Motion Estimation Algorithm Based On Mean Deviation and Sorting Approach

Akshat Agrawal

[1]Assistant Professor, Amity University Haryana

***Abstract*— Technologies such as TV, HDTV, 3D TV, Video telephony, Video surveillance, and Wireless multimedia communication have put a great demand for development of efficient and fast video compression algorithms. Motion Estimation (ME) is being used to achieve and increase the effectiveness of video compression. Traditional ME algorithms are not suited to current market demands. In this paper we propose a Motion Estimation algorithm which is based on Partial Distortion Search and Sorting schemes. It tries to minimize the calculations by rejecting those partial distortion calculations early which would yield bad motion vectors. It is based on the concept that if deviation of sub-block is higher, then its contribution to total SAD (Sum of Absolute difference) is also higher. It uses this approach along with sorting of sub-blocks on the basis of their deviation. The results show that this algorithm can save 60 to 70 % of computation costs as incurred by traditional algorithms as well as performs well than other Partial Distortion Algorithms. Also PSNR degradation is very less and is 0.017 on average.**

***Keywords*— ME(Motion Estimation), MV(Motion Vector), SAD(Sum of Absolute Difference), BMV(Block Mean Value).**

I. INTRODUCTION

Motion estimation is the process which generates the motion vectors that determine how each motion compensated prediction frame is created from the previous frame. The basic concept of motion estimation is that in most cases, consecutive video frames will be similar except for changes induced by objects moving within the frames or camera movement. Motion Estimation module is the core module of all video coding schemes because of its great effectiveness in minimizing the temporal redundancy that exists between successive frames of video stream and thus enables the transmission and storage of video signals using much lower video rate[1]. Most of the video coding schemes like MPEG 1 / 2 [2] [3] [4] [5] MPEG-4 , ITU-T H.261/262/63 [6] employ various motion estimation algorithms for predicting motion vectors. It has been observed that ME contribution to algorithmic complexity of H.264 codec is around 70% on average [7]. The mostly used technique is BLOCK MOTION ESTIMATAION, it divides the video frame into fixed N*N blocks usually 16*16 and obtains a motion vector for each of the blocks within a search window in the reference frame by obtaining minimum SAD for the previous frame. The simple algorithm is Full Search Algorithm which provides accurate results by matching all candidate blocks inside a search window but full FS algorithm suffers from high computational complexity and is not suited for real time video compression. In order to reduce the complexity load many algorithms have been proposed such as three step search (3SS) and new three step search(N3SS) [8], diamond search(DS) [9], four step search(4SS)[10], 2D-Logarthimic Search algorithms. These are able to reduce the complexity by matching only some of the pre defined points within the search window. They assume that the BDM increases as the checking point moves away from global minimum point. However this assumption doesn't hold in real word video streams [7] especially in complex and irregular motion videos at low frame rates and thus they get trapped in local minimum points and thus result in high matching errors and more video degradation than FS algorithm. More improved versions for lossy estimation have been proposed for more accurate prediction. These are based on Partial distortion technique [11][12][13][14]. The approach is to terminate the ME calculation as early as possible ie' during the calculation of the matching criterion in each macro block. The PDS reduces the complexity by terminating the measuring distortion ie' SAD calculation early if it finds that partial SAD is greater than the minimum SAD encountered so far during search. To improve PDS two sorting based algorithms were proposed by Montrucchio and Qualgia called as fast full search with sorting by distortion (FFSSD) and fast full search with sorting by gradient(FFSSG) [15] which perform better than normal PDS algorithm. One of the major improvements over normal PDS was NPDS (Normalized partial Distortion Search) which tries to reduce the complexity by rejecting the non-possible candidate motion vectors much earlier. It serves as the basis of almost all partial distortion search algorithms.

II. REVIEW OF NPDS

In this section, we will summarize the NPDS algorithm since most of the partial distortion algorithms are based on this algorithm. Usually in PDS, the frequently used measure for matching Distortion is SAD. Let F(x , y) is the Intensity of pixel in frame N, (p , q) is the location of upper left corner of 16 * 16 block. The SAD between the Block with coordinates (p , q) of frame N and the reference block with coordinates (p+u , q+v) where u , v describe the motion is given by equation 1

$$\sum_i^{16}\sum_j^{16} |f_n(x+i, y+j) - f_{n-1}(x+i+u, y+j+v)| \quad (1)$$

The SAD, which is the matching error accumulated for every period is computed and compared with the minimum SAD already found with another candidate vector. If at any stage it is found to be larger than the minimum SAD (MIN_SAD) , the candidate block cannot be the most similar block regardless of the rest of the incomplete matching computations. Therefore, the PDS algorithm can find and remove impossible candidates before complete matching error calculation of candidate block. NPDS is used to reduce to computations further. This algorithm reduces computations by using a half stop technique in the calculation of block distortion measure. In order to increase the chances of early rejection of non-possible motion vectors, it normalizes the accumulated partial distortion and the current minimum distortion before comparison. The probability of early rejection of non-possible candidate motion vectors (CMV) is thus increased.

To reduce the number of comparison operations, the partial distortion is considered as a group of pixels distortion instead of single pixels distortion as used in the traditional PDS algorithm. Thus, the SAD(x,y:u,v) is divided into 16 partial distortions SADp, where each partial distortion consists of 16 points spaced equally between adjacent points, as shown in Fig 1. The purpose of this grouping is to make sure that each SADp does not get localized in particular region of the search space. The p-th partial distortion is defined as

$$\sum_{i=0}^{3}\sum_{j=0}^{3}\left|f_n(x+4i+h_p\,,y+4j+v_p)-f_{n-1}(x+4i+h_p+u\,,y+4j+v_p+v)\right| \qquad (2)$$

The values hp and vp are the horizontal and vertical offsets of the upper left corner point of the p-th partial distortion from the upper left corner point of the block, respectively. The order of SADp calculation is chosen in such way so as to ensure that for each of the accumulated partial distortion, the pixels considered for the calculation are evenly distributed on the block. This order may vary in the different versions of algorithm. The corresponding (hp, vp) values for the 16 partial distortions are listed in Table 1. The p-th accumulated partial distortion is defined as

$$AC\_SAD_p(x,y:u,v)=\sum_{i=1}^{p} SAD_p(x,y;u,v) \qquad (3)$$

(hp,vp)

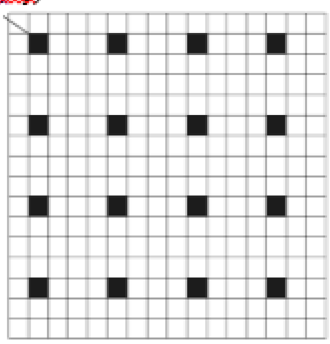

Table 1. Offsets (hp,vp) of partial distortions

P	(h_p,v_p)	P	(h_p,v_p)
1	(0,0)	9	(1,0)
2	(2,2)	10	(3,2)
3	(2,0)	11	(0,1)
4	(0,2)	12	(2,3)
5	(1,1)	13	(3,0)
6	(3,3)	14	(1,2)
7	(3,1)	15	(2,1)
8	(1,3)	16	(0,3)

NPDS algorithm performs better than other BMA's and is close to FS algorithm. It maintains MSE performance very close to FS algorithm even in case of large motion in video. It is thus more robust than other BME algorithms that limit the no of checking points for reducing computation.

However NPDS has a limitation on the video quality performance and the computational reduction. It does not take into account the the order of pixels within a block. The computational complexity can be further reduced if we take into account the order of pixels within the block. The technique is discussed in next section.

The computational complexity can further be reduced if we will be able to reject the impossible candidate motion vectors as

NPDS ALGORITHM

SAD_p: p-th partial distortion

AC_SAD_p: p-th accumulated partial distortion

MIN_SAD: current minimum SAD.

STEP 1. Initialize P=1 and SAD_p=0.

STEP 2. Calculate partial distortion SAD_p using equation 2.

STEP 3. Calculate accumulated partial distortion

$AC\_SAD=AC\_SAD_{p-1}+SAD_p$

STEP 4. If(16AC_SAD_p>p MIN_SAD) i.e. if Current partial distortion greater than normalized minimum distortion)continue with next CMV

Else

i. p=p+1

ii. If p>16 then MIN_SAD=SAD_{16} and continue with next CMV.

iii. else go to step 2

early as possible. In order to improve the efficiency of PDS algorithms many techniques were adopted. Cheng and Po [16] proposed adjustable PDS which can fine-tune the prediction precision against the searching speed by a quality factor. E-NPDS is presented in [17] which also rejects unnecessary calculations. A two stage sorting based partial distortion algorithm which is based on pattern similarity matching error is presented in [18]. They reject invalid motion vectors at early stage by predicting a total matching error between candidate and matching block. However it leads to image degradation because its prediction approach is based on linear model. In this paper we propose another approach which is based on [19] along with sorting approach [18]. They show that there is close relationship between the distance from the block mean value of the current block and the contribution to the SAD. Our proposed algorithm "Enhanced Motion Estimation based on Block Mean and Sorting approach" EBD-PDS tries to utilize the block mean of the pixel block metric to assist in early rejection of impossible motion vectors. It divides the 16*16 block into 16 4*4 sub-blocks as shown in fig 2 and follows the searching order-center of search window, then MV of left block, then MV of upper block, then MV of upper left block ,then MV of upper right block and finally the spiral search of all remaining search points in search window.

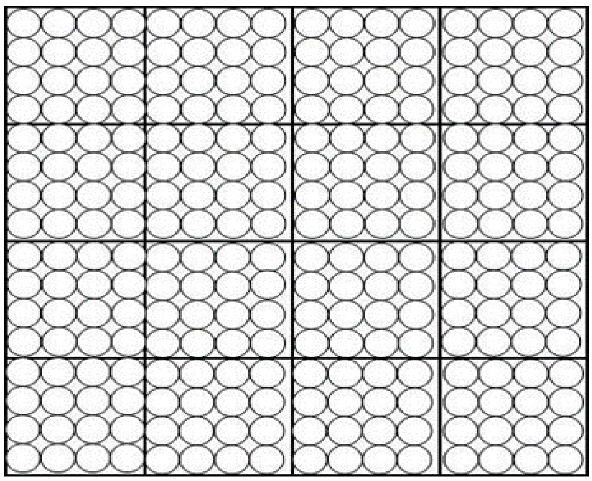

Fig 2. 4*4 sub-blocks of 16*16 Image block

III. PROPOSED ALGORITHM.

Our proposed algorithm tries to utilize two metrics, the block mean of the pixel block(BMV) and Threshold SAD(T_SAD) to assist in early rejection of impossible motion vectors. Rather than comparing the current partial distortion with the current minimum distortion we compare it with T_SAD. This T_SAD is adaptive and varies for every SADn. For every nth partial distortion we find the threshold SAD and compare our partial SAD with Threshold SAD and at any time if partial distortion exceeds the T_SAD we skip the calculation of remaining partial distortions.

Different algorithms use different ways to calculate T_SAD. In this algorithm we calculate T_SAD as follows

$$T\_SAD_n = T\_SAD_{n-1} + E_n * MIN\_SAD \qquad (4)$$

Initial value of T_SAD=0 and we calculate all threshold SAD ie T_SADn : n= 1 to 16

En is the weighting measure for n-th partial distortion. It measures how much n-th sub block contributes to total SAD calculation and is defined as

$$E_n = \partial_n / \sum_{n=1}^{16} \partial_n \qquad (5)$$

This formula is based on the fact that if the deviation of block is higher then this block is expected to contribute much to the SAD. Therefore we can say that Absolute deviation of nth block is directly proportional to distortion.

Here ∂n (Absolute deviation of n-th block) is calculated as

$$\partial n = \sum_{i=1}^{4} \sum_{j=1}^{4} |I_n(i,j) - BMV_k| \qquad (6)$$

BMVn is the block mean pixel intensity of nth sub-block.

3.1 Algorithm Steps

STEP 1. Divide the 16*16 block into 4*4 sub-blocks. Initialize Current Minimum Partial Distortion MIN_SAD to zero and Threshold SAD T_SAD to zero.

STEP 2. Find the BMV(Block Mean Value) of each 4*4 sub-blocks.

STEP 3. Calculate Absolute divergence of each block. The absolute divergence of nth sub-block is given by

∂n = ∑_(i=1)^4▒∑_(j=1)^4▒|I_n (i,j)-BMV_k | where BMVk is the block mean pixel intensity of nth sub-block.

STEP 4. Sort all the ∂n (n=1 to 16) in Descending order so that we can pick those blocks first which are complex and contribute much to SAD.

STEP 5. Using Equation 2 for finding SAD, start finding Partial SAD's starting from largest ∂. The AC_SADn (Accumulated n-th SAD) is equal to sum of all SAD's till n-th block and is equal to

AC_SADn=SADn + SADn-1 + SAD n-2 +…….. +SAD1

STEP 5. Use equations 4 and 5 to calculate T_SAD

If (AC_SADp > T_SAD) then

Terminate the calculations of remaining partial distortions and start the procedure again with different search location based on appropriate strategy in the Block being encoded.

Otherwise find the next partial distortion and compare again with T_SAD. In this way we can skip those calculations to a large extent which are unnecessary.

STEP 6. If final AC_SAD16 is still less than T_SAD, then we compare it with Current MIN_SAD. If AC_SAD16 is less than MIN_SAD, then the AC_SAD16 becomes Current MIN_SAD. This process continues till the search window completes. And finally we will get the motion vector of the matching block.

However to compute these various measures there is some computational overhead. The calculation of BMV for each block will require around 15 additions and one division operation .For calculation of Absolute deviation of each block we require 16 absolute operations, 15 addition operations,16 subtraction operations, and 15 more addition operations for finding BMV and one division operation. Also we need to sort all Absolute deviations and it can be done very quickly because we have to sort only 16 values. Despite these computations involved this algorithm still achieves better performance than normal PDS algorithms as shown in the results.

IV. RESULTS

In this work various parameters are used to analyze the performance of this algorithm which include PSNR, No of SAD calculations per block . The algorithm is run on various types of video sequences as shown in the Table 3 and Table 4. The block size is set to 16*16 pixels and the search window size is ± 16. The results show that the PSNR reduction is less than in NPDS and is equal to 0.016 dB. The results also show that there is around 60% saving in SAD Calculations. Ie very fewer partial SAD are computed per motion block in EBD-PDS than in PDS and NPDS. The PSNR performance of PDS and FS are almost same.

Table 1

Video Sequence	PSNR				
	ALGORITHM USED				
	FS	PDS	NPDS	EBD-PDS	PSNR DEGRADT-ION
Football	27.36	27.36	27.20	27.33	
Flower	34.93	34.93	34.71	34.93	NPDS= 0.172
Table	43.92	43.92	43.77	43.90	
Foreman	27.35	27.35	27.24	27.34	EBD-PDS= 0.016
Bus	35.39	35.39	25.17	25.36	
Average	33.79	33.59	31.61	31.77	

Table 4

Video Sequence	No of computed partial SAD per Motion Block				
	ALGORITHM USED				
	FS	PDS	NPDS	EBD-PDS	Less no of SAD computations involved in EBD-PDS
Football	16	5.38	2.09	2.08	
Flower	16	4.09	2.07	1.98	
Table	16	1.68	2.01	2.00	
Foreman	16	5.51	2.11	2.05	
Bus	16	5.52	2.12	2.00	

V. CONCLUSIONS

In this we proposed a new algorithm based on block mean value and sorting sub-blocks on the basis of their deviation. Due to these approaches we achieve efficient and fast motion estimation because of early rejection of impossible MV's. As the results show the computational overhead is significantly reduced. Moreover the same algorithm can be modified by applying some new method to calculate deviation of the sub-blocks.

REFRENCES

[1]. Encyclopedia of Multimedia Furht, Borko (Ed.) 2nd ed., 2008,

[2]. Yao Wang, Jorn Ostermann, and Ya-Qin Zhang. “Video Processing and Communication” Prentice Hall Signal Processing Series, pp. 406-454, 2002.

[3]. Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video, ISO/IEC 11 172-2 (MPEG-1 Video), 1993.

[4]. Bernd Girod, E. Steinbach, N. Farber. “ Comparison of the H.263 and H.261 Video Compression Standards” SPIE Proceedings Vol. CE60, Standards and Common Interfaces for Video Information Systems, 1995.

[5]. Borko Furht, Joshua Greenberg, Raymond Westwater, Motion Estimation Algorithms For Video Compression. Massachusetts: Kluwer Academic Publishers, 1997. Ch. 2 & 3.

[6]. A. puria, X. Chenb, and A. luthrac. “ Viseo coding Using the H.264/MPEG-4 AVC Compression standard” , SP:IC(19),No, 9, October 2004,pp. 793-849rl.doc

[7]. Z. Chen, P. Zhou, Y. He, “Fast integer pel and fractional pel motion estimation in for JVT”, JVT-F017, Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, Awaji, island, Dec, 2002.

[8]. Renxiang Li, Bing Zeng, and Ming L. Liou, “A New Three-Step Search Algorithm for Block Motion Estimation”, IEEE Trans. Circuits And Systems For Video Technology, vol 4., no. 4, pp. 438-442, August 1994.

[9]. 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.

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

[11]. C. K. Cheung and L. M. Po "Normalized partial distortion algorithm for block motion estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 3, pp.417 -422 2000

[12]. C. K. Cheung and L. M. Po "Adjustable partial distortion search algorithm for fast block motion estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 1, pp.100 -110 2003

[13]. W. G. Hong and T. M. Oh "Enhanced partial distortion search algorithm for block motion estimation", Electronic Letters, vol. 39, no. 15, pp.1112 -1113 2003

[14]. J.N. Kim , S.C. Byun , Y. H. Kim and B. H. Ahn "Fast full search motion estimation algorithm using early detection of impossible candidate vectors", IEEE Trans. Signal Process., vol. 50, no. 9, pp.2355 -2365 2002

[15]. B. Montrucchio and D. Qualgia “New Sorting based lossless motion estimation algorithms and a partial distortion elimination performance abnalysis” IEEE Trans. Circuits and Systems for Video Tech.,vol. 15, pp.210-220,Feb. 2005.

[16]. C. K. Cheung and L. M. Po, “Adjustable partial distortion search algorithm for block motion estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 13,no 1 pp. 100–110, jan 2003.

[17]. X. Yi and N. Ling, “Improved Normalized Partial Distortion Search with dual-halfway-stop for rapid block motion estimation”, IEEE Trans. on Multimedia, vol. 9, no. 5, pp. 995-1003, Aug. 2007.

[18]. C. C. Wang and C J Lo, “Using two stage sorting based partial distortion search for motion estimation in h.264/AVC” Optical Engineering, 46(9),09.7002,Sep, 2007

[19]. Sarwer, M.G.; Wu, Q.M.J.; , "Efficient partial distortion search algorithm for block based motion estimation," Electrical and Computer Engineering, 2009. CCECE '09. Canadian Conference on , vol., no., pp.890-893, 3-6 May 2009 doi: 10.1109/CCECE.2009.5090256

[20]. Sarwer, M.G.; Wu, Q.M.J.; , "Efficient Two Step Edge based partial Distortion Search for Fast Block Motion Estimation" IEEE Trans. Cosumer Electron, vol. 55, no 4, pp.2154-2162 Nov 2009.

[21]. T. H. Cormen , C. E. Leiserson , R.L. Rivest and C. Stein. Introduction to Algorithms, pp.168 -170 2001 :MIT Press and McGraw-Hill

