International Journal For Technological Research In Engineering
Volume 2, Issue 7, March-2015 *ISSN (Online): 2347 - 4718*

FAST BLOCK MATCHING ALGORITHM FOR MOTION ESTIMATION

Mr.Shohambhai B. Patel[1], Mr.Hitul Patel[2]
[1]PG Student, [2]HOD
Computer Engineering Department, (SCET, Saij), Kalol, Gujarat, India

Abstract: The digital video application has become increasingly popular in mobile terminals such as cellular phones and personal digital assistance. Due to inherent data intensity, storage and transmission of raw video data is difficult. With the limited storage capacity and bandwidth, this data must be compressed to a transportable size. In videos, the changes in the subsequent frames are very less, which causes redundancy. This encourages video compression resulting in efficient usage of storage devices. But this comes at the cost of some quality degradation. Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. In all the applications, the demand is high for accurate estimates of motion requiring minimal computational cost. In our work, we have discussed different motion estimation techniques. We have focused on Block Matching Algorithms like exhaustive search, three step search, new three step search, diamond search, hexagonal search and octagonal search. For these algorithms, performance in terms of accuracy (PSNR) and number of computations required to obtain motion vectors is discussed. In existing algorithms Hexagonal Search provides significant PSNR with least number of computations. Also we have tried to classify the motion. To modify the existing Hexagonal Search Algorithm, we have proposed a new approach for early termination based on threshold to avoid redundant calculation when in point obtain in earlier step is good enough. Comparison of Existing algorithm and proposed approach in terms of PSNR and Computations required has been carried out. Significant speedup gain has been achieved over the current hexagon-based search algorithm while maintaining similar distortion performance.
Keywords: PSNR, 2-D Image

I. INTRODUCTION

Video has become essential source of entertainment and part of our life. For the transmission of video and efficient storage of video in less memory, video compression is essential. For example if you are transmitting video over the internet, it is nearly impossible to transmit video in uncompressed format because of limited amount of available bandwidth.

A. Motivation

Video compression is needed in Internet Video streaming and TV broadcast where multimedia signals are transmitted over a fixed amount of limited bandwidth channels. Video compression is also useful when multimedia data is stored in memory devices like video-CDs or hard disk, which has limited storage capability. In case video is stored in CD without compression, it will barely store 5 to 8 minutes of video. In another example of, PAL system with standard TV resolution 720x576 pixels, and 8 bit RGB color scheme, the frame rate is 30 frames/second, and the data rate is 720x576x30x8x3=284 Mb/s. For HDTV with 1920x1080 pixels/frame, the data rate is 2.78 Gb/s. Such bandwidth and storage are not possible even with current powerful computer systems.[1] But fortunately video contains a lot of repetitive information for human capabilities. In Video compression, essentially the unnecessary information and repetitive data is discarded. Video compression allows transmitting and storing video information in a compact and efficient manner. There is a tradeoff between quality and compression. If compression ratio is increased, data size decreases but the quality also decreases. On the other hand, lower compression ratio contains more information that means more data size but it increases video quality. There are several standards developed for video compression.[1]

B. Redundancies in video

1) Redundancy: It is the duplication of critical components or functions of a system with the intention of increasing reliability of the system. There are two types of redundancies available in video which are explained as under.[1] 2) Spatial Redundancy Videos are nothing but sequences of images or frames with predefined frame rate. If you observe any frame, neighboring pixels have nearly the same brightness and color values, which is shown pictorially in fig.1.1. Instead of sending the same number for each and every sample, one number could be sent representing a block of sample points in an area where the information content remain same.

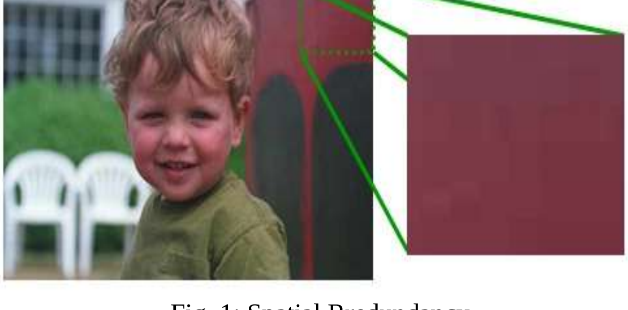

Fig. 1: Spatial Rredundancy

In videos, similar scenes exist between a frame and its adjacent frame. For example, consider frames from the video sequence caltrain as shown in fig. 1.2, here large numbers of pixels are same. In order to minimize the amount of information to be transmitted, block-based video coding standards encode the displaced difference block instead of the original block.

3) Temporal Redundancy

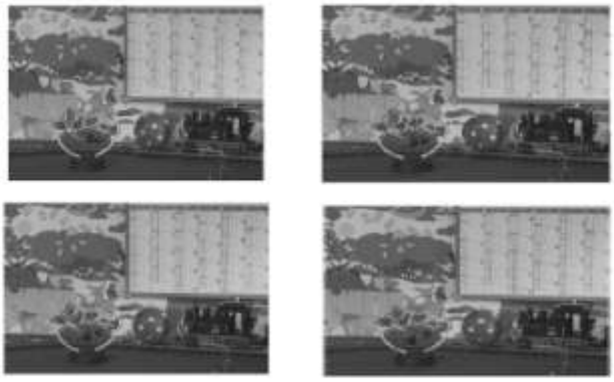

Fig. 2: temporal redundancy in caltrain video sequence

In videos, similar scenes exist between a frame and its adjacent frame. For example, consider frames from the video sequence caltrain as shown in fig. 1.2, here large numbers of pixels are same. In order to minimize the amount of information to be transmitted, block-based video coding standards encode the displaced difference block instead of the original block. Temporal redundancy can be exploited using Motion Estimation. In general, there are several approaches to motion estimation, and these can be divided as follows: region matching, gradient based methods and transform methods [1]. Block matching approach is mostly selected as the ME package in video coding applications and is also implemented in most existing video coding standards because of its simplicity and good performance. The steps in block matching are as follows

– Step1: Subdivide every frame into non-overlapping square blocks such as 16×16.

– Step2: Find out motion vector for each block.

– Step3: Within a search range, find a best match that minimizes an error measure.

C. Applications of Video Compression.

Videos are everywhere in our life. Applications of it can be broadly classified into below categories.

1) Conversational Applications

It includes Video calls and video conferencing.

2) Video streaming Applications

In video streaming applications, you don't have to wait for the entire video to be decoded. As frames come you can see that much part. Videos on YouTube are the best example of this.

3) Multimedia Applications

Broadcasting of video on television, Theaters, Video on demand etc. falls under this category. In all these applications it is not at all possible to deal with video without compression.

II. RELETED WORK

It is important to have a deep understanding on the existing algorithm of Motion Estimation. In this chapter we have mentioned several research papers based on different Block Matching Algorithms, implemented by various people to understand the work done so far in this domain. This chapter gives us a detailed and extensive description of the work done by various people and the various methods implemented.

A. Research Paper Literature Survey

Renxiang Li et al[3], proposed New Three Step Search (NTSS) algorithm for motion estimation, in this paper. Previous algorithm Three Step Search (TSS) uses uniformly allocated search point pattern in 1st step, which becomes inefficient for the estimation of small motion.. The features of NTSS are that it employs centre biased checking point pattern in 1st step, which is derived by making the search adaptive to the motion vector distribution and a half way stop technique to reduce the computational cost. Simulation results show that as compared to TSS, NTSS is much more robust, produces small motion compensation errors and has a very compatible computational complexity. In this paper M. Ghanbari[4] presented a fast block matching algorithm for motion estimation. It is based on a logarithmic step where in each search step only 4 locations are tested. For a motion displacement of w macroblocks/frame, this technique requires 5 + 4 2 log w computations to locate the best match. Using sequences of CIF standard pictures, the interframe motion compensated prediction error with this technique is compared to the other fast methods. The computational complexity of this algorithm is also compared against those methods. In this paper authors Lai-Man Po and Wing_chung Ma has proposed four-step search (4SS) algorithm[5]. Halfway-stop technique is employed in the new algorithm with searching steps of 2 to 4 and the total number of checking points is varied from 17 to 27. Simulation results show that the proposed 4SS performs better than the wellknown three-step search and has similar performance to the new three-step search (N3SS) in terms of motion compensation errors. In addition, the 4SS also reduces the worst-case computational requirement from 33 to 27 search points and the average computational requirement from 21 to 19 search points as compared with N3SS. Shan Zhu and Kai-Kuang proposed Diamond Search algorithm[6] in this paper. Based on the study of motion vector distribution from several commonly used test image sequences, diamond search (DS) algorithm for fast block-matching motion estimation (BMME) is proposed in this paper. Simulation results demonstrate that the proposed DS algorithm greatly outperforms the well-known three-step search (TSS) algorithm. Compared with the new three-step search (NTSS) algorithm, the DS algorithm achieves close performance but requires less computation by up to 22% on average. Experimental results also show that the DS algorithm is better than recently proposed four-step search (4SS) in terms of visual quality and required number of search points.

In this paper a novel algorithm using hexagonbased search (HEXBS) pattern for fast block motion estimation is introduced by Ce Zhu et al[8]. The proposed HEXBS algorithm may find any motion vector with fewer search points than the diamond search (DS) algorithm. The speedup gain of the HEXBS method over the DS algorithm is more striking for finding large motion vectors. Experimental results substantially justify the fastest performance of the HEXBS algorithm compared with several other popular fast algorithms. L.P.Chau and CeZhu[10], introduced a novel algorithm using octagon-based search (OCTBS) pattern for fast block motion estimation. The proposed OCTBS algorithm can find a motion vector with fewer search points than the diamond search (DS) algorithm. The speedup gain of the OCTBS method over the DS algorithm is more striking for finding large motion vectors. Experimental results justify the performance of the OCTBS algorithm compared with several other popular search algorithms. In this paper a new technique called Fast Computations of Full Search (FCFS) algorithm has been introduced by Zaynab Ahmed et al. [12]. This technique keeps the resolution of the decompressed videos the same as the one generated from using Full Search Block Matching Algorithm while decreasing the computational time required to determine the matching macroblock from the reference frame to the current macroblock. This is performed by stopping the calculation of the sum absolute different between the pixels in the current macroblock and the macroblocks in the reference frame when the current uncompleted sum absolute value is greater than the previous calculated one. Experimental results of various video sequences types showed that the proposed technique reduces the search time of the macroblock matching, and keeps the resolution exactly the same as Full Search. S. Immanuel Alex Pandian et al[13], reviewed different block matching algorithms used for motion estimation in video compression. It compares 10 different types of block matching algorithms that range from the very basic Full Search to the recent fast adaptive algorithms like Pattern Based Search. The algorithms that are evaluated in this paper have been used in implementing various standards ranging from MPEG1 / H.261 to MPEG4 / H.263. In the overview of these techniques, they have discussed various block matching algorithms that range from the very basic Full Search to the recent fast adaptive algorithms like Pattern Based Search. Normally Motion Estimation is quite computationally intensive and can consume up to 80% of the computational power of the encoder if the full search (FS) is used by exhaustively evaluating all possible candidate blocks within the search window. As a consequence, the computation of video coding is greatly reduced with pattern based block motion estimation. In this paper authors ChittranjanPradhan and DipannitaAdak[14], have discussed the commonly used motion estimation algorithms such as- Full Search (FS), Three-Step Search (TSS), New Three-Step Search (NTSS), Four-Step Search (FSS), Diamond Search Algorithm (DS), and Hexagon Based Search Algorithm (HEXBS). They have also analyzed these techniques by using Peak Signal to Noise Ratio (PSNR) values. From this analysis, they have found that the full search (FS) technique produces better quality image as it gives better performance in PSNR calculation, but takes larger number of search points; whereas, diamond search (DS) and hexagon-based search (HEXBS) algorithms take a few numbers of search points and also give average performance in PSNR calculation. Other algorithms (i.e. TSS, NTSS, and FSS) take lesser number of search points, but produce distorted image because of poor PSNR performance. As DS and HEXBS algorithms take fewer number of search points, they are faster. The image quality of DS and HEXBS can be improved by increasing the PSNR values. Therefore, getting a better quality image by reducing number of search points remains a goal.

B. Summary

After studying research papers discussed in this chapter, the conclusions are summarised in the table shown below.

METHOD	ADVANTAGE	DISADVANTAGE
Exhaustive Search(ES)	Maximum PSNR	Too much computations required
Cross Search Algorithm(CSA)	Less Computations as compared to Es	Low PSNR
Three Step Search (TSS)	Less computations as compared to ES	Low PSNR
New Three Step Search (NTSS)	Less computations as compared to TSS	Low PSNR
Four Step Search(FSS)	Less computations as compared to NTSS	Low PSNR
Diamond Search (DS)	Less computations as compared to FSS PSNR similar to ES	-
Hexagonal Search (HS)	Minimum Computations Significant PSNR	-
Octagonal Search(OS)	Less computations as compared to DS	Low PSNR as compared to DS

Table. 1: Comparison of existing Block Matching Algorithms

C. Problem Definition Formation

After studying Block Matching Algorithms like Exhaustive Search, Three Step Search, New Three Step Search, Four Step Search, Diamond Search, Hexagonal Search and Octagonal Search we came to conclusion that Exhaustive Search Provides best PSNR but it requires highest number of computations to obtain Motion Vectors. On the other hand Hexagonal Search provides better result as compare to other methodologies. It requires least number of computations to obtain significant PSNR. But even in this method one problem is that no stopping technique is there when the point obtained in earlier steps is good enough. So in this dissertation work we have focused on threshold based early termination algorithm.

III. PROPOSED WORK

In the previous section, it has been observed that Hexagonal Search is best among all the Block Matching Algorithms. As explained earlier in chapter 4, the hexagonal search algorithm follows iteration until the same winner node comes twice. But it is not necessary to check the points when the earlier winning point is good enough. Therefore, we set an adaptive threshold at the end of search after the winning point meets the requirements. There are basically two options for the threshold, fixed threshold and variable threshold. With fixed threshold again problem arises when threshold is small and motion is high in current frame. So in such case, not sufficient computations can be saved. So we have proposed variable threshold as per the motion of the current frame.

A. Classification of Motion

Motion classification of moving objects or scenes among the set of frames has been carried out with the help of parameter defined as 'Motion Factor'.

$$\text{motion factor}=\frac{\sum|\text{motion vectors}|}{\text{effective no. of blocks}}$$

'Motion Factor' is a newly introduced term by us which is mathematically defined as equation 6.1. Based on calculation of this parameter, motion can be classified as

1) Low motion when reference frame and current frame both are almost similar.
2) Average motion when there are some changes in reference frame and current frame.
3) High motion when lots of changes are there in reference frame and the current frame
4) Scene change when reference frame and current frame are from different scenes

Always some movement has been captured in the border macro blocks of the frame. So, to avoid the false detection of motion, 2 macro blocks from each side of the current frame are not considered in the calculation of motion factor, which is shown pictorially in fig. 3.1.

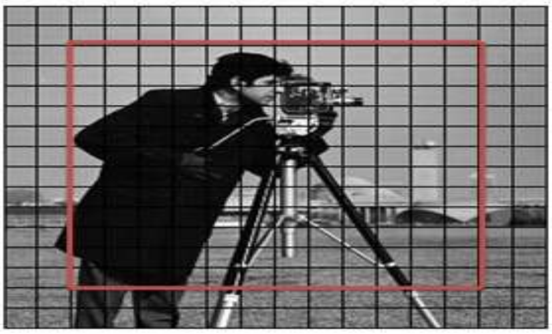

Fig. 3: Effective macroblocks of frames to calculate motion factor

Frames from the video sequence 'salesman' to calculate motion can be seen in fig. 3.2. It can be observed that here only little bit change is there in facial expression and hand movement. And most of the contents remain same. That's why motion factor obtained here is 0.0783, which indicates low motion.

Fig. 4: (a) reference frame (b) current frame from video sequence salesman

Frames from the video sequence 'garden' to calculate motion can be seen in fig. 3.3. It can be observed that here reference frame is shifted version of the current frame. That's why motion factor obtained here is 3.3, which indicates average motion

Fig. 5: (a) reference frame (b) current frame from video sequence garden

Frames from the video sequence 'football' to calculate motion can be seen in fig. 3.4. It can be observed that here lots of changes are there in both the frames. That's why motion factor obtained here is 4.22, which indicates high motion.

Fig. 6: (a) reference frame (b) current frame from video sequence football

Frames from the video sequence 'tennis' to calculate motion can be seen in fig. 3.5. It can be observed that here no similarities are there. That's why motion factor obtained here is 5.66, which indicates scene change.

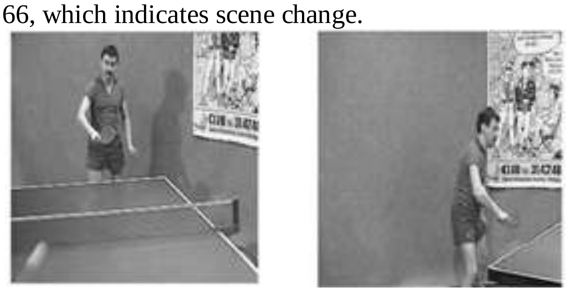

Fig. 7: (a) reference frame (b) current frame from video sequence tennis

Tabular form of different motions considering motion factor is available in table 3.1

Figure	Motion factor	Comment
6.2	0.0783	Low motion
6.3	3.3	Average motion
6.4	4.22	High motion
6.5	5.66	Scene change

Table .2: classification of motion using motion factor

B. Proposed Approach

In our approach to obtain the threshold, we have used the curve fitting approach. Curve fitting is a process for finding the equation of the curve for best fit which may be most suitable for predicting unknown values. There are several methods for curve fitting, but the least square error method provides a unique equation, so we have used the same. In this method the sum of square of error is minimum.

C. Steps for proposed approach

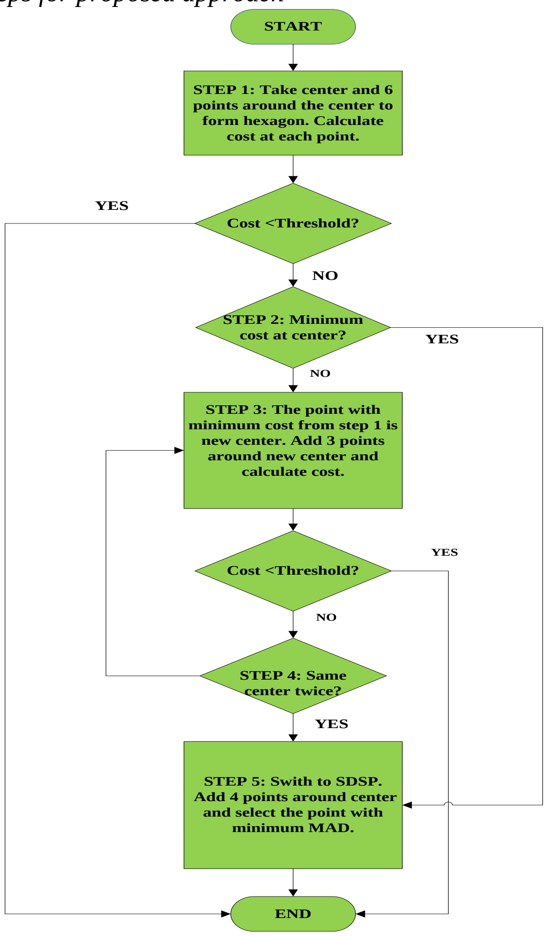

Fig. 8: flowchart for Modified Hexagonal Search

Step 1: Calculate the threshold by using equation 3.2. It remains same for the entire frame. The large hexagon with seven checking points is formed at the center of a predefined search window. Calculate cost of each of the 7 points. If minimum cost is less than the threshold then end the procedure.

– Step 2: If minimum cost is found to be at the center of the hexagon, go to step 5.

– Step 3: With the minimum cost point in the previous search step as the center, a new hexagon is formed. Three new candidate points are checked, and the minimum cost point is again identified. That is the new center. If the minimum cost is less than the threshold then end the procedure otherwise proceed to step 2.

– Step 4: Repeat step 3 until you get the same center in two successive iterations. And proceed to step 5.

– Step 5: Switch the search pattern from large to small size of diamond. Add 4 points around the new center and calculate the cost. The new minimum cost point is the final motion vector. minimum cost point is the final motion vector.

IV. CONCLUSION

We have proposed a novel approach which will decide threshold for the MAD according to the motion and terminate automatically when the MAD is less than threshold. Significant speedup gain has been achieved over the current hexagon-based search algorithm while maintaining similar distortion performance. The proposed approach consistently performing faster convergence search performance as compared to existing Hexagon Search, irrespective of low, medium or high motion.

REFERENCES

[1] M. Manikandan, P. Vijayakumar and N. Ramadass, "Motion estimation method for video compression - an overview", Wireless and Optical Communications Networks, IFIP Intl' Conference, pp. 5-11, 13 April 2010.
[2] X. Jing and L. P. Pui., "An efficient three-step search algorithm for block motion estimation."IEEE Transactions on Multimedia, volume 6, 2004, pp. 435 - 438.
[3] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Transactions Circuits Syst. Video Technoolgy., volume 4, Aug. 1994, pp. 438–442.
[4] M.Ghanbari, "The Cross-Search Algorithm for Motion Estimation", IEEE Transactions on Communications, Vol. 38, No. 7, July 1995.
[5] L M Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation", IEEE Transactions Circuits System Video Technology, vol. 6, June 1996, pp. 313-317.
[6] Shan Zhu and Kai-Kuang Ma. "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," IEEE Transactions on Image Processing, Vol. 9, No. 2, pp.287-290, February

2000.
[7] Chun-Ho Cheung, and Lai-Man Po, “A Novel CrossDiamond Search Algorithm for Fast Block Motion Estimation”, IEEE Transactions Circuits And Systems For Video Technology, vol12., no.12, December 2002, pp. 1168-1177.
[8] Ce Zhu, Xiao Lin, and Lap-PuiChau. "Hexagon-Based Search Patten for Fast Block Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, Vol.12, No.5, May 2002,pp.349-355.
[9] Ce Zhu, X. Lin, L. Chau, and L. M Po., “Enhance hexagonal search for fast block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, 2004, pp.1210 - 1214.
[10] L. Chau and Ce Zhu, “A Fast Octagon Based Search Algorithm for Motion Estimation”, Signal Processing, Volume 83 Issue 3, March 2003, pp.671-675.
[11] Liang Yaling Liu Jing Du Minghui , , “A Cross Octagonal Search Algorithm For Fast Block Motion Estimation”, International Symposium on Intelligent Signal Processing and Communication Systems, December 13-16, 2005 Hong Kong, pp.357-360.
[12] ZaynabAhmed ,AbirJaafarHussain and DhiyaAlJumeily, “Fast Computations of Full Search Block Matching Motion Estimation (FCFS)”, ISBN: 978-1- 902560-25-0 © PGNet(2011)
[13] S. Immanuel Alex Pandian, Dr.G. JoseminBala and Becky Alma George, “A Study on Block Matching Algorithms for Motion Estimation”, International Journal on Computer Science and Engineering, Jan 2011.
[14] ChittranjanPradhan and Dipannita Adak, “Survey on Block Matching Algorithms using Motion Estimation”, International Journal of Computer Applications (0975 – 8887, Volume 46– No.16, May 2012,pp.6-10.
[15] BorkoFurhut, “A survey on multimedia compression technique and standards”, Department of computer science and engineering, Florida Atlantic university,2005.
[16] Wei-Yi Wei, “Digital Video Compression, Fundamentals and standards ” , National Taiwan University, Taipei, Taiwan, ROC
[17] Yung-Ming Chou and Hsueh-Ming Hang,” A New Motion Estimation Method Using Frequency Components”, Journal of Visual Communication and Image Representation Vol. 8, No. 1, March, 1997.
[18] Ming Fai Fu, Oscar Au, Wing Cheong Chan,” Temporal Interpolation using Wavelet Domain Motion Estimation and Motion Compensation”, IEEE ICIP, 2002.
[19] Min Li, MainakBiswas, Sanjeev Kumar and Truong Nguyen, “DCT-Based Phase Correlation Motion Estimation”, UCSD, ECE Dept., La Jolla CA 92093.
[20] M. Manikandan, P. Vijayakumar and N. Ramadass, “Motion estimation method for video compression - an overview”, Wireless and Optical Communications Networks, IFIP Intl’ Conference, pp. 5-11, 13 April 2006.

