An Efficient Search Algorithm for Motion Estimation

Shung-Chih Chen[1], Tzyy-Kuen Tien[2], and Chung-Wei Tsai[3]

Department of Electronic Engineering, Southern Taiwan University
No.1 NanTai St., YongKang City, Tainan County, Taiwan
{[1]scchen and [2]tktien}@mail.stut.edu.tw
[3]m9535204@webmail.stut.edu.tw

***Abstract*— An efficient search algorithm using the principle of Hexagon-Based Search algorithm (HEXBS) for motion estimation called Hexagon-Based Fast Search algorithm (HEXFS) is proposed in this paper. The proposed algorithm combines the hexagon search algorithm and a two-step search algorithm which uses cross search patterns to reduce the huge calculation complexity. The experimental results show that the search points' improvement ratio of the modified search algorithm can be as high as almost 70% comparing with Unsymmetrical-Cross Multi-Hexagon-Grid Search (UMHexagonS). In short, the proposed algorithm can find the same motion vector (MV) with fewer search points than which are obtained by UMHexagonS.**

***Keywords*— Motion estimation, hexagon-based search, cross search patterns, motion vector.**

1. INTRODUCTION

Motion estimation is a process for estimating motion vectors (MV) that transform from reference frame to current frame in video sequence coding. Block-matching algorithm for motion estimation is a temporal redundancy elimination technique between two or more consecutive frames for video compression. It can make the inter-frame coding achieve a very high compression ratio. Comparing with various motion estimation methods, block-matching algorithm is a vital technique for the international video compression standards, like MPEG-1 [1][2], MPEG-2 [3], MPEG-4 [4][5], H.261 [6], H.263 [7], and H.264 [8].

Among all of the existing block-matching search algorithms, the Full-Search algorithm, which performs searching all possible candidate blocks within the search window, can obtain the best accuracy for motion estimation. However, it is quite computationally intensive and consumes up to 80% of the computational power of the encoder. Thus, efficient algorithms are highly needed to reduce the calculation burden of Full-Search algorithm. These algorithms utilize a kind of fixed search pattern to decrease the search points in the search window. There are several approaches such as the Three-Step Search algorithm (TSS) [9], the New Three-Step Search algorithm (NTSS) [10], the Four-Step Search algorithm (FSS) [11], Diamond Search algorithm (DS) [12], and Hexagon-Based Search algorithm (HEXBS) [13]. These fixed search patterns can acquire a good video quality without too many search points. These techniques performed well in the comparatively stationary image sequences and the small picture size. Nevertheless, coding effectiveness of these methods is not good enough when the image size and the motion of the image are too large. Hence, some other types of algorithms were proposed to reduce the calculation for every search candidate. They use sub-sampling technique and partial calculation to obtain better search result. Sub-sampling technique is effective for reducing the search time, but it is not accurate; partial calculation is accurate for search point, but it is not efficient enough. Therefore, many fast hybrid algorithms have been proposed recently. For example, Unsymmetrical-Cross Multi-Hexagon-Grid Search (UMHexagonS) [14] combines the hybrid and hierarchical motion search strategy to improve both fast speed and good image quality. However, these algorithms are very complicated that they are not fast enough and not easy to be transplanted to hardware implementation.

In the following section, the hexagon pattern search algorithm will be examined to discuss its advantages and disadvantages. Section 3 will present and describe our Hexagon-Based Fast Search algorithm thoroughly. Experimental results will be illustrated in the section 4. We will compare our method with many other algorithms which have been proposed. Section 5 concludes this paper.

2. REVIEW OF REFERENCE TECHNIQUES

Our proposed method is related to hexagon-based search patterns which are reformed from diamond search patterns. In this section, we will review these two algorithms.

2.1. Diamond Search Algorithm

Diamond Search algorithm (DS) is an efficient search technique which uses two search patterns as illustrated in Fig. 1. The first pattern, called large diamond search pattern (LDSP), consists of nine checking points from which eight points surround the center one to form a diamond shape. The second pattern containing five checking points creates a smaller diamond shape, called small diamond search pattern (SDSP). Fig. 2 shows three cases of checking-point overlapping in LDSP when the minimum block distortion point found in the previous search step (blue dots) is located at (a) one of the corner points, (b) one of the edge points, and (c) the center point. The black dots are the new checking points where the computation of block-distortion measurement is required for the current search step. In the searching procedure of the Diamond Search algorithm, LDSP is repeatedly used until the minimum block distortion occurs at the center point. The search pattern is then switched from LDSP to SDSP as reaching to the final search stage. Among the five checking points in SDSP, the point with the minimum block distortion is the best matching block. Fig. 3 is an example of the Diamond Search algorithm.

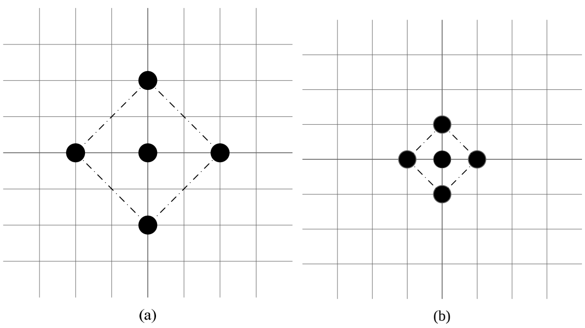

Fig. 1 Two search patterns of Diamond Search algorithm: (a) large diamond search pattern; (b) small diamond search pattern.

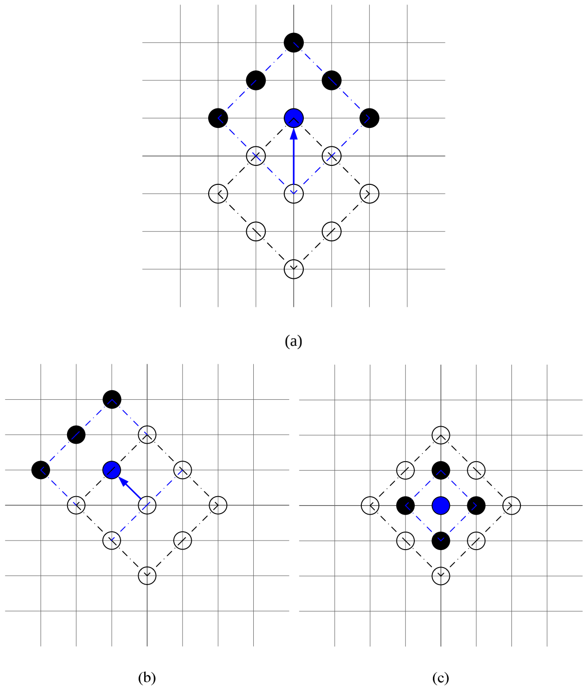

Fig. 2 Three cases for checking points of Diamond Search algorithm: (a) the corner point: LDSP → LDSP; (b) the edge point: LDSP → LDSP; (c) the center point: LDSP → SDSP.

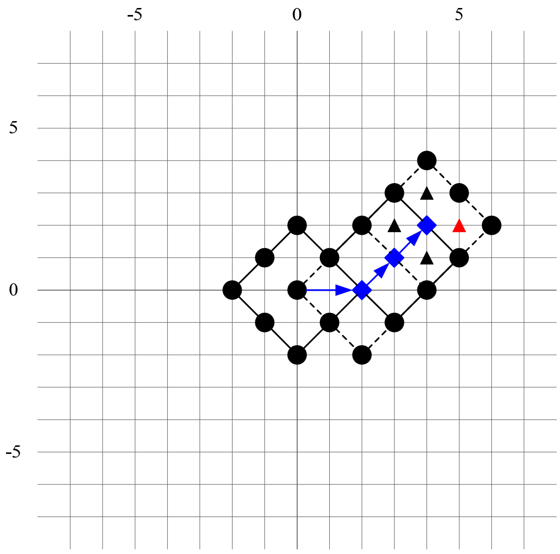

Fig. 3 Search example leading to the motion vector (5, 2) for Diamond Search algorithm.

2.2. Hexagon-Based Search Algorithm

The introduction of the Diamond Search algorithm tells the importance of the search pattern's shape and size in motion estimation. In [13], the authors proposed the hexagon-based search patterns (HEXBS). The hexagon has two patterns. The larg one consists of seven search points with one center surrounded by six points. The small hexagon is like SDSP having four search points around the center. The two hexagon-based search patterns are illustrated as Fig. 4. Fig. 5 shows two cases of checking-point overlapping in HEXBS when the minimum block distortion point found in the previous search step (blue dots) is located at (a) one of the horizontal points, (b) one of the diagonal points, and (c) the center point. The search procedures of the HEXBS algorithm can be described as follows. In the first step, the large hexagon-based search pattern (LHSP) with seven search points is checked to find out the minimum block distortion point. If the best point is at the center, the large hexagon-based search pattern will be replaced by a small hexagon-based search pattern (SHSP). Then the four search points covered by this small hexagon-based search pattern will be examined to determine the final solution with minimum block distortion. Otherwise, the search procedure continues by shifting the center of the large hexagon-based search pattern to the point with minimum block distortion and evaluating the new candidate points.

We can notice that while the large hexagon-based search pattern moves along the direction, only three new non-overlapped search points will be evaluated each time. Fig. 6 illustrates an example of the search leading to the motion vector, where 17 search points are evaluated in four steps in sequence. The simulation results presented in [13] confirm that a hexagon-based search algorithm can achieve substantial speed improvement over the TSS, the NTSS, the FSS, and the DS algorithms with similar distortion performance. Moreover, more search points can be saved by the HEXBS algorithm.

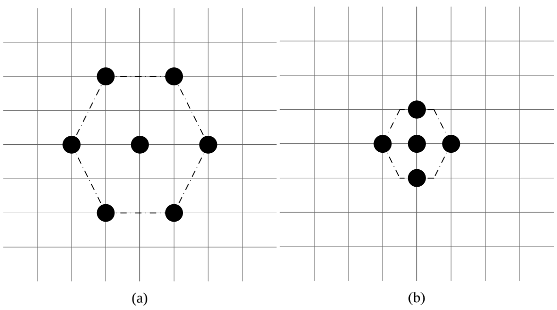

Fig. 4 Two search patterns of Hexagon-Based Search algorithm: (a) large hexagon-based search pattern; (b) small hexagon-based search pattern.

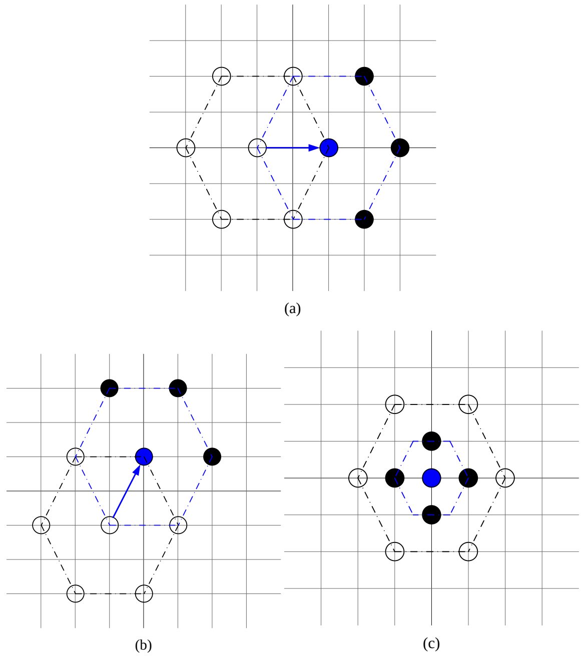

Fig. 5 Three cases for checking points of Hexagon-Based Search algorithm: (a) the horizontal point: LHSP → LHSP; (b) the diagonal point: LHSP → LHSP; (c) the center point: LHSP → SHSP.

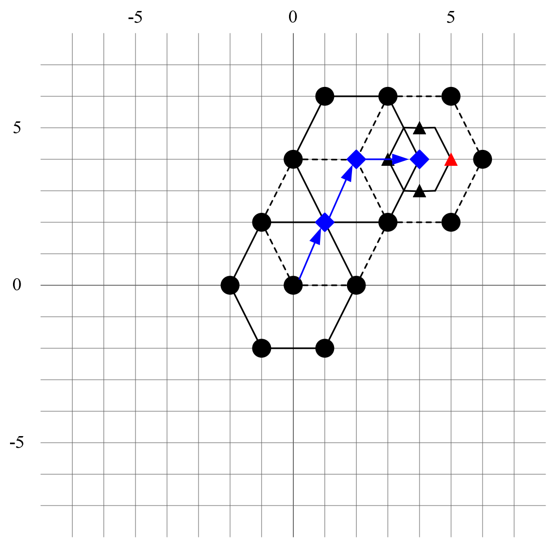

Fig. 6 Search example leading to the motion vector (5, 4) for Hexagon-Based Search algorithm.

2.3. Overview of DS and HEXBS

According to the experimental results of DS and HEXBS, these two methods improve the search performance comparing to the Full-Search algorithm. Also, they perform well in the comparatively stationary image sequences and the small picture size. However, coding efficiency of these techniques is not good enough when the image size and the transformation of the MV are large. Therefore, we proposed an efficient search algorithm to advance the performance for coding the large image.

3. PROPOSED ALGORITHM

Diamond Search and Hexagon-Based Search algorithms are two efficient methods for motion estimation, but they still have two main problems. First, they search too many search points. Second, the speeds of convergence by large diamond search pattern and large hexagon-based search pattern are not good enough. Therefore, we use a hybrid technique to solve these problems.

At first we utilize the small hexagon-based search pattern or large hexagon-based search pattern to search. It is used for a coarse search for the minimum block distortion point and for small motion. Then we use small hexagon-based search pattern to do the fine search. It will reduce some search points for the point near the center comparing with DS or HEXBS algorithms. Secondly, we use two kinds of cross search Pattern (CSP) to solve the slow convergence problem. CSP is mainly for large motion. It is not only suitable for the horizontal and vertical direction movement, but also the diagonal direction, so it is easy to find a coarse position for the minimum block distortion point. After that coarse search, we use large hexagon-based search pattern and small hexagon-based search pattern to do the fine search. It is a two-step coarse-to-fine approach.

In [15], we learned about that we can classify the motion for the macro-block into three different types. One is fast motion, another is medium motion, and the last one is slow motion. The distance of slow motions is smaller than 2 pixels (in the small hexagon of Fig. 7). The distance of medium motions is larger than 2 pixels but less than 6 pixels from the search center (between the small hexagon and the big hexagon of Fig. 7). The fast one is larger than 6 pixels (outside of the big hexagon of Fig. 7). Our two-step search is based on this classification. From Fig. 7, we can see most possibility of the motion is the slow motion and the medium motion, so we should focus on these two motions more and save as many search candidate points as possible.

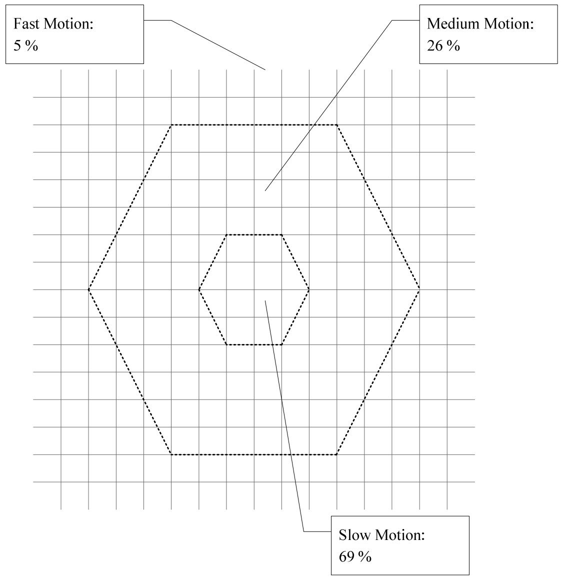

Fig. 7 The distribution of the motion vector.

According to the statistics, we use large hexagon-based search pattern and small hexagon-based search pattern for two slow motions respectively. Our large hexagon-based search pattern is different from HEXBS's large hexagon-based search pattern as Fig. 8. Its length is twice longer (4 pixels) than the HEXBS's large hexagon-based search pattern (2 pixels). We use small hexagon-based search pattern as fine search; small hexagon-based search pattern in the first step is to solve the slow motion, and large hexagon-based search pattern is to solve the medium motion. For the fast motion, we employ the x cross search pattern (XCSP) and rood cross search pattern (RCSP) to do coarse search. Its radius is 6 pixels distance, which fulfill the coarse hexagon's radius, and they are used to find the coarse hexagon center. Fig. 8 shows our four kinds of search patterns.

Fig. 9 shows an example of our Hexagon-Based Fast Search algorithm (HEXFS). First of all, we search around the center point for slow motion by Step 1.1 and medium motion by Step 1.2 respectively. In Step 1.1, if the result meets the early termination criterion, we go to Step 4, but if not, go to Step1.2. In Step 1.2, if the minimum block distortion point is found at the center, go to Step 3; otherwise, we go to the Step 2- fast motion search. We use XCSP on Step 2.1. After finding the best coarse search point, we use RCSP to refine it in Step 2.2 until the center point of cross search is the best candidate point, then we use large hexagon-based search pattern on Step 3 and small hexagon-based search pattern on Step 4 to get the best fine search point. This is a fast method to find the minimum block distortion point and gets rid of the disadvantages of HEXBS.

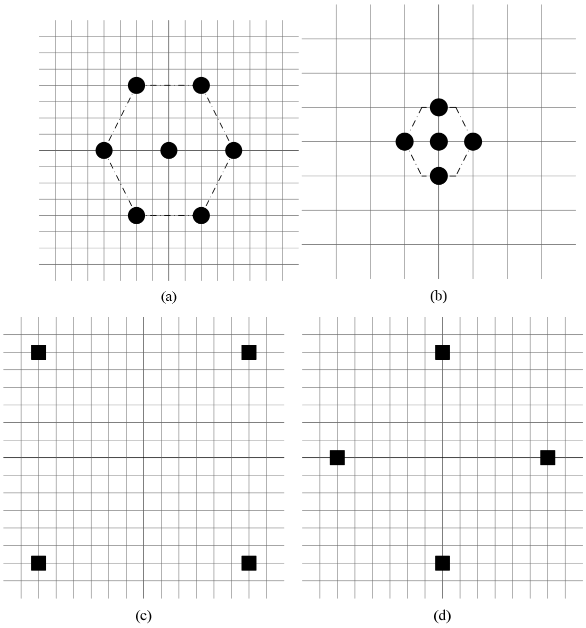

Fig. 8 Four kinds of search patterns: (a) large hexagon-based search pattern; (b) small hexagon-based search pattern; (c) x cross search pattern; (d) rood cross search pattern.

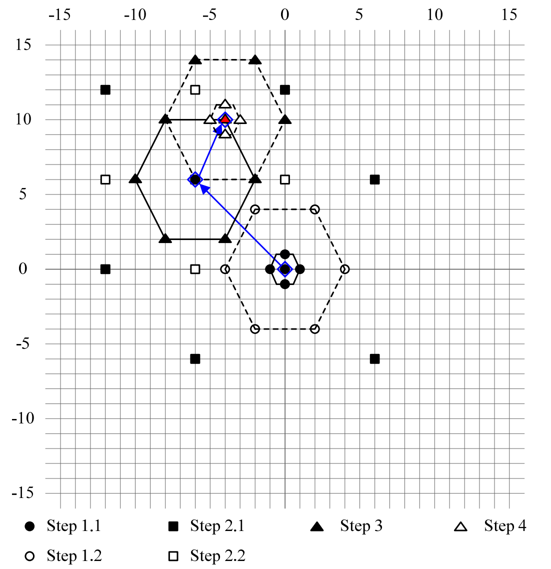

Fig. 9 Search example leading to the motion vector (-4, 10) for our Hexagon-Based Fast Search algorithm.

4. Experimental Results

Table I shows the number of search points per macro-block for motion vector search on QCIF video sequence. It demonstrates that our proposed algorithm is the most effective method comparing with Two-Step Cross-Diamond Fast Search algorithm (TSCDS) and UMHexagonS.

TABLE 1
NUMBER OF SEARCH POINTS PER MACROBLOCK (PROPOSED METHOD IS HIGHLIGHTED)

Number of Search Points Per Macroblock	UMHexagonS [14]	TSCDS [15]	**Proposed HEXFS**
football	367.360	138.002	**106.155**
tempête	313.375	114.428	**93.830**
mobile	303.175	107.892	**87.393**
coastguard	244.215	98.348	**77.695**
foremen	201.340	89.885	**73.706**
carphone	165.865	77.918	**62.334**
table	166.600	77.135	**60.165**

Table II illustrates that our algorithm is less than UMHexagonS about 70% search points, and it can save around 20% search points comparing with TSCDS. That is why our proposed algorithm will be faster than UMHexagonS and other conventional algorithms.

TABLE 2
THE PERFORMANCE OF HEXFS COMPARED WITH OTHER ALGORITHMS (PROPOSED METHOD IS HIGHLIGHTED)

	UMHexagonS [14]	TSCDS [15]	**Proposed HEXFS**
Average Number of Search Points Per Macroblock	251.704	100.515	**80.183**
Average Winning Points Using HEXFS	**171.521**	**20.332**	
Average Reduction Percentage Using HEXFS	**68.14%**	**20.23%**	

5. Conclusions

In this paper, we proposed an efficient search algorithm for motion estimation called Hexagon-Based Fast Search algorithm (HEXFS). HEXFS utilizes two-step coarse to fine strategy and hexagon-based search patterns to search the motion vector. This technique solves the problems of the conventional algorithms for motion estimation as searching too many search points. Experimental results show that our HEXFS algorithm save around 70% search points comparing with the existing methods. Shortly, HEXFS is a very effective algorithm for real-time video coding applications.

This work was supported in part by the National Science Council under Grant NSC98-2221-E-218-018

REFERENCES

[1] D. J. L. Gall, "The MPEG video compression algorithm," *Signal Processing: Image Commun., 1992, Vol. 4, No. 4,* pp. 129-140.

[2] ISO/IEC 11172-2 (MPEG-1 Video), "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s: Video," 1993.

[3] ISO/IEC 13818-2 | ITU-T H.262 (MPEG-2 Video), "Information technology - Generic coding of moving pictures and associated audio information: Video," 1995.

[4] ISOIIEC JTCl lISC291WG11 N2323, "Overview of the MPEG-4 standard," Jul. 1998.

[5] ISOIIEC JTCllISC291WGll W2502, "ISOIIEC 14496-2. Final draft international standard. part 2: Visual," Atlantic-City, Oct. 1998.

[6] CCITT SGXV, "Description of reference model 8 (RM8)," Document 525, Working Party XV/4, Specialists Group on Coding for Visual Telephony, Jun. 1989.

[7] "Video coding for low bit rate communication," *ITU-T, ITU-T RecommendationH.263* Ver. 1, 1995.

[8] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC)," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG,* JVTG050, 2003.

[9] J. Jain and A. Jain "Displacement measurement and its application in inter-frame image coding," *IEEE Trans.s on Commun.s, Vol. 29,* pp. 1799-1806, Dec. 1981.

[10] 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.

[11] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuit Syst. Video Technol., Vol. 6,* pp. 313-317, Jun. 1996.

[12] S. Zhu and K-K. M, "A new diamond search algorithm for fast block matching motion estimation," *Proc. Int. Conf: Inform., Commun., Signal Process*, pp. 292-296, Sept.1997.

[13] C. Zhu, X. Lin, and L. Chau, "Hexagon-based search patten for fast block motion estimation," *IEEE Trans. on CSVT, Vol. 12, No. 5,* pp. 349-355, May 2002.

[14] Z. Chen, P. Zhou, and Y. He, "Fast motion estimation for JVT," *JVT-G016.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 7th meeting*, Pattaya II, Thailand, 7-14 Mar. 2003.

[15] Q. Liu, S. Hiratsuka, S. Goto, and T. Ikenaga, "Two-step cross-diamond fast search algorithm on motion estimation in H.264," *Commun., Circuits and Syst., 2007. ICCCAS 2007. Int. Conf.*, pp. 782-786, Jul. 2007.