The University of Texas at Austin

Department of Electrical and Computer Engineering

EE 381K: Multidimensional Digital Signal Processing

Course Project Final Report

New Motion Estimation Algorithm Used in H.263+

Ying, Lu , Yue Chen, Yu Wang

Department of Electrical and Computer Engineering

Email:Luying@ece.utexas.edu, chenyue@ece.utexas.edu, etctowy@mail.utexas.edu

Abstract

The motion estimation algorithm is one of the most important issues in the video coding standards such as ISO MPEG-1/2 and ITU-T H.263. Recently developed fast motion estimation algorithms [4,5,6] greatly reduced the computational complexity, which is critical for real time implementations. In this report, a new fast motion estimation algorithm is proposed. The experimental results show that the new algorithm can greatly reduce the number of average searching points per marcoblock as well as the computation complexity, while still maintaining an acceptable signal to noise ratio.

Keywords: Motion Estimation, H.263+, Video Coding, Full Search, Diamond Search

I. Introduction

Video compression is a critical component when transmitting or storing digital video sequences. Hybrid video compression, exploits temporal redundancy by motion compensation and spatial redundancy by DCT transformation, and it has been widely adopted by H.261, H.262, H.263 and MPEG 1/ MPEG 2 international standards.

Motion estimation is a major part in such video compression systems. To facilitate hardware and software implementation, macroblock matching motion estimation is deployed in standard –based video compression. Several motion estimation criteria have been advised for block matching search. In practice, the widely adopted motion search criteria is the sum of absolute difference (SAD).

The selection of searching window size demands balanced consideration of application type, frame rate and processing power. Given a search window, search methods to obtain motion vectors greatly affect the computation requirement. The exhaustive motion estimation search covers all candidate blocks in the search window to find the best match. However its high computational complexity makes it impractical for real time applications. Many schemes have been invented to reduce the complexity of motion estimation that is associated with the exhaustive search. Some reduce the calculation of cost function in down-sampling space [5,7], others reduce the total searching points, such as Three-step search, 2D-Logarith search, Parallel hierarchical one dimensional search. One drawback to these methods is that the above algorithms are based on the following assumption: the matching error decrease monotonically as the motion vector moves closer to the global optimum. Unfortunately, this assumption does not always hold true especially in slow-moving video sequences, which have many local minima in each search window.

In this report, we propose a new fast motion estimation algorithm, which prevents finding the local minimum point as the best match. The algorithm also reduces the computation of the cost function and decreases searching points. The tradeoff between the compression efficiency and the image quality is well satisfied.

II. Proposed Algorithm

In the proposed algorithm, we first make full use of the motion vectors of the previous frame as the reference vector (scaled ½), which can be regarded as the starting search center of the current searching macroblock. The algorithm is center-biased. The advantage of this step is similar to that of telescopic search. Then in each search window (16×16), we adopt the diamond search. The diamond search algorithm (Figure 1) is described as follows:

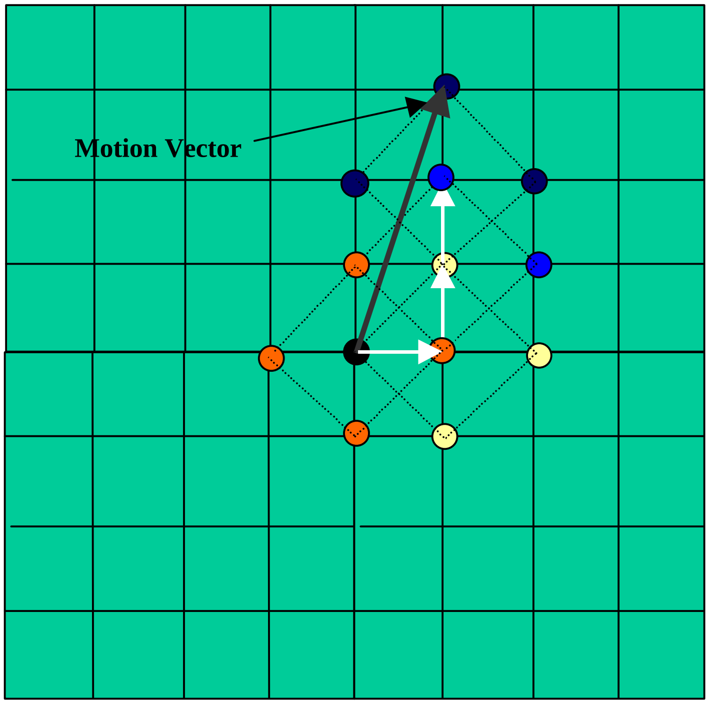


Figure 1. Diamond Search

1) First, we obtain the start searching center point based on the above method. Choose this point as center point (x, y), which has 8 neighboring points. Compute the SAD value of four points (x+1,y), (x-1, y), (x, y+1), (x, y-1). The common SAD formula is as follow:

$$SAD(k,l;u,v)=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}|B_{i,j}(k,l)-B_{i-u,j-v}(k,l)|$$

Where, $B_{i,j}(k,l)$ represents the $(k,l)^{th}$ pixel of a 16x16 macroblock from the current picture at the spatial location (i,j). $B_{i-u,j-v}(k,l)$ represents the $(k,l)^{th}$ pixel of a candidate macroblock from a reference picture at the spatial location (i,j) displaced by the vector (u,v).

In fact, redundancy still exists between the pixels within each macroblock (16×16) as the difference between two adjacent points is always small. By doing down sampling along the horizontal direction with a rate of 2, the computation complexity of SAD decreases to 50% of that of the normal SAD. The proposed SAD formula is:

$$SAD(k,l;u,v)=\sum_{k=0}^{M/2-1}\sum_{l=0}^{N-1}|B_{i,j}(2k,l)-B_{i-u,j-v}(2k,l)|$$

Compare the four SAD values and save the minimal one in a variable D1.

2) Choose the point with minimal SAD value in step 1) as the center point and use the similar method as above to find point with a new minimal SAD value among its four neighboring points. Move D1 to the variable D2 and save the minimal SAD value to D1.

3) Choose the point with minimal SAD value in step 2) as the center point. Again compute its four neighbors' SAD values and find one with minimal SAD value. Move D2 to the variable D3 and D1 to D2 and save the new minimal SAD value to D1. If the conditions of both D3 <= D1 and D2 <= D1 are satisfied, the search procedure stop. Otherwise, go to step 4.

4) Adopting the point with the latest minimal SAD value as the center point, compute its four neighbors' SAD values and acquire a new minimal SAD value. Move D2 to the variable D3 and D1 to D2 and save the new minimal SAD value to D1. If the conditions of both D3 <= D1 and D2 <= D1 are satisfied, the search procedure stop. Otherwise, repeat step 4.

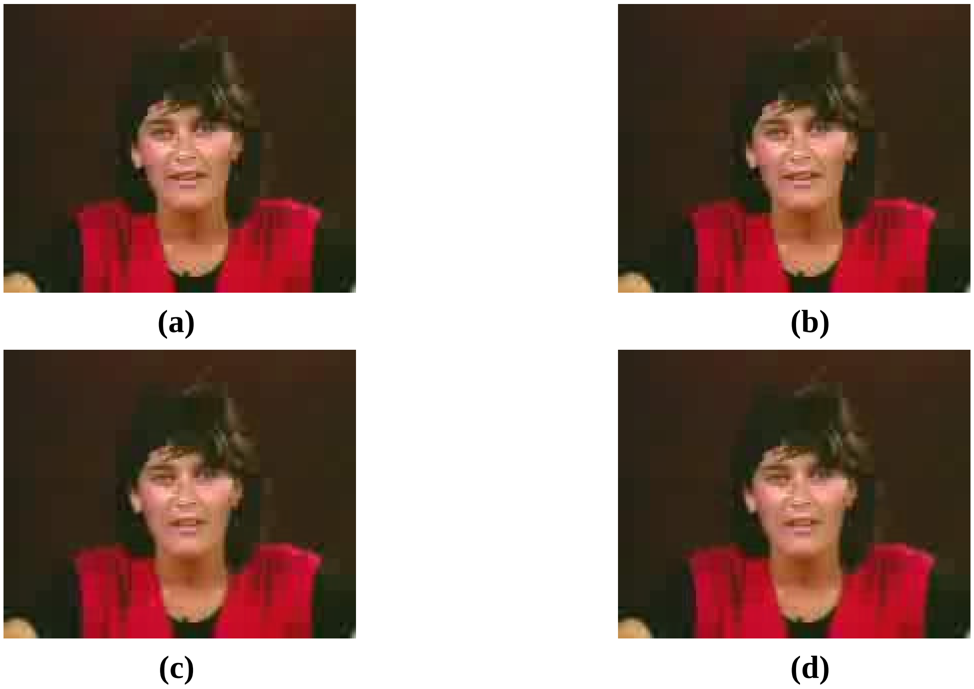
III. Experimental Result

The fast algorithm adopted in this project is based on the H.263+ Encoder/Decoder from the Signal Processing and Multimedia Group at the University of British Columbia, Canada. By encoding two kinds of image sequences (one is Miss American-Figure 2, another is Foreman-Figure 3), it shows that the four strategies implemented in this project contribute greatly to simplify the computation complexity and the image quality can still be kept in an acceptable level (SNR is 0.2dB (Figure 4 and Figure 5) lower than that of the full search).

1) By down sampling the image in the horizontal direction, we save half of the overhead in SAD computation, compared to the common SAD computation.
2) Using partial distortion, we can further decrease the computation complexity in our SAD computation.
3) Using the diamond-searching algorithm, we can achieve average searching points to 20-points/per macroblock (Figure 6 and Figure 7), compared with 256-points/per macroblock in the full search algorithm.
4) Using the previous motion vector to determine the current starting center point in each macroblock (Step 1 in our algorithm) produces a good initial starting center point.

IV. Conclusion

1) Our algorithm can decrease the computation complexity sharply, it is 25 times faster than Full Search algorithm.
2) By testing 33 frames in two kinds of QCIF image sequences, for Miss America image sequences, its compression efficiency is near 60:1, for Foreman image sequences, its compression efficiency is near 47:1.
3) The image quality of our encoder can match that of Full Search encoder. Testing two kinds of image sequences shows that the average SNR in our algorithm is less 0.2 dB greater than that of Full Search algorithm.

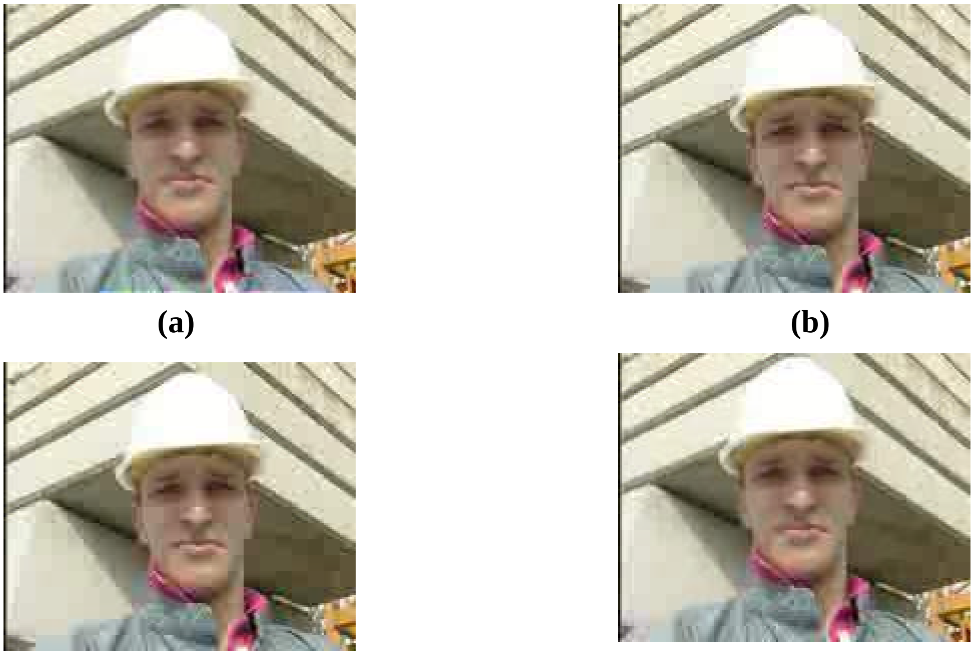


(a) **(b)**

(c) **(d)**

Figure 2. Miss America image sequences

(a) and (c) - full search, (b) and (d) – our algorithm



(a) **(b)**

(c) **(d)**

Figure 3. Foreman image sequences

(a) and (c) - full search, (b) and (d) – our algorithm

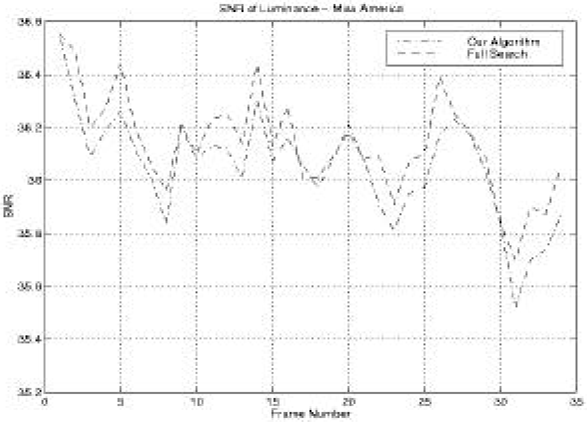


Figure 4. SNR of Miss America

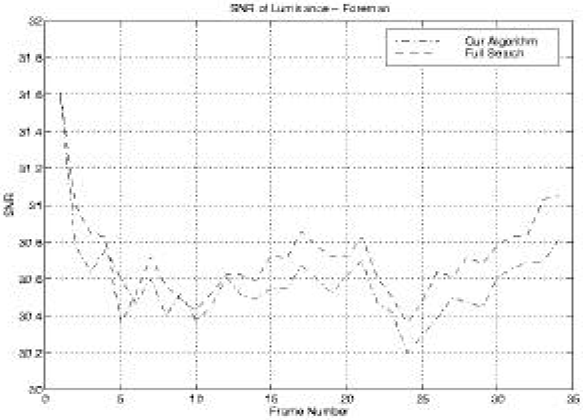
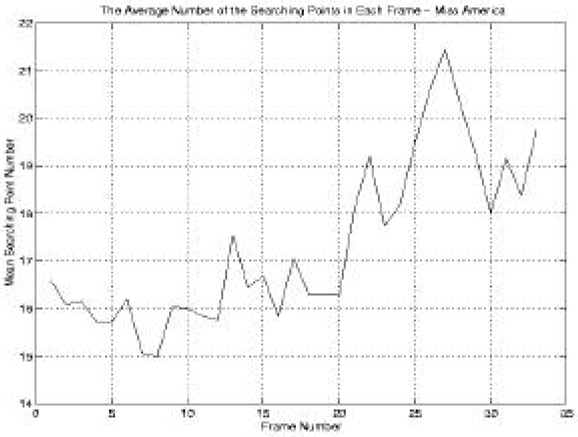


Figure 5. SNR of Foreman



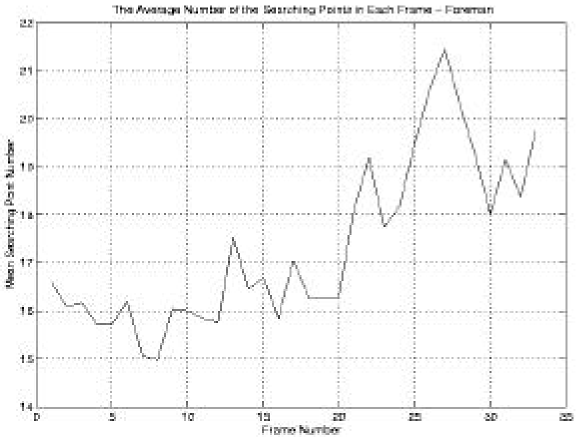


Figure 6. The average searching points of Miss America

Figure 7. The average searching points of Foreman

Reference

[1] T. Koga, K. Iinuma, A. Hirano, Y Iijima, and T. Ishiguro, "Motion compensated interframe image coding for video conference," *Proc. NTC81*, pp. G5.3.1, Nov. 1981.

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

[3] M. Bierling, "Displacement estimation by hierarchical block-matching," in *Proc. SPIE Conf. on Vis. Commun. And Image Proc.*, vol. 1001, pp. 942-951, Jan. 1988.

[4] C. K. Cheung and L. M. Po, "A hybrid adaptive search algorithm for fast block motion estimation," *IEEE Int. Symp. Signal Proc. And Its Appl.*, vol. 1, pp. 365-368, Aug. 1996.

[5] C. K. Cheung and L. M. Po, "A hierarchical block motion estimation algorithm using partial distortion measure," *IEEE Int. Conf. on Image Processing*, vol. III, pp. 606- 609, Oct. 1997.

[6] Kai Sun, "Adaptive Motion Estimation Based on Statistical Sum of Absolute Difference," *IEEE Int. Conf. on Image Processing*, vol. III, pp. 601-604, Oct. 1998.

[7] Alan C. K. Ng and Bing Zeng, "A New Fast Motion Estimation Algorithm Based on Search Window Sub-sampling and Object Boundary Pixel Block Matching," *IEEE Int. Conf. on Image Processing*, vol. III, pp. 605-608, Oct. 1998.

[8] Yuen-Wen Lee, Faouzi Kossentini and Rabab Ward, "Efficient RD Optimized macroblock coding mode selection for MPEG-2 video encoding," *IEEE Int. Conf. on Image Processing*, vol. II, pp. 803-806, Oct. 1997.

[9] Bing Zeng, Renxiang Li and Ming L. Liou, "Optimization of Fast Block Motion Estimation Algorithms," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, No. 6, pp. 833-844, Dec., 1997.

[10] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *Trans. Circuits and Systems for Video Technology*, vol. 6, No. 3, pp. 313-317, June, 1996.

[11] Xiaobing Lee and Ya-Qin Zhang, "A Fast Hierarchical Motion Compensation Scheme for Video Coding Using Block Feature Matching," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, No. 6, pp. 627-635, Dec., 1996.