

ADVANCED BLOCK MATCHING MOTION ESTIMATION ALGORITHM FOR VIDEO COMPRESSION

MR. A. J. TANKARIYA¹, PROF. MUKESH TIWARI², PROF. JAIKARAN SINGH³, DR. ANUBHUTI KHARE⁴

¹Department of electronics and communication engineering, SSSIST- Sehore,
Rajiv Gandhi Proudhyogiki Vishwavidhyalaya, University of M.P. (India)

²Associate Professor, Department of electronics and communication engineering, SSSIST- Sehore,
Rajiv Gandhi Proudhyogiki Vishwavidhyalaya, University of M.P. (India)

³Associate Professor, Department of electronics and communication engineering, SSSIST- Sehore,
Rajiv Gandhi Proudhyogiki Vishwavidhyalaya, University of M.P. (India)

⁴ Associate Professor, Department of electronics and communication engineering, University Institute of Technology,
Rajiv Gandhi Proudhyogiki Vishwavidhyalaya, university of M.P. (India)

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. Developed fast motion estimation algorithms [4, 5, 6] greatly reduced the computational complexity, which is critical for real time implementations. Here, 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 macroblock as well as the computation complexity, while still maintaining an acceptable peak signal to noise ratio.

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

1. 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 also in compression standards, such as MPEG-4 and H.264/AVC, due to its simplicity and effectiveness. Motion estimation is a major part in such video compressionsystems.

The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame. The idea behind block matching is to divide the current frame into a matrix of 'macro blocks' that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to p pixels on all four sides of the corresponding macro block in previous frame. This 'p' is called as the search parameter. Larger motions require a larger p and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macro block is taken as a square of side 16 pixels, and the search parameter p is 7 pixels. The idea is represented in Fig1.

The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block. There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (i). Another cost function is Mean Squared Error (MSE) given by equation(ii).

$$MAD = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (i)$$

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (ii)$$

Where MxN is the size of the macro block, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively.

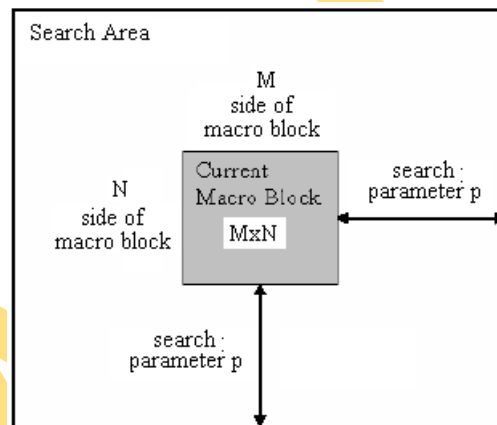


Figure 1. A macro block

The best matched block can be obtained by using the full search algorithm (FSA) which matches all possible candidates within the search window. However, due to its heavy complexity, the FSA has a significant performance problem in real-time applications. To solve this problem, fast BMAs like new three-step search (N3SS) and diamond search (DS) have been developed. However, these traditional BMAs easily trapped into local minima since they limit the searchpoints.

To facilitate hardware and software implementation, macro block matching motion estimation is deployed in standard –based video compression. Several motion estimation criteria other than above have been advised for block matching search. In practice, the SAD criterion is often used as the criterion for choosing the best-matching block in the reference frame due to its simplicity and good performance. So the widely adopted motion search criterion 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, and 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 paper, 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.

2. PROPOSEDALGORITHM

In the proposed algorithm, we first make full use of the motion vectors of the previous frame as the reference vector (scaled 1/2), which can be regarded as the starting search center of the current searching macro block. The algorithm is center-biased. The advantage of this step is similar to that of telescopic search. Then in each search window (16x16), we adopt the diamond search. The diamond search algorithm (Figure 2) is described as follows:

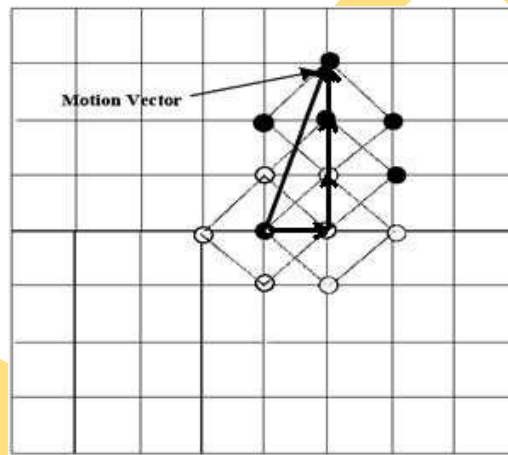


Figure 2. Diamond Search

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 SAD between the reference block and the current macro-block is defined as below and its common SAD formula is asfollow:

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

Where, $B_{ij}(k, l)$, represents the (k, l)th pixel of a 16x16 macro block from the current picture at the spatial location (i, j). And $B_{i-u, j-v}(k, l)$ represents the (k, l)th pixel of a candidate macro block 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 macro block (16x16) 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 formulais:

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

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

- 2) Choose the point with minimal SAD value in first step 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 second step 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 $D3 \leq D1$ and $D2 \leq D1$ 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 $D3 \leq D1$ and $D2 \leq D1$ are satisfied, the search procedure stop. Otherwise, repeat step 4.

3. EXPERIMENTAL RESULTS

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 3, another is Foreman-Figure 4), it shows that the four strategies implemented in this report contribute greatly to simplify the computation complexity and the image quality can still be kept in an acceptable level as shown in Figure 5 and Figure 6, PSNR is 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 macro block (Figure 7 and Figure 8), compared with 256-points/per macro block in the full search algorithm.
- 4) Using the previous motion vector to determine the current starting center point in each macro block (Step 1 in our algorithm) produces a good initial starting center point.

4. CONCLUSION

- 1) This 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 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 PSNR in our algorithm is less 0.2 dB greater than that of Full Search algorithm.

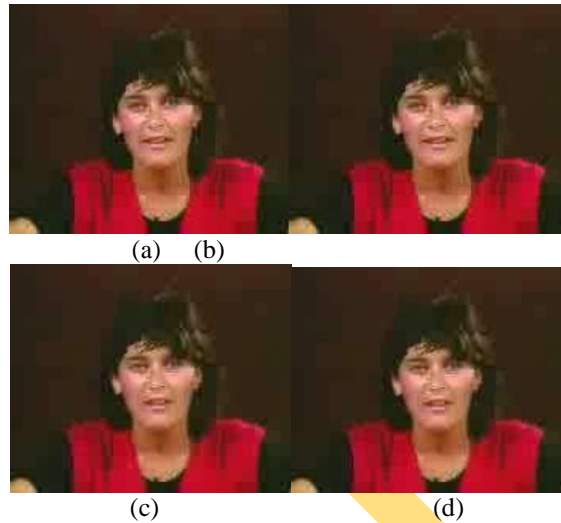


Figure 3. Miss America image sequences (a) and (c) - full search, (b) and (d) – proposed algorithm

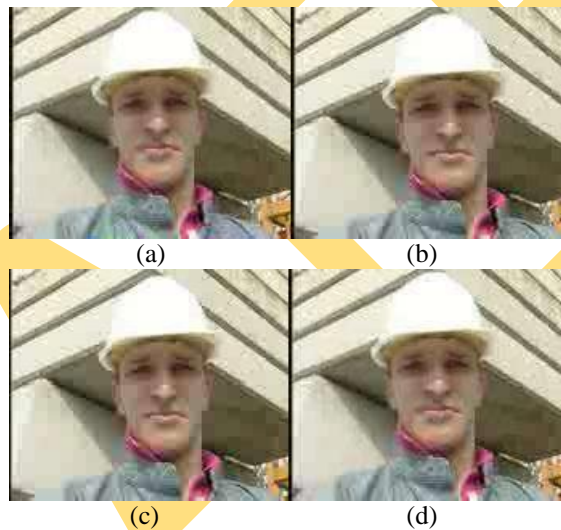


Figure 4. Foreman image sequences (a) and (c) - full search, (b) and (d) – proposed algorithm

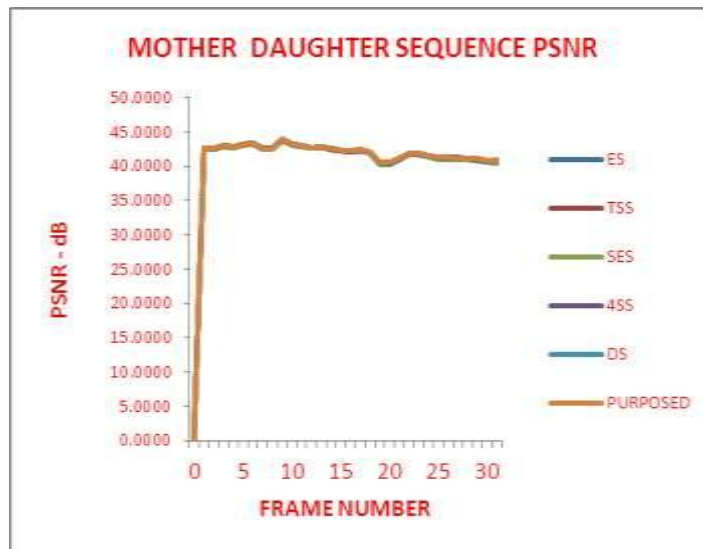


Figure 5. PSNR of Miss America

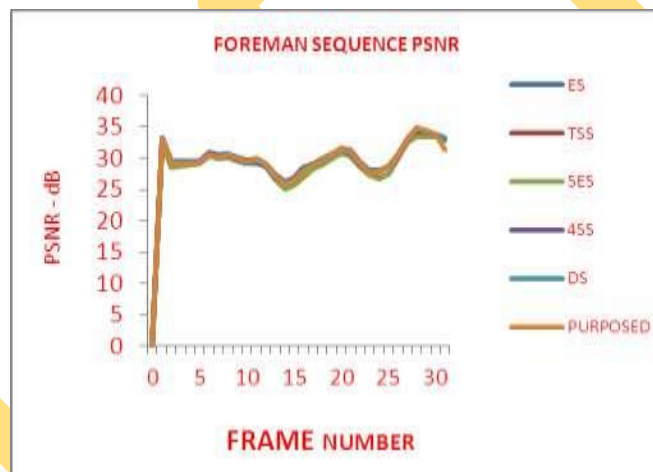


Figure 6. PSNR of Foreman

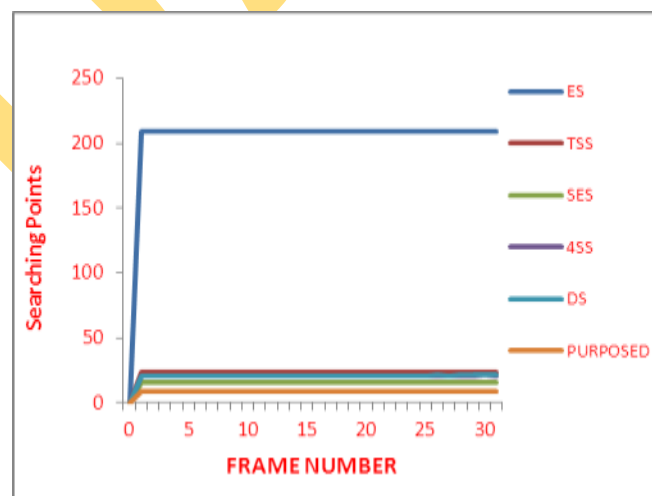


Figure 7. The average searching points of Miss America

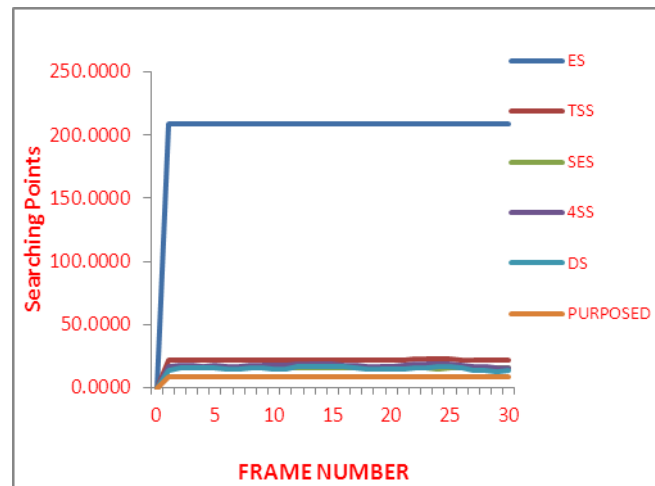


Figure 8. The average searching points of Foreman

5. REFERENCES

- [1] T. Koga, K. Inuma, 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 Subsampling and Object Boundary Pixel BlockMatching," 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 macro block 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.