International Journal of Electrical Electronics & Computer Science Engineering
Volume 6, Issue 2 (April, 2019) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

Investigation and Analysis of Different Fast Block Matching Motion Estimation Algorithms for Video Compression

Rajender Kumar[1], Krishan Kumar[2], Seema Jangra[3]
[1-2]Assistant Professor, Department of ECE, BPS Mahila Vishwavidyalaya, Sonepat, Haryana, India
[3]Assistant Professor, Department of Computer Science, IP College for Women, Delhi,, India
[1]rajseebpsmv14@gmail.com , [2]krishan.bpsmv@gmail.com.com, [3]seema.mtech@gmail.com

Abstract:* With advancement in video processing applications such as Digital TV, Internet Streaming Video, Videophones, Multi-media Communication and Video conferencing in last decade, one of the most essential components of video is its compression which is used for efficient storage and transmission of video signal. Video contains temporal redundancy between two successive frames which can be exploited to improve coding efficiency. One of technique used to remove temporal redundancy is motion estimation. A number of motion estimation techniques are used, out of which block matching motion estimation is most popular and proved efficient in terms of quality and bit rate and adopted in many video coding standards. Many fast block matching algorithms have been developed to improve processing speed, visual image quality and power consumption. This paper explores recently developed fast block based motion estimation algorithms such as Three Step Search (TSS), Simple and efficient search (SES), Diamond Search (DS) and most recently developed Adaptive Rod Pattern Search (ARPS) and presents the computational and performance trade-offs involved in preferring a motion estimation algorithm for video coding applications. These block based motion estimation algorithms are compared and implemented for different video sequences with different types of motion in terms of PSNR and Computational complexity. It is proved from results that Adaptive Rod Pattern Search (ARPS) achieves the best trade-off between PSNR and Computational complexity for different video sequences with slow, medium and fast motion activity.

***Keywords:* Video, Motion Estimation, Block Matching, Search Points.**

I. INTRODUCTION

With an increase in internet bandwidth and CPU performance, video compression becomes a state-of-art technology with one of active field research in last two years. The computational complexity and performance of video compression depends upon the method employed for video coding. A video which consists of a series of sequence called frames contains spatial and temporal redundancy. The temporal redundancy can be exploited using motion estimation techniques. Block matching Motion estimation is one of the ME technique used in various video coding standards such as MPEG-2, MPEG-4 and most recent H.264 due to its simplicity and better performance in order to reduce the computational complexity of ME algorithms. One of critical task is efficient implementation of ME because it is applied to block matching algorithm for choosing the best motion vector based on different quality constraints but it consume 80-90% time in the process of encoding a video signal [1]. Hence, it is important to study and analyze the computational complexity and performance tradeoffs for various motion estimation algorithms being employed in a video coding standard. In this paper, extensive studies of some of the block based motion algorithms available in the literature is done and analyze their performance through simulation in terms of PSNR and computational complexity. The rest of paper is organized as follows: Section II exploit framework of fast BMME algorithms, general properties of good matching algorithms and matching criterion used in block matching algorithms. Section III illustrates experiment /simulation based result and compared them in terms of PSNR and Computational time with different macro-block size and search window size for different types of motion activity present in video sequences and Section IV presents conclusion followed by references.

II. FRAMEWORK OF FAST BLOCK MATCHING MOTION ESTIMATION ALGORITHMS

It is confirmed from literature that motion estimation consumes approximately 70-90% of total computation time which affect the quality of video image efficiency and recovery. Therefore, to decrease such a huge computational time while retaining video quality, many fast and an efficient motion estimation (ME) algorithms have been proposed based on pixel, region and block. Block based motion estimation being simple and effectively implementation in hardware has led its use in most of the existing video compression standards like H.26X [2], MPEG-X, [3] and High Efficiency Video Coding [4]. In block matching motion estimation [5], frames are divided into numerous non-overlapping macro blocks and best matches is searched within the search range by using all possible position for each macro block and it is assumed that each block undergoes independent uniform translation given by the displacement vector called motion vector (MV). The micro-blocks are always taken as a square with 16*16 pixels with search parameter (p) is 7 pixels respectively as shown in Fig.1. The matching of these micro-blocks is done using a cost function and perfect matching is obtained when the cost function is minimum for current block. There are various matching criteria used for finding the cost function for matching the blocks, of which the most popular and less computationally expensive are: i) Mean Absolute Difference (MAD) given by eq. (1), ii) Mean Squared Error (MSE) given by eq. (2) and Peak Signal to Noise Ratio (PSNR) given by eq. (3).

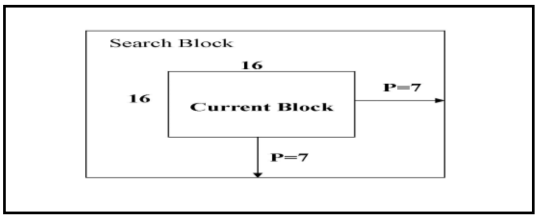

Fig. 1. Block Matching a macro block of side 16 pixels and a search parameter p of size 7 pixels [6].

$$MAD(d_1,d_2)=\frac{1}{N_1 \times N_2}\sum_{n_1,n_2 \in \phi} |s(n_1,n_2,k)-s(n_1+d_1,n_2+d_1,k)| \quad (1)$$

$$MSE(d_1,d_2)=\frac{1}{N_1 \times N_2}\sum_{n_1,n_2 \in \phi} [s(n_1,n_2,k)-s(n_1+d_1,n_2+d_1,k)]^2 \quad (2)$$

$$PSNR=10*\log_{10}\frac{Peaktopeakvalueoforiginalframe^2}{MSE} \quad (3)$$

Where ϕ denotes N1 x N2 blocks, for a set of candidate motion vectors (d1, d2). The estimate of the motion vector is taken to be the value of (d1, d2) which minimizes the MSE. However, MSE as matching criteria is not commonly used because it contains square terms which are difficult to realize in VLSI implementation hardware.

Full Research Block Matching Algorithm:

The most computationally expensive block matching motion estimation algorithm is full search algorithm (FS) which examines exhaustively all positions in the search window giving optimal result within the search range [6]. One of disadvantage of FS is that requiring a large amount of computations, especially for a large search pane which is offensive for real-time software-implemented video applications.

Three-Step Search Algorithm (TSS):

For real-time applications, quick and efficient fast block matching motion estimation algorithms using quadrant monotonic model [7] are proposed to reduce the computational complexity of FS while retaining similar prediction quality was three step search algorithms proposed by Koga et al (1981). TSS which is the most popular BMME is based on fine-coarse search mechanism. One of the disadvantages with TSS is that it uses a uniformly allocated checking point pattern which becomes inefficient for small motion estimation.

Simple and Efficient Search Algorithm (SES):

Simple and Efficient Search algorithm [8] is the variant of TSS which is also having three steps. Steps are divided into two phases: In the first phase, a search quadrant is selected, and in the second phase, the location of the minimum error in the selected quadrant is found. One of advantage of this algorithms is saving of computational time but it PSNR achieved is poor compared to TSS.

Diamond Search Algorithm:

Diamond search algorithm (DS) was projected by S. Zhu and K. K. Ma in 2000 [9]. It is similar to 4SS but search point pattern is adopted is diamond instead of square and no limit on number of steps it can take. Two different types of fixes search pattern is used namely Large Diamond Search Pattern (LDSP) Small Diamond Search Pattern (SDSP).

Adaptive Rood Pattern Search (ARPS):

ARPS [10-11] algorithm which consists of two sequential search stages: 1) initial search and 2) refined local search. This algorithm uses the motion vector of the macro block to its immediate left to predict its own motion vector. For the initial search stage, an adaptive rood pattern (ARP) is proposed, and the ARP's size is dynamically determined for each MB, based on the available motion vectors (MVs) of the neighboring MBs. In the refined local search stage, a unit-size rood pattern (URP) is exploited repeatedly, and unrestrictedly, until the final MV is found.

New Three Step Search (NTSS):

B.Zeng developed the New Three Step Search (NTSS) to improve on the TSS which was good for large motions but was prone to missing small motions (Jing and Chau, 2004) [12]. It used a centre biased searching scheme like TSS but it had a provision for a half way stop after the first or second step - thus reducing computational cost. It was one of the first widely accepted fast algorithms and frequently used for implementing earlier standards like MPEG 1 and H.261[13].This algorithm searches a minimum of 17 points but in the worst case it needs 33 locations to be checked.

III. Experimental Result And Discussion

The platform used for comparing the performance of various block matching motion estimation algorithms with three different video sequences representing different motion activity like slow, medium and fast in consecutive frames is MATLAB Software installed on 1.6 MHZ Intel Pentium machine with 2 GB RAM memory. All the fast block matching algorithms mentioned in section II are compared and analysis is done with different search parameters and different micro block size. Fig 2-3 provides the performance of each of the algorithms in terms of Computational time and PSNR with p=8 and micro block size w=16. It is clearly shown in fig.2 that there is little variation in value of PSNR for all the algorithms compared to ES algorithms for Susie video sequence. Fig.4 shows the variation of PSNR for different BMME algorithms with Susie, Caltrain and Miss America video sequences. From the figure, we conclude that there is little variation in PSNR for video sequences. Fig. 5 is related to computational time taken by motion estimation algorithms for finding the best matching block cost function. Full search algorithms consume maximum computational time as compared to ARPS which consume lowest computational time.

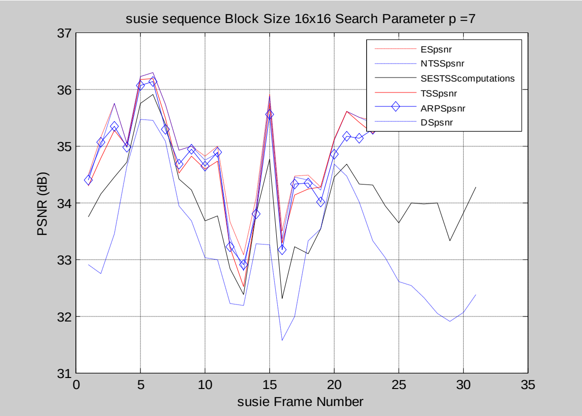

Fig. 2. PSNR Vs Number of Frames

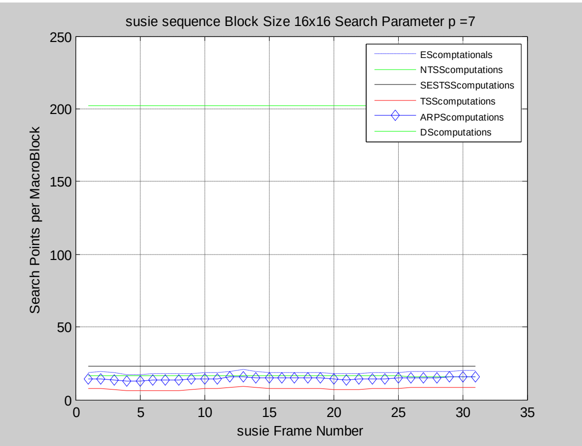

Fig. 3. Search Point Vs Number of Frames

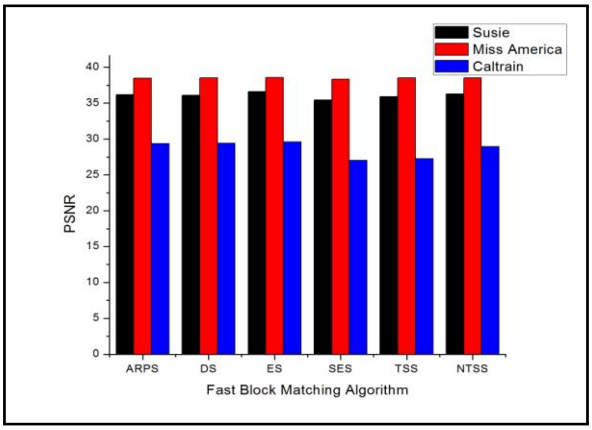

Fig. 4. PSNR Vs Block matching Algorithms

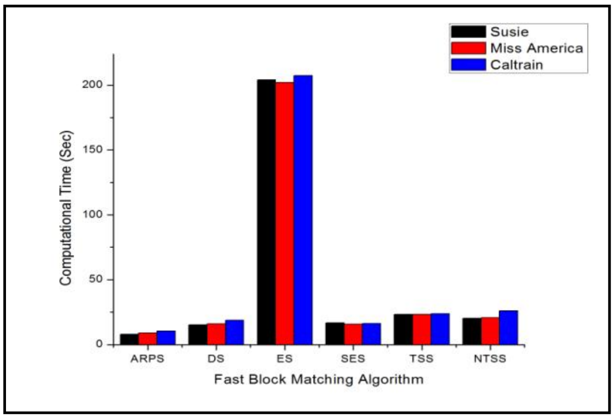

Fig. 5. Computational Time Vs Block matching Algorithms

IV. CONCLUSION

The past two decades have seen the extensive growth in the field of multimedia. Block matching are the most popular and efficient algorithms for motion estimation. In this paper, six different Block matching motion estimation algorithms are tested and compared from the aspects of computational complexity and PSNR. Three video sequences i.e Caltrain, Susie and Miss America Claire with variable motion activity, were used to test the performance of these BMME algorithms. The result reveals that ARPS requires lesser computational complexity and FS has highest. Similarly PSNR values for all BMA are calculated and it is found that all BMA have almost same PSNR value. A comparative study of computational time and PSNR for different Block Matching Motion estimation algorithms is shown in fig. 4 and fig. 5 respectively. It is concluded from the experimental results that this work could be improved by proposing new fast motion estimation algorithms.

V. REFERENCES

[1] Santamaki, Harry, Juha Leppanen, Pontti Haikonen, and Ilkka Korhonen. "Video compression method." U.S. Patent 4,717,957, issued January 5, 1988.

[2] Le Gall, Didier J. "The MPEG video compression algorithm." Signal Processing: Image Communication 4, no. 2 (1992): 129-140.

[3] Sullivan, Gary J., and Jens-Rainer Ohm, "Recent developments in standardization of high efficiency video coding (HEVC).", In SPIE Optical Engineering+ Applications, pp. 77980V-7798V, International Society for Optics and Photonics, 2010.

[4] Richardson I.G. H.264 and MPEG-4 video compression. Wiley, Chichester, England, 2003.

[5] Aw. Hussain, L. Knight, D. Al-Jumeily, P. Fergus, and H. Hamdan, “Block Matching Algorithms dor

Motion Estimation–A Comparison Study", International Journal of Scientific & Engineering Research, vol. 264, no. 2, pp. 356-369, 2014.

[6] M. Ahmadi and M. Azadfar, "Implementation of fast motion estimation algorithms & comparison with full search method in H.264 ", IJCSNS International Journal of Computer Science & Network Security, vol. 8(3) , pp. 139-143,2008.

[7] T. Koga, K. Iinuma, "Motion Compensated Inter drame Coding dor Video Conderencing," in proceedings of the NTC, pp. 230-236, 1981.

[8] JianhuaLu and MingL.Liou,"ASimple and Eddicient Search Algorithm dor Block-Matching Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, volume 7, no. 2, pp. 429-433, 1997.

[9] S. Zhu and K. Ma," A new diamond search algorithm for fast block Matching motion estimation" IEEE Transactions on Image Processing, vol. 9(2), pp. 287-317, 2000.

[10] Y. Nie, "Adaptive Rood Pattern Search dor Fast Block-Matching Motion Estimation," IEEETransactions on Image Processing, vol. 11, no. 12, pp. 1442-1448, 2002.

[11] Casey, J.: An Investigation of Block Searching Algorithms for Video Frame Codec, Master's Dissertation. Dublin, Dublin Institute of Technology, 2008.

[12] R. Li, B. Zeng and M. L. Liou, "A new three step search algorithm for block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, pp. 438-442, 1994.

[13] Barjatya, Aroh, "Block Matching Algorithms For Motion Estimation", DIP 6620 Final Project Paper in Digital Image Processing, Utah State University, pp. 1–6., 2004 [online]. Available from his homepage at http://cc.usu.edu/~arohb and also on the Mathworks file exchange site at http://www.mathworks.com/mathlabcentral.