An Improved Fast full Search Algorithm for Motion Estimation

Mr. Aliraja Vasaya[1] Prof. Kirit Rathod[2] Dr. K. H. Wandra[3]
[1]P. G. Student [2]Asst. Prof. [3]Principal
[1,2]Department of Computer Engineering
[1, 2, 3] C. U. Shah Engineering College, Wadhwan

Abstract— *Motion estimation and motion compensation is a predictive technique for exploiting the temporal redundancy between successive frames of video sequence. Block matching techniques are widely used motion estimation method to obtain the motion compensated prediction. By splitting each frame into macro blocks, motion vector of each macro block is obtained by using block matching algorithm (or motion estimation algorithm). In order to get motion vector of each macro block, the most obvious and simplistic method is full search algorithm. All possible displacements in the search window are evaluated using block-matching criteria. The advantage of full search is that we can find the absolute optimal solution. However, its high computational complexity makes it impossible for real-time implementation. Because the computational complexity of video compression, the compression efficiency and the compression quality is determined by the motion estimation algorithm, development of Fast Motion Estimation Algorithm for real-time application becomes compelling.*

Keywords—Algorithm, Motions, Three Step Search, Four Step Search, Diamond Search, Adaptive Root Search, Mean Absolute Difference, Mean Squared Error, Sum of Absolute Differences, Peak-Signal-to-Noise-Ratio, Large Diamond Search Pattern, Small Diamond Search Pattern.

I. INTRODUCTION

The temporal prediction technique used in MPEG video is based on motion estimation. The basic premise of motion estimation is that in most cases, consecutive video frames will be similar except for changes induced by objects moving within the frames. In the trivial case of zero motion between frames and no other differences caused by noise, etc.., it is easy for the encoder to efficiently predict the current frame as a duplicate of the prediction frame. When this is done, the only information necessary to transmit to the decoder becomes the syntactic overhead necessary to reconstruct the picture from the original reference frame. When there is motion in the images, the situation is not as simple. Another goal of Motion Estimation is to reduce the total amount of bits required for transmission or storage of the frames of an image sequence. Motion compensated image sequence coding primarily focuses on the reduction of the high temporal correlation of the signal to be stored or transmitted. To that end, motion information has to be extracted from the sequence in order to relate locations in consecutive frames that correspond in gray level. This motion is represented by so called corresponding or displacement vectors. Motion estimation techniques can be divided into four main groups.

- Gradient techniques
- Pel recursive techniques
- Block matching techniques
- Frequency based techniques

Gradient techniques have been developed for image sequence analysis applications. They solve the optical flow and results in a dense motion field. Both Pel recursive and block matching techniques have been developed in the framework of image sequence coding. Pel recursive techniques can be considered as a subset of gradient techniques. However as they constitute as important contribution in the field of coding, they are considered as a separate group. Block matching techniques are based on the minimization of a disparity measure. They are most widely used in coding applications. In block matching techniques, a block in the current picture is matched with block of previous picture and hence motion is calculated. Finally, frequency-domain techniques are based on the relationship between transformed coefficients (e.g., Fourier transform) of a shifted image. However they lack widespread use, especially in the field of image sequence coding and are not popular.

II. MOTION VECTOR AND VIDEO ANALYSIS

To define the motion from frame to frame, the motion vectors in the present frame are subtracted from the vectors in the previous frame. However, depending on what type of frame (I, P, or B) is in present and past, not all of the arrays can be used. An explication of this is given below. A vector defines a distance and a direction, it does not define a position. We have to know the vectors initial position (reference point) to define all the motion from frame to frame. Only vectors with the same reference point can be subtracted from each other. To illustrate, let's take the simple example of "x" moving across a portion of the screen as shown in Figure below.

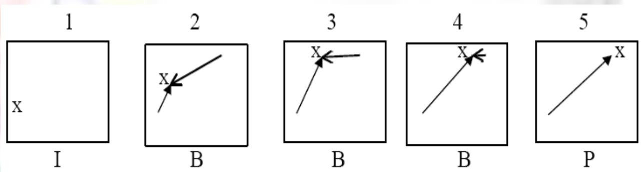


Fig. 1: Motion vectors associated with a moving picture.

The arrows represent a forward vector and represent a backward vector [Note a forward vector does not have to be pointing forward, and a backward vector pointing backward. It is just the naming convention for whither the reference frame in the past (forward) or future (backward)].
The values for the vectors are given below:
In the first frame there are no motion vectors. In this example the transition from frame 1 to frame 2 can be calculated as before.
If the second transition is calculated as before we get:
Forward motion:
(4, -6) - (2, -3) = (2, -3)
Backward motion:
(-5, 1) - (0, 0) = (-5, 1)
Total motion = (-1.5, -1)
This result is incorrect.

To get the correct result, only the forward motion can be used. Similarly only the backward motion is used for the third transition. The motion for the final transition cannot be found because there is only a backward vector in frame 4 and only a forward vector in frame 5. Only similar types of vector can be subtracted from each other.

Below are further rules to complement the rules that were established in table

- Only if there is a similar type of vector (forward, backward or both) present in both frames can the motion be found.
- A reference frame is said to have all vectors equal to (0, 0)
- If there is a skipped macro-block in the present P frame, there is zero motion for that transition.
- If there is a skipped macro-block in the previous P frame, the motion for that transition can't be calculated. An exception to this is if there is also a skipped macro-block in the present P frame in which case the motion will be zero.
- If there is an Intra macro-block in either the present or previous frame, the motion for that transition can't be calculated.

III. PROPOSED ALGORITHM FOR MOTION ESTIMATION

Three step search (TSS), Four Step Search (FSS), Diamond Search (DS) and Adaptive Root Search (ARS) are implemented and the comparison of average number of search for macro-block computed by different algorithms is shown in figure The graph clearly indicates that Adaptive root search has minimum average number of search for macro-block and three step search has maximum average number of search for macro-block. TSS takes much time to search macro-block

Now in comparison with full search algorithm, TSS, FSS, DS and ARS are better in macro block comparison and takes less time to search macro-blocks but full search algorithm find accurate macro-block than all others.

A. *Peak-Signal-to-Noise-Ratio (PSNR)*

PSNR ratio of all algorithms is shown in figure and from that graph it is clear that PSNR ratio of TSS is very less compared to all other algorithms and full search algorithm having very high PSNR ratio.

B. *Implementation of Proposed Algorithm*

As with the proposed algorithm we have implement Full Search (FS) using macro block. In which we have taken random images from sample video and algorithm is as shown below

Step 1: These FS algorithm searches the entire search region for a block such that the BDM (Block distortion measure) is a global minimum.

Step 2: If more than one block generates a minimum BDM (Block distortion measure), the FS algorithm selects the block whose motion vector has the smallest magnitude

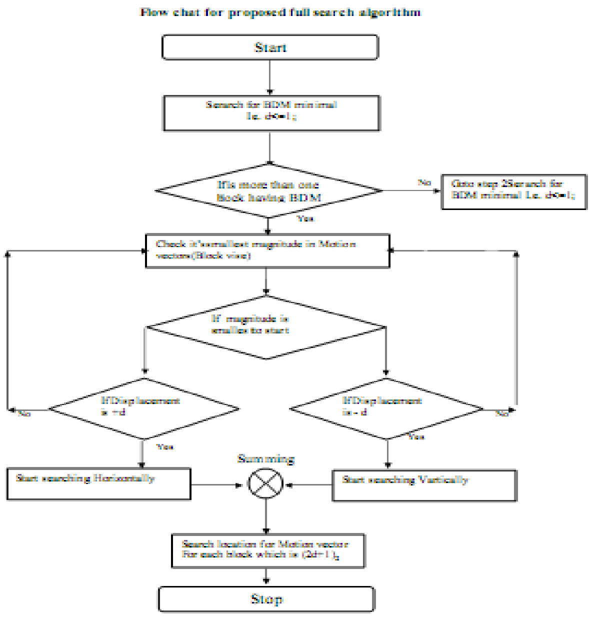


Fig. 2: Flow chart for proposed algorithm for FFSAME

Step 3: To achieve this, checking points are used in a spiral trajectory starting at the centre of the search region.
Step 4: If the maximum displacement of a motion vector in both the horizontal and vertical directions is + d or -d pixels
Step 5 & 6: The total number of search points used to locate the motion vector for each block can be as high as (2d+1)2.

This implementation we have used macro block with 16*16 and 8*8. Also we have implemented by changing parameter of fromas that is 7 and 10. For proper and efficient result we have implemented by using more than 200 images for what we got different results by comparing the results or our proposed algorithm and other known algorithms to make simple to identify this result with our results we have given different colour to each known algorithms for better identification and comparison.

The details as results analysis are as follows.

All title and author details must be in single-column format and must be centered. Every word in a title must be capitalized. Email address is compulsory for the corresponding author.

Fig. 1

Fig. 2

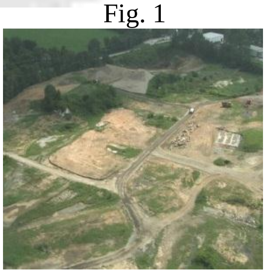


Fig. 3

Fig. 4

IV. Result Analysis

Result shown with block size 16*16 with parameter 7

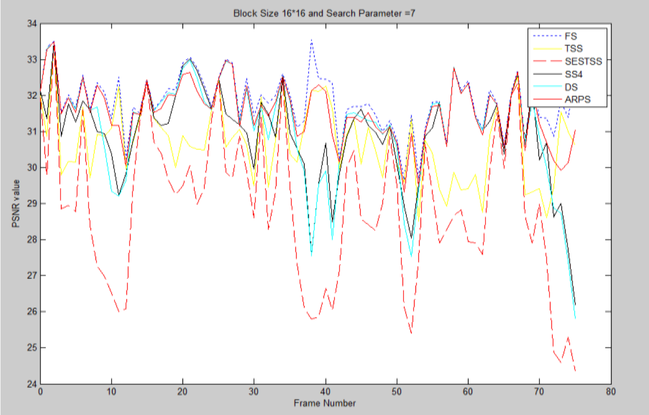
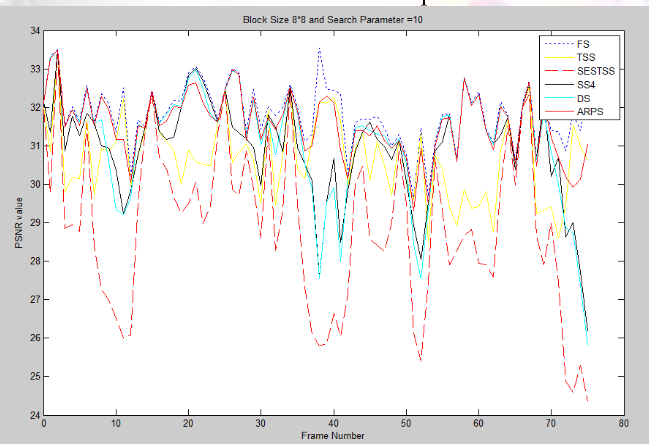


Fig. 3: Graph of proposed system

Here also we have shown the comparative graph which we have shown using some other Images and all the required results as well. This comparative study of our proposed algorithm gives us a result that it searches fast and very efficiently for even 250 or more images. Here we have shown our result with 16*16 block size and search parameter 7. Under which we have computed more than 80 images were but among all existing algorithm our proposed algorithm have performed well and searched very fast as shown in the above graph(shown in blue dotted line).

Result shown with block size 8*8 with parameter 10



V. Conclusion

This work has explored the theory of all motion estimation algorithms and examined basic features of motion estimation algorithms.

Even though more commonly linked to lossy video compression, motion estimation is in fact a technique that goes beyond and allows for video processing and computational vision algorithms and applications. It allows a computer to detect movement as well as to perform comprehensive video sequence analysis, identifying scenes, and camera and object movements. Motion estimation is one technique that allows for a simple, yet effective, object identification scheme. Five different algorithms for motion estimation are tested and compared Average number of search and PSNR ratio.

It is concluded that fast full search(FS) motion estimation algorithm gives better result compared to all other algorithms but takes somewhat more time to search particular macro-blocks because it compare with each and every macro-blocks in search area.

VI. Future Scope

In future this algorithm can be further improved by developing new search technique so that it can take less time to search macro-blocks and by changing some size of macro-blocks, quality of motion can also be further improved.

References

[1] CCITT SG XV, "Recommendation H.261—Video codec for audiovisual services at p*64 Kbits/s," Tech. Rep. COM XV-R37-E, Aug. 2013.

[2] MPEG, "ISO CD 11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 M bits/s," Nov. 2010.

[3] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecomm. Conf. New Orleans, pp. G5.3.1–G5.3.5.

[4] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Comm.., vol. COM-29, pp. 1799–1808, Dec. 2001.

[5] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," IEEE Trans. Comm., vol. COM-33, pp. 888–896, Aug. 2005.

[6] S. Kapagantula and K. R. Rao, "Motion predictive interframe coding," IEEE Trans. Comm, vol. COM-33, pp. 1011–1015, Sept. 20011.

[7] Puri, H. M. Hang, and D. L. Schilling, "An efficient block-matching algorithm for motion-compensated coding," in Proc. Int. Conf. Acoust., Speech, Signal Processing, 2007, pp. 25.4.1–25.4.4.

[8] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., vol. 38, pp. 950–953, July 2010.

[9] H. Hsieh, P. C. Lu, J. S. Shyn, and E. H. Lu, "Motion estimation algorithm using interblock correlation," Electron. Lett., vol. 26, pp. 276–277, Mar. 2010.

[10] Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Syst. Video Technol., vol. 3, pp. 148–157, Apr. 2003.

[11] L. W. Lee, J. F. Wang, J. Y. Lee, and J. D. Shie, "Dynamic search- window adjustment and interlaced search for block-matching algorithm," IEEE Trans. Circuits Syst. Video Technol., vol. 3, pp. 85–87, Feb.2003.

[12] M. J. Chen, L. G. Chen, and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," IEEE Trans. vol. 4, pp. 504–509, Oct. 2013.

[13] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new

contribution,"Proc. IEEE, vol. 83, pp. 858–876, June 2005.

[14] K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," IEEE Trans. Circuits Syst., vol. 36, pp. 1317–1325, Oct. 2009.

[15] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Parallel architectures for 3-step hierarchical search block-matching algorithm," IEEE Trans. Circuits Syst. Video Technol. vol. 4, pp. 407–416, Aug. 2013.

[16] J. S. Kim and R. H. Park, "A fast feature-based block matching algorithm using integral projections," IEEE J. Select. Areas Comm., vol. 10, pp. 968–971, June 2012.

[17] Y. H. Fok and O. C. Au, "An improved fast feature-based block motion estimation," in Proc. IEEE 1994 Int. Conf. Image Processing, 1994, pp.741–745.

[18] H. B. Park and C. Wang, "Image compression by vector quantization of projection data," IEICE Trans. Inform. Syst., vol. E75-D, pp. 148–155, Jan. 1992.

[19] K. H. Jung and C. Wang, "Projective image representation and its application to image compression," IEICE Trans. Inform. Syst., vol. E79-D, pp. 136–142, Feb. 2006.

[20] Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," IEEE Trans. Comm.., vol. COM-33, pp. 1132–1133, Oct. 2005.

[21] S. H. Huang and S. H. Chen, "Fast encoding algorithm for VQ-based image coding," Electron. Lett. vol. 26, pp. 1618–1619, Sept. 1990.

[22] C. H. Lee and L. H. Chen, "A fast search algorithm for vector quantization using mean pyramids of code words," IEEE Trans. Comm., vol.43. pp. 1697–1702, Feb. 20010