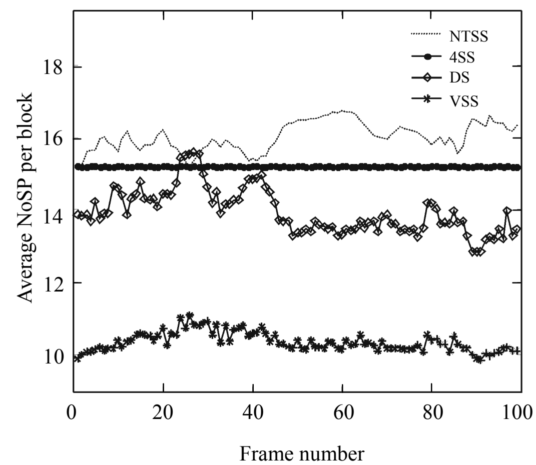
	Salesman	Claire	Tennis	Gardener	Mobile& Calendar
NTSS	1.68	1.73	15.79	1.49	10.05
4SS	1.70	1.71	15.05	1.49	10.26
DS	1.69	1.91	15.85	1.51	12.46
VSS	1.70	1.94	16.04	1.51	12.70

Table 2 Average NoSP per block for different fast BMAs and video sequences

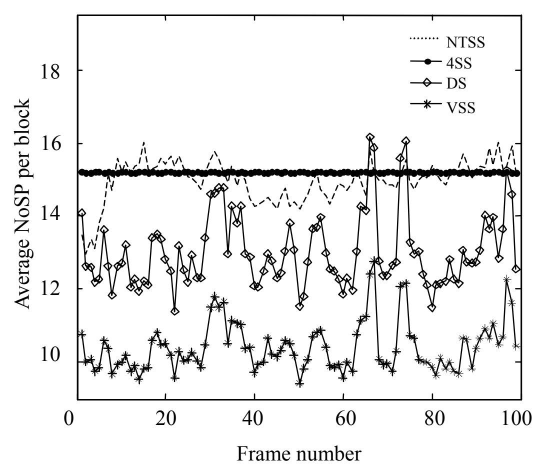
	Salesman	Claire	Tennis	Gardener	Mobile& Calendar
NTSS	14.32	15.00	19.88	14.52	15.98
4SS	15.19	15.20	15.25	15.19	15.20
DS	11.88	13.01	26.85	11.24	13.55
VSS	9.69	10.40	19.89	9.34	10.34

It can be clearly seen that the proposed VSS algorithm achieves the smallest NoSP with marginal increase in terms of the MAD compared with other fast BMAs. Table 1 demonstrates the similar MAD performance for all the BMAs tested. It is obvious as shown in Table 2 that the VSS has better NoSP performance compared to the NTSS, 4SS or DS. Here we mainly compare the DS algorithm adopted in MPEG-4 VM with the proposed VSS algorithm in terms of the NoSP as well as the MAD. For the video conferencing sequences such as Salesman, the VSS achieves 22.60% NoSP speedup over the DS, while the MAD only increases 0.59%. For the sequence Mobile&Calendar with relatively higher degree of motion, the VSS gives about 31.04% NoSP speedup, while the MAD only increases 1.92%. For the sequence Tennis with fast motion content, as predicted in theory, the VSS yielded higher NoSP speedup over the DS, here up to 34.99%, while reasonably introducing slight degradation of less than 1.20% in quality. Generally speaking, the larger motion the video sequences have, the larger the speedup ratio of the VSS over the DS. On the other hand, the MAD increase of our VSS algorithm compared with the DS algorithm is trivial, less than 2.0% for all the video sequences in our experiments. At the price of slight quality degradation compared with the DS algorithm, the realization of the VSS algorithm may consume fewer MAD calculations each time, thus saving cost and hardware size.

Fig.4 plots the corresponding frame-wise NoSP comparison for the different fast BMAs applied to the sequence Mobile&Calendar and the sequence Claire. The figure further shows that the proposed VSS algorithm can speed up the procedure of the block-matching significantly.



(a)



(b)

Fig.4 NoSP comparison for NTSS, 4SS, DS and VSS using the sequence Mobile&Calendar (a) and using the sequence Claire (b)

CONCLUSION

In this paper, an effective VSS algorithm for the fast BMAs in the MPEG-4 Simple Profile is proposed. Experimental results showed that the proposed VSS algorithm improves the search speed significantly with similar distortion, especially for video sequences containing complex motion. The VSS algorithm possesses the regularity and simplicity of hardware-oriented features, and is suitable for the current 3G real-time video applications.

References

Cheung, C.H., Po, L.M., 2002. A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Trans. on Circuits & Systems for Video Technology*, **12**(12):1168-1177. [doi:10.1109/TCSVT.2002.806815]

Ghanbari, M., 1990. The cross-search algorithm for motion estimation. *IEEE Trans. on Communications*, **38**(7):950-953. [doi:10.1109/26.57512]

Jain, J.R., Jain, A.K., 1981. Displacement measurement and its application in interframe image coding. *IEEE Trans. on Communications*, **29**(12):1799-1808. [doi:10.1109/TCOM.1981.1094950]

Li, R., Zeng, B., Liou, M.L., 1994. A new three-step search algorithm for block motion estimation. *IEEE Trans. on Circuits & Systems for Video Technology*, **4**(4):438-442. [doi:10.1109/76.313138]

Po, L.M., Ma, W.C., 1996. A novel four-step search algorithm for fast block motion estimation. *IEEE Trans. on Circuits & Systems for Video Technology*, **6**(6):313-317.

Tham, J.Y., Ranganath, S., Ranganath, M., Kassim, A.A., 1998. A novel unrestricted center-biased diamond search algorithm for block motion estimation. *IEEE Trans. on Circuits & Systems for Video Technology*, **8**(8):369-377.

Zhu, S., Ma, K.K., 2000. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. on Image Processing*, **9**(2):287-290. [doi:10.1109/83.821744]

Zhu, C., Lin, X., Chau, L.P., 2002. Hexagon-based search pattern for fast block motion estimation. *IEEE Trans. on Circuits & Systems for Video Technology*, **12**(5):349-355. [doi:10.1109/TCSVT.2002.1003474]