

Video Compression Motion Estimation Algorithms – A Survey

1. C.Arunkumar Madhuvappan, Research Scholar, Anna University, Chennai
 amadhuvappan@gmail.com
2. Dr.J.Ramesh, Asst.Prof (Sr.Gr) / ECE, PSG College of Technology, Coimbatore
 jramesh60@yahoo.com

Abstract— Motion Estimation (ME) algorithm plays a vital role in video compression. This is used to find the motion information of the video sequence represented by Motion Vector (MV). The motion estimation is the primary reason for computational complexity of video compression. Some of the motion estimation algorithms such as Block matching Algorithm, Diamond Search, Adaptive Rood Pattern Searches, Reduced Bit Sum of Absolute Difference, Multipoint and Dynamic Multipoint Diamond Search are briefly discussed in this paper. A brief review is to provide the succeeding researchers with some useful information in design of the fast ME algorithms. The performances of the various algorithms are discussed with specific attention to the important trade-off between computational complexity, PSNR, Quality and Video Resolution.

Index Terms— Computational Complexity, Motion Estimation, Motion Vector, Macroblock, PSNR, Resolution & Pixel.

1 INTRODUCTION

In present scenario video data transmission plays a vital role for conveying lot of information. The video data cannot be transmitted as such because it occupies more memory for storage. To reduce the memory usage as well as improve the speed digital video coding has increased drastically since the 90s when MPEG-1 first came to picture (1). The different video coding techniques are developed to meet higher compression ratio even without loss of subjective quality of original video. Numerous video coding techniques have been developed to make less usage of memory.

The standards of video compression are given

by two organizations International Standard Organization (ISO) and International Telecommunication Union (ITU). In this ISO provides the standard related to the consumer applications where JPEG (Joint Picture Expert Group) for compressing still image while for moving pictures is MPEG (Moving Picture Expert Group). Most of the occasions both the organizations provided the standard slightly different based on applications. In the recent previous standard they worked together as JVT (Joint Video Team) and formed the common one as H.264/MPEG-4 AVC (Advanced Video Coding). The progression of the video compression standards are provided below.

S.No.	Standardization Period	Organization	Standards	Applications
1.	1984 - 1990	ITU-T	H.261	Video Conferencing
2.	1988 - 1993	MPEG	MPEG-1	CD-ROM
3.	1990 - 1994	JVT	H.262/MPEG-2	DVD, Digital Terrestrial TV, Satellite
4.	1993 - 1995	ITU-T	H.263	Video Conferencing, Surveillance
5.	1993 - 2000	MPEG	MPEG-4	Surveillance, DVD, Digital Still Cameras, Digital Video Camcorders and Cellular Media.
6.	1995 - 1998	ITU-T	H.263+	----
7.	1998 - 2001	ITU-T	H.263++	----
8.	1997 - 2003	JVT	H.264/MPEG-4 AVC	Surveillance, Videoconferencing, DVD, Satellite, DSL-based Video On

				Demand, Digital Still Cameras, and Cellular Media.
9.	2013	JVT	HEVC(High Efficiency Video Coding)/ H.265	Mobile, HDTV, Surveillance, Videoconferencing, DVD, Satellite, DSL-based Video On Demand, Digital Still Cameras, and Cellular Media.

Table No.1 Progression of the Video Compression Standards

The motion estimation is one of the blocks of video compression, which makes computational complexity more. In motion estimation techniques analysis of the movements between consecutive frames carried out and their temporal redundancy being reduced in time domain. Motion estimation is done by selecting a same block position of two consecutive frames or two different blocks of same frame to calculate the motion vectors of the two where the best matching was found. It performs interframe prediction as the main operation and uses 80% of the total computational complexity of video coders which in turn results more power consumption [1]. Motion estimation necessitates more computation to carry out its entire operation. To do motion estimation with reduced computational complexity different types of fast block matching algorithms were developed. In these algorithms massive computation used in the part to find the best match candidate blocks by careful search.

A simple and foremost motion estimation algorithm is Block Matching Algorithm (BMA) and the exhaustive search of BMA is known as the Full Search Motion Estimation (FSME) algorithm which is widely used. In this the best match candidate block is found by using more number of searches. The matching of the block is measured using one of the cost functions like Sum of Absolute Difference (SAD) or Mean Absolute Difference (MAD) or Mean Square Error (MSE) which also gives the block distortion measure.

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

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (2)$$

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (3)$$

Where C_{ij} and R_{ij} are current and reference frames respectively.

To find the best match the error metric term most often used is Sum of Absolute Difference

(SAD). The computational complexity of calculating SAD of BMA is more. To enhance the speed and to reduce computations the number of search can be limited by using fast search Block Matching algorithms. In that one of the fast search algorithms is three step search algorithm. It uses the concept of center biased checking point patterns to make search locations smaller to find the best match at faster speed. The center biased concept was based on the feature centre biased motion vector distribution of the image sequence [2].

The other fast search algorithm is Diamond Search (DS) [3, 4] Algorithm in which it uses two search patterns. The patterns are Large Diamond Search Pattern (LDSP) and Small diamond Search Pattern (SDSP). In LDSP nine checking points are there in that eight points surround the center one to form the diamond shape and that of SDSP five checking points used.

To simplify and make less number of searches to find the best match of candidate blocks Adaptive Road Pattern Searches (ARPS) algorithm being developed [5]. In this it has two sequential search stages which are initial and refined local search. Using the calculated Motion Vectors (MVs) of the adjacent MacroBlocks (MBs) the Adaptive Road Pattern is proposed for initial Search. The initial search is done only at the beginning for each MacroBlock (MB) which avoids the intermediate search. Till the final motion vector is found a Unit-Size Road Pattern (URP) is broken repeatedly and unrestrictedly.

The motion estimation computational complexity is reduced in the fast search sub-sampling algorithms even though the result quality can depreciate as well. The hardware implementation using VLSI can be a more complex one due to irregular data flow. In these algorithms the data reuse concept is lacking which in turn causes multiple read of the same data from external memory. The multiple read of the same data causes more power consumption [6].

The computation burden can be reduced by truncating the pixel values from 8-bit luminance to 4-bit values i.e., depreciating the pixel data resolution. The truncation is done at the lower 4-bit

and the same recovered whenever to compensate the quality loss using the spatio-temporal correlation based algorithms. This method is known as Reduced Bit Sum of Absolute Difference (RBSAD). The video sequences with high spatial and temporal correlations will have poor PSNR values and need greater bit rate in this method [6 - 9].

Most of the algorithms used for motion estimation are mainly for lower resolution videos such as QCIF (Quadrature Common Intermediate Format) and CIF (Common Intermediate Format). The fast motion estimation algorithms result quality will change as there is increase in resolution of video. The difference in resolution will be based on number of pixels, in high resolution it has more number of pixels or detail representation of video sequence and in low resolution it is inverse. The quality results of the Fast Algorithms will be close to each for low resolution videos. For high-resolution videos the fast algorithms uses the local minima as the best matching which leads quality losses. To enhance the quality in high resolution two new fast algorithms have been reported they are of Multi-Point Diamond Search Algorithm (MPDS) and the Dynamic Multipoint Diamond Search (DMPDS) Algorithm [10, 11]. These Algorithms are used for motion estimation in case of high resolution video.

The paper is prepared as follows: Section 2 provides a brief explanation about Block Matching Algorithm. Section 3 show details about the Fast search algorithm. Section 4 provides the comparison table. Finally, Section 5 presents the conclusion.

2 BLOCK MATCHING ALGORITHM

Block Matching Algorithm [12] is simple and initial one of motion estimation. The algorithm divide the frame into group of macroblocks and then each block is compared with the corresponding block or its adjacent neighbors in previous frame to find the motion between them. The resultant value of the motion is named as motion vector. The same principle is repeated for all macroblocks in entire frame and the motion is estimated for the best match. The number of search is based on the search parameter 'p' pixels on four side of the block. Normally the macroblocks is of size 16 pixels and the search parameter of 7 pixels. Larger the motions between the frames larger the search parameter required. The estimated motion vector is used for predict the new frame from reference frame at the receiver side which is known as motion compensation. The motion

vector comprises of two parts which is horizontal and vertical. The values of the parts of motion vector can be positive or negative value. Based on the sign of value the motion can be predicted. The parts as positive value means the motion is to the left or upward and if it is negative value then the motion is to the right or downward [13].

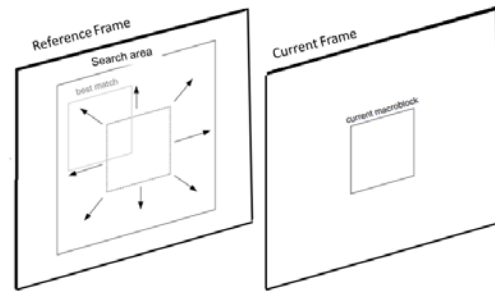


Figure No.1. Block Motion Estimation

Larger motion in a video sequence makes more number of macroblock to be compared to estimate the motion vector which increases the computation. The computation can be reduced by increasing the macroblock size (i.e., fewer blocks per frame). This will result in poor prediction of motion between the frames. The exhaustive search of BMA is known as Full Search Motion Estimation (FSME) Algorithm. In this the cost function is calculated in each and every location of the search area. The FSME algorithm will give highest PSNR compare to other algorithm along with best possible match. The disadvantage of FSME algorithm requires more computation as the search area is large.

3 FAST SEARCH ALGORITHM

The Fast search algorithm is classified into six categories in that first five are lossy and the other one is lossless. The lossy five categories will produce poor quality results than FSME and lossless one will produce the same as that of FSME. The Categories are of

a. Reduction in Search Positions

The search positions to find the best match is reduced with the assumption as the distortion increases monotonically as it moves away from the minimum distortion point. Decimation of search position helps to reduce the computation of algorithms under this category. Since 1981, many algorithms like Two Dimensional Logarithmic Search [14], Three Step Search [15], New Three Step Search [16], Four Step Search [17], Cross

Search [18], and Diamond Search have been proposed [19 -20] .

b. Simplification of Matching Criterion

Block matching is done based on the cost function SAD or MAD or MSE which can be simplified using sub-sampling scheme [21]. The Scheme takes into account only every second pixel for the estimation of distortion which in turn reduces the computation. Four Sub-sampling patterns are used in different search position to avoid the aliasing effect if not using low pass filtering aliasing effect can be avoided [22,23]. Also the algorithms have been proposed simplifying the matching criterion using the concept of Adaptive - Pixel decimation, Pixel Difference Classification (PDC), MinMax Criterion and integral projections.

c. Bit-width Reduction

Normally a pixel is represented using 8-bits which can be represented as one bit using transform in some algorithms [24, 25]. Bit-Width of the pixel representation can be truncated on an average more than four bits without affecting the quality [26]. The truncation of pixel can help in saving power and reduces hardware complexity. Fixed length truncation can save power and reduces hardware complexity but it affects the quality. Adaptive pixel truncation adopts masking of least significant four bits of pixel as zero reduces area and saves power without degrading quality. Pixel truncation can also be adopted in software implementation using SIMD (Single Instruction Multiple Data). This technique is used in some of the algorithms like RBSAD [6-9].

d. Predictive Search

Initial estimate of target Motion Vector (MV) can be predicted utilizing the motion information in the spatial and/or temporal domain between current and neighbouring blocks. The overhead information of MV is less in this it makes additional compression by reducing the motion search. Certain criteria being adopted for selecting neighbouring motion vector to predict motion vector or it can be predicted using the statistical average calculation of neighbouring motion vectors. The predicted motion vectors can be of top, left or top right and their median, they can also be the MV of the collocated Macroblocks of the previous frame or previous two frames. This result in reduction of search area as well as power

consumption but it requires additional memory for storing neighbouring motion vector [27]. For example algorithms like Adaptive Rood Pattern (ARP) [28] and simplified Block Matching Algorithm for Motion Estimation [29] adopts this technique.

e. Hierarchical Search

The technique finds the correlation between the different levels of resolution of the same frame to predict the initial estimate at the coarse level and then it is being refined at the fine level. This technique is also known as Multiresolution Structure or Pyramid Structure. In the fine level number of levels will be less compared to the original search range. There is higher probability of being trapped in local minimum when more number of levels is used to save computation, which is mainly because while the images are scaled down it loses the detailed textures of it. This technique is one of the best techniques in Block Matching Algorithm and it is mainly used in larger search areas and frames. Two level or three level Hierarchical search is usually used [30-32].

f. Fast Full Search

In this technique at the earlier stage the optimal candidate block will be detected. Further distortion calculation will be based on this candidate block and it results in reduction of larger number of computation. Successive Elimination Algorithm (SEA) calculates the absolute difference between sum of all pixels of candidate block and current block and checks whether larger than the up-to-date SAD minimum notated as SAD_{min} and eliminates the search position under the condition larger SAD value [33]. The current block sum will be calculated only once and that of candidate block can be speed up by partial result reuse technique. Skipping process can be speed up further using Multilevel Successive Elimination Algorithm (MSEA) [34]. MSEA computation can be still reduced by combining with SIMD.

Partial Distortion Elimination (PDE) [35] Algorithm is a Simple and effective concept. This algorithm adopts half way stop technique to reduce the computation and improve the speed of process. The candidate block can be skipped when partial distortion value larger than the current minimum distortion. Normalization of partial and current distortions can speed up earlier skipping of candidate block [36]. The probability of skipping can be improved by grouping partial distortion

and spiral scan of search. However the normalized PDE result may not be exactly same as the Full Search Motion Estimation Algorithm but it will be closer.

Winner Update algorithm is one of the interesting fast search algorithms. The algorithm uses the game of poker cards where there are five players in that the entire player shows a card and then next card can be shown by the player with minimum value of first card. Then the player minimum sum of value of card can reveal the next card and at last the player who shows four cards first will be the winner. In the algorithm the candidate block corresponds to the players and pixel in a block will be the number of cards dealt for each player [37]. The results of fast full search algorithm will not be exactly same as the Full

Search Motion Estimation Algorithm it will have some minor differences. In searching two or more search position may have the same minimum SAD Value and hence the result will dependent on scan order. But these minor differences will not cause any effect on quality.

4 COMPARISON OF MOTION ESTIMATION ALGORITHMS

The parameters mainly considered for analyzing motion algorithms are of PSNR (Peak Signal to Noise Ratio), Quality of the Video and also computational complexity. In below table comparison of motion estimation algorithms related to the above parameters are mentioned.

S.No.	Algorithm	PSNR	Quality	Video Resolution	Computational Complexity
1	Block Matching Algorithm	More	Good	Holds good for Low & High Resolution Videos	More Number of blocks increases the computation
2	Full Search Algorithms	Highest	Good	Holds good for Low & High Resolution Videos	Larger Search Range increases the computation
3	a.Fast Search Algorithms (First Five Categories)	Reduced	Reduced	Holds good for Low Resolution Videos but quality losses are there for high resolution videos.	The Five Categories of Lesser Search Area, Simplified Matching Criterion, Bit Width Reduction, Predictive and Hierarchical Search make the computation less.
	b. Fast Search Algorithms (Last Category)	Improved	Good (Not Exactly as FSME but only minor differences)	Holds good for Low Resolution Videos but quality losses are there for high resolution videos not to the level of first five categories.	The Last Category of Fast Full Search makes the computation less.
4	MPDS & DMPDS	Improved more than Fast Search but not to the level FSME	Better Quality than Fast Search but not to the level FSME	Used for High Resolution Videos.	Lesser than Full Search Algorithms

Table No.2 Comparison of Motion Estimation Algorithms

5 CONCLUSION

Motion Estimation (ME) algorithm is a very essential process in video compression. The motion estimation is the primary reason for computational complexity of video compression. In this paper, a review of various algorithms, such as Block matching Algorithm, Diamond Search, Adaptive Rood Pattern Searches, Reduced Bit Sum of

Absolute Difference, Multipoint and Dynamic Multipoint Diamond Search are presented, also computational complexity, result quality, advantages and disadvantages of these algorithms have been presented. From the above survey it found that reduction in computational complexity achieved with the quality loss, we plan to reduce the computational complexity without degrading the quality with the help of novelty algorithms.

REFERENCES

- [1]. Cheng,Y., et al.2009. "An H.264 Spatio – Temporal Hierarchical Fast Motion Estimation Algorithm for High-Definition Video," IEEE ISCAS,pp.880-883, May 2009.
- [2]. Ning-ning, S., Chao, F., Xu, X. "An Effective Three-step Search Algorithm for Motion Estimation. IEEE Int Symposium on IT in Medicine & Education," 2009:400- 403.
- [3]. J. Y. Tham, S. Ranganath, A. A. Kassim. "A novel unrestricted centre-biased diamond search algorithm. IEEE Trans. Circuits and Systems for Video Technology," 1998, 8: 369–377.
- [4]. Shan Zhu, and Kai-Kuang Ma, " A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," IEEE Trans. Image Processing", vol 9, no. 2, pp. 287-290, February 2000.
- [5]. Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," IEEE Trans. Image Processing, vol 11, no. 12, pp. 1442-1448, December 2002.
- [6]. P. Kuhn. "Algorithms, complexity analysis and VLSI architectures for MPEG4 motion estimation," Germany: Kluwer Academic Publishers, 2003.
- [7]. V. Ila, R. Garcia, F. Charot. "VLSI architecture for an underwater robot vision system. Oceans Europe," 2005: 674–679.
- [8]. C. Hisham, K. Komal, A. K. Mishra. "Low power and less area architecture for integer motion estimation. International Journal of Electronics, Circuits and Systems," 2009, 3(1): 11–17.
- [9]. J. Miyakoshi, Y. Murachi, K. Hamano, et al. "A low power systolic array architecture for block-matching motion estimation," IEICE Trans. Electronics, 2005, 88(4): 559–569. JCT, Working Draft 3 of High-Efficiency Video Coding, JCTVC-E603, 2011.
- [10]. M. Porto, G. Sanchez, D. Noble, S. Bampi, and L. Agostini, "An efficient ME architecture for high definition videos using the new MPDS algorithm," in Proceedings of the 24th symposium on Integrated circuits and systems design (ACM SBCCI '11), pp. 119–124, 2011.
- [11]. G. Sanchez, M. Porto, S. Bampi, and Agostini, "Real time QFHD motion estimation architecture for DMPDS algorithm," in IEEE Southern Programmable Logic Conference,2012.
- [12]. A. Barjatya, "Block Matching Algorithms for Motion Estimation," DIP 6620 Spring 2004.
- [13]. S. Metkar and S. Talbar, "Fast motion estimation using modified orthogonal search algorithm for video compression," Signal, Image and Video Processing, vol. 4, pp. 123–128, 2010.
- [14]. J. Jain and A. Jain, "Displacement Measurement and its Application in Internal Image Coding," IEEE Trans. Commun., vol.COM-29, no. 12, 1981, pp. 1799–1808.
- [15]. T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., 1981, pp. C9.6.1–C9.6.5.
- [16]. R. Li, B. Zeng, and M.L. Liou, "A New Three-step Search Algorithm for BlockMotion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, no. 4, pp. 438/442, Aug. 1994.
- [17]. 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, 1996, pp. 313–317.
- [18]. M. Ghanbari, "The Cross Search Algorithm for Motion Estimation," IEEE Trans. Commun., vol. 38, no. 7, 1990, pp. 950–953.
- [19]. 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, 1998, pp. 369–377.
- [20]. S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block-matching Motion Estimation," IEEE Trans. Image Processing, vol. 9, no. 2, 2000, pp. 287–290.
- [21]. M. Bierling, "Displacement Estimation by Hierarchical Block Matching," in Proc. of SPIE Visual Commun. Image Processing (VCIP'88), 1988, pp. 942–951.
- [22]. A. Zaccarin and B. Liu, "Fast Algorithms for Block Motion Estimation," in Proc. of IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP'92), 1992, pp. 449–452.
- [23]. B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 2, 1993, pp. 148–157.
- [24]. B. Natarajan and V. Bhaskaran, "Low-complexity Block-based Motion Estimation via one-bit Transforms," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 4, 1997, pp. 702–706.
- [25]. J.H. Luo, C.N. Wang, and T. Chiang, "A Novel All-binary Motion Estimation (ABME) with Optimized Hardware Architectures," IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 8, 2002, pp. 700–712.
- [26]. Z.L. He, C.Y. Tsui, K.K. Chan, and M.L. Liou, "Low-power VLSI Design for Motion Estimation using Adaptive Pixel truncation," IEEE Trans. Circuits Syst. Video Technol., vol. 10, no.5, 2000, pp. 669–678.
- [27]. J. Chalidabhongse and C. C. J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, pp. 477-488, 1997.
- [28]. Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," IEEE Trans. Image Processing, vol. 11, pp. 1442-1448, 2002.
- [29]. M. Ezhilarasan and P. Thambidurai, "Simplified Block Matching Algorithm for Fast Motion Estimation in Video Compression," Journal of Computer Science vol. 4, pp. 282-289, 2008 2008.
- [30]. D. Tzovaras, M.G. Strintzis, and H. Sahinlou, "Evaluation of Multiresolution Block Matching Techniques for Motion and Disparity Estimation," Signal Processing: Image Commun., vol. 6, 1994, pp. 56–67.
- [31]. J.H. Lee, K.W. Lim, B.C. Song, and J.B. Ra, "A Fast Multiresolution Block Matching Algorithm and its VLSI Architecture for Low Bit-rate

Video Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 12, 2001, pp. 1289–1301.

[32]. J.H. Lee and N.S. Lee, "Variable Block Size Motion Estimation Algorithm and its Hardware Architecture for H.264," in Proc. of IEEE Int. Symp. Circuits Syst. (ISCAS_04), 2004, pp. 740–743.

[33]. W. Li and E. Salari, "Successive Elimination Algorithm for Motion Estimation," IEEE Trans. Image Processing, vol. 4, no. 1, 1995, pp. 105–107.

[34]. X.Q. Gao, C.J. Duanmu, and C.R. Zou, "A Multilevel Successive Elimination Algorithm for Block Matching Motion Estimation," IEEE Trans. Image Processing, vol. 9, no. 3, 2000, pp. 501–504.

[35]. "Digital Video Coding Group, ITU-T recommendation H.263 software implementation," Telenor R'D, 1995.

[36]. C.K. Cheung and L.M. Po, "Normalized Partial Distortion Search Algorithm for Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 3, 2000, pp. 417–422.

[37]. Y.S. Chen, Y.P. Huang, and C.S. Fuh, "Fast Block Matching Algorithm based on the Winner-update Strategy," IEEE Trans. Image Processing, vol. 10, no. 8, 2001, pp. 1212–1222.

IJSER