IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 2, July 2012
ISSN (Online): 1694-0814
www.IJCSI.org

An Improvement of Unsymmetrical-Cross Multi-Hexagon-grid Search algorithm

Vu Thanh Nguyen[1], Tran Xuan Chien[2], Phan Cong Canh[3]

Software Engineering Department, University of Information Technology
Ho Chi Minh City, Vietnam

***Abstract*— In this paper, we present an new algorithm based on the Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) algorithm and and the Fast Directional Gradient Descent Search (FDGDS). The experiment results show that this new algorithm can reduce the motion estimation time with just a very minor bit rate increase and almost the same image quality.**

***Keywords*— Video compression, block matching, improvement, motion estimation, video coding, UMHexagonS.**

I. INTRODUCTION

Block matching motion estimation in video compression is a technique used to find the best matching image block in the previous frame from the image block in current frame. The best matching block is found by using certain similarity measurement, such as sum of absolute (SAD) or mean of square error (MSE). However, the SAD is commonly used because of its simplicity and efficiency.

Obviously, the simplest method is full search algorithm that exhaustively searches all the points in the search window to find the best matching candidate. However, it requires lots of computational operation and is not feasible in most of video application. For years, researchers have proposed many hybrid algorithms ([1]-[9]) to tackle the problem of full search, each adapt different search point patterns, trying to reduce the motion estimation time without affecting much the video quality and the bit rate. These algorithms, however, can easily be trapped in local minimum point, especially in video with rapid motion scenes.

In H.264 Reference Software, UMHexagonS ([10]) is adapted as the main search algorithm that is used to find the best matching block in search window. This algorithm uses different kind of patterns to find the best matching block, and it has the early termination strategy to reduce the search time. This algorithm works very well with both slow and fast motion video sequences. It runs quite fast and gives accurate motion vectors, and good rate-distortion performance, nearly the same as the full search algorithm.

Though the performance of UMHexagonS is good, it could be improved a little bit more. Lai-Man Po introduces a new algorithm called Fast Directional Gradient Descent Search (FDGDS) ([11]). This new algorithm runs faster than the UMHexagonS. However, it increases the bit rate of compressed video, especially video sequences with rapid motion scenes.

By integrating the FDGDS into the UMHexagonS and applying new search pattern, we can improve the UMHexagonS algorithm. The experiment results show that our improvement can speed up the motion estimation time about 10% to 30% without much loss of video quality and rate-distortion performance.

The paper is organized as follow: In section II, the UMHexagonS and the FDGDS are presented. Next, we will describe two improvements will be employed in UMHexagonS algorithm. Finally, the experiment results are presented to demonstrate the efficient of proposed algorithm.

II. UMHEXAGONS AND FDGDS ALGORITHM

A. UMHexagonS Algorithm

This algorithm consists of many steps, each step uses different search pattern: unsymmetrical-cross search, small rectangular full search, uneven multi-hexagon-grid search, extended hexagon based search, diamond search. In Fig. 1, a demonstration of search procedure in UMHexagonS is presented with the search range equals 16 and the initial search point is (0, 0). The UMHexagonS algorithm also has the early termination strategy to reduce the estimation time. Fig. 2 shows each step in UMHexagonS and where the early termination is used.

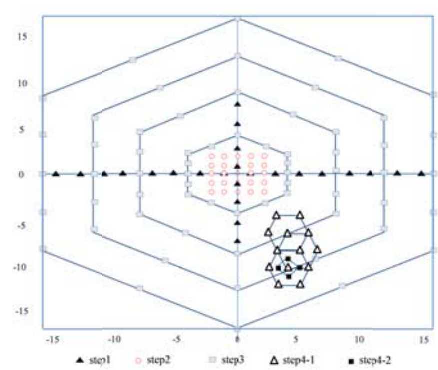

Fig.1. Search process of UMHexagonS algorithm, W = 16

There are four methods employed in UMHexagonS algorithm for prediction. They are median prediction, up layer prediction, corresponding block prediction and neighboring reference picture prediction. All these methods and the early termination are described in ([10]).

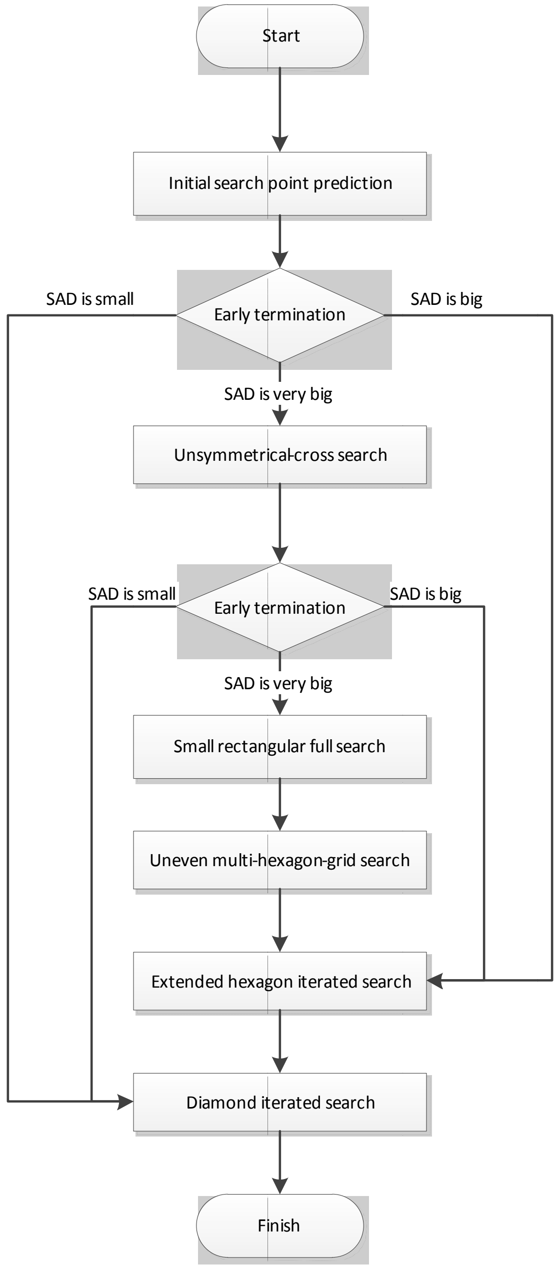

Fig. 2 Flow chart of UMHexagonS algorithm

B. *FDGDS Algorithm*

The FDGDS algorithm is based on the one-at-a-time search (OTS). In OTS, the search is continued in one particular direction until the minimum point along that direction is found. In Fig.3, an example of the first OTS-based block matching algorithm ([1]) is shown. First, the OTS is performed on horizontal direction, and then the OTS is performed on vertical direction from the minimum point found on horizontal direction.

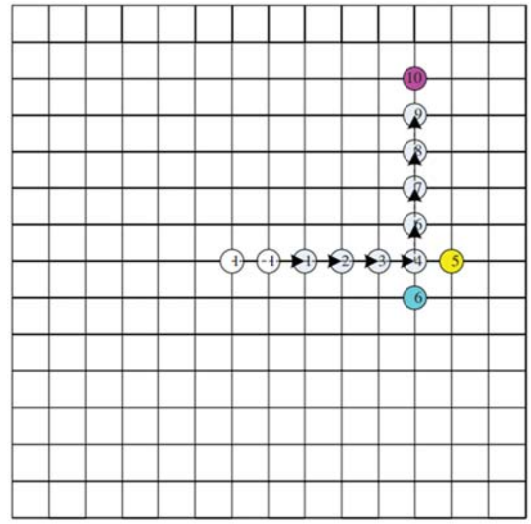

Fig. 3 Example of OTS algorithm

FDGDS uses OTS strategy on eight directions around the center to search for the minimum point. The algorithm is as follows:

Step 1: Calculate the block distortion measurement (BDM) of the search window center and set the value as CURRENT_MIN

Step 2: For each of the eight directions of the point with CURRENT_MIN:

(a) perform point-to-point directional OTS;

(b) set the minimum BDM found in the current direction as a DIRECTIONAL_MIN;

(c) if DIRECTIONAL_MIN is found, calculate the relative rate distortion (RDR) for the current direction. If RDR < T, update CURRENT_MIN with this DIRECTIONAL_MIN and repeat Step 2 (i.e., skip the remaining directional searches).

Step 3: If no point with DIRECTIONAL_MIN is found, go to Step 5; Otherwise go to Step 4.

Step 4: DIRECTIONAL_MINs are compared. The lowest one is set as CURRENT_MIN. This is the end of a search round. Go to Step 2 with updated CURRENT_MIN and its corresponding position.

Step 5: The algorithm is completed. Return with the final motion vector pointing to the position with CURRENT_MIN.

The RDR for a certain direction is calculated as follow:

$$RDR = \frac{DIRECTIONAL\_MIN}{CURRENT\_MIN}$$

According to ([11]), the threshold value T equals to 0.5 is often used because it will keep the balance between the motion estimation time and the video quality. Fig.5 demonstrates the example of FDGDS algorithm.

The FDGDS algorithm is shown to be more efficient than the UMHexagonS algorithm in motion estimation time. However, the bit rate of video sequences also increase, especially the video sequences with rapid motion scenes.

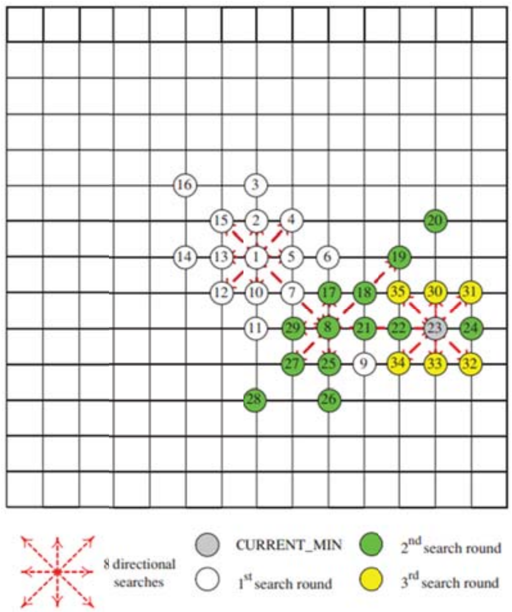

Fig.4.. Example of FDGDS algorithm

III. Proposed Algorithm

The proposed algorithm is different from the original UMHexagonS algorithm in two aspects. Firstly, instead of using unsymmetrical-cross search and small rectangular full search, the modified FDGDS is used. Secondly, the hexagon pattern in uneven multi-hexagon-grid search is replaced with a new octagon/decagon pattern to reduce the computational complexity. These two aspects are described in the following.

A. A Modified FDGDS Algorithm

In UMHexagonS algorithm, the unsymmetrical-cross search and small rectangular full search are used to find the minimum point from the initial point. But here, instead of using these two search methods, the FDGDS algorithm is used to quickly find the minimum point on this step to reduce the computational time. Moreover, in the original algorithm, early termination has been used in between unsymmetrical-cross search and small rectangular full search. In order to take advantage of the early termination, it is integrated into the FDGDS. The new FDGDS with early termination used in proposed algorithm is as follows:

Step 1: Calculate the BDM of the search window center and set the value as CURRENT_MIN

Step 2: For each of the eight directions of the point with CURRENT_MIN:

(a) perform point-to-point directional OTS;

(b) set the minimum BDM found in the current direction as a DIRECTIONAL_MIN;

(c) if DIRECTIONAL_MIN is found, try using the early termination strategy. If early termination could not be performed, calculate the RDR for the current direction. If RDR < T, update CURRENT_MIN with this DIRECTIONAL_MIN and repeat Step 2 (i.e., skip the remaining directional searches).

Step 3: If no point with DIRECTIONAL_MIN is found, go to Step 5; otherwise go to Step 4.

Step 4: DIRECTIONAL_MINs are compared. The lowest one is set as CURRENT_MIN. This is the end of a search round. Go to Step 2 with updated CURRENT_MIN and its corresponding position.

Step 5: The algorithm is completed. Return with the final motion vector pointing to the position with CURRENT_MIN.

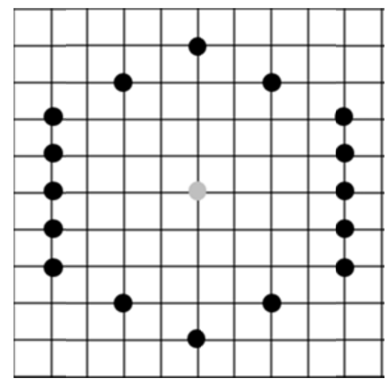

Fig.5. Hexagon pattern in UMHexagonS algorithm

B. Octagon and Decagon Pattern

The second improvement applied into the algorithm is for the uneven multi-hexagon-grid search step. Currently, this step adopts the hexagon pattern with 16 points shown in Fig.5 to look for the minimum point in large area of search window. Li Hong-ye introduces the new fast block motion estimation algorithm ([12]) that use the 8-point octagon pattern instead of 16-point hexagon pattern to reduce the search time. However, our simulation shows that Li Hong-ye's method can make the bit rate of compressed video larger than expected on video with rapid motion scenes.

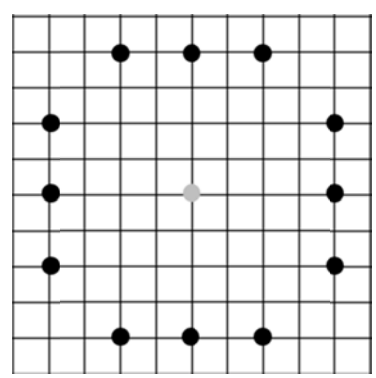

Fig.6. Octagon pattern

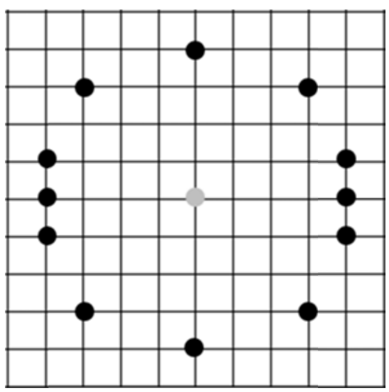

Fig.7. Decagon pattern B

Instead of using one hexagon pattern, here two different patterns are used alternatively to further reduce the computational time at this step without much bit rate increase. The first pattern is the octagon with 12 points shown in Fig.5, and the second pattern is the decagon with 12 points shown in Fig.6. In our e xperiment, using these two alternatively will help cover most points in the search window, meanwhile the number of search points each round is reduced from 16 to 12. When performed the search at this stage, two patterns will be applied alternatively. First, the octagon pattern is used, next is the decagon pattern, and so on. The whole search process of proposed algorithm is shown in Fig.8

IV. EXPERIMENT RESULTS

The proposed algorithm is implemented in JM Reference Software 9.6 with the following configurations: frame sequence is IPPPIPPP…, sub-pel motion estimation is disabled, one r eference frame, Quantization Parameter value 28, RD Optimzation is turne d off. Video sequences: Akiyo, News, Coastguard, Foreman, Stefan and Container are tested on the first 100 frames. The reference software is running on Windows operating system with the Intel Core i5-2410M processor. The proposed algorithm is compared with the UMHexagonS algorithm in three different aspects: average PSNR (dB), average Motion Estimation time (s) and ave rage bit rate (kbit/s). The experiment results are shown in Table I.

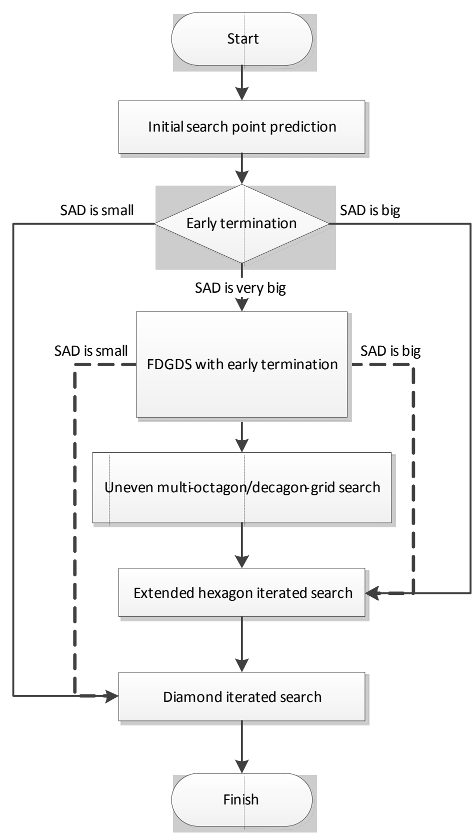

Fig.8. Search process of proposed algorithm

It can be seen that the proposed algorithm runs faster than UMHexagonS in all video sequences, especially in video sequences with rapid motion scenes such as Coastguard or Stefan. The bitrate of new algorithm is not much greater than the original algorithm with less than 0.5% increase, and it has even smaller bitrate in some videos such as Akiyo, Coastguard and Container. The PSNR does not change in most of the video sequences, except the Foreman with 0.01dB lost.

V. CONCLUSIONS

In this paper, a new algorithm based on the FDGDS and UMHexagonS algorithm is proposed. It takes advantage of the fast running time from FDGDS and accuracy from UMHexagonS to help reduce the computational complexity while maintaining the video quality and bitrate. And two new search patterns, octagon and decagon, are also presented to further reduce the motion estimation time. This new algorithm is shown to run faster than original UMHexagonS without much of PSNR lost and bitrate increase. It especially works well with videos with complex motion contents.

REFERENCES

[1] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. 33, no. 8, pp. 888-896, Aug. 1985.

[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, Dec. 1981, pp. G5.3.1-G5.3.5.

[3] R. Li, B. Zeng, and M. L. Lio, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438-443, 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, no. 3, pp. 313-317, Jun. 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, no. 4, pp. 419-422, 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, no. 4, pp. 369-377, Aug. 1998.

[7] C. H. Cheung and L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168-1177, Dec. 2002.

[8] 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, no. 5, pp. 349-355, May 2002.

[9] C. H. Cheung and L. M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 16-22, Feb. 2005.

[10] Z. B. Chen, P. Zhou, and Y. He, "Fast integer pel and factional pel motion estimation for JVT", in *Proc. 6th Meeting: JVT-F017*, Awaji, Island, Japan, 2002.

[11] Lai-Man Po, Ka-Ho Ng, Kwok-Wai Cheung, Ka-Man Wong, Yusuf Md. Salah Uddin, and Chi-Wang Ting, "Novel Directional Gradient Descent Searches for Fast Block Motion Estimation", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, August 2009.

[12] Li Hong-ye, Liu Ming-jun and Zhang Zhi-qiang, "A New Fast Motion Estimation Algorithm Based on H.264," *International Conference on Multimedia Information Networking and Security*, vol. 1, pp. 287-290, Nov. 2009.