International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 10, October - 2017

Adaptation of Motion Estimation Algorithms for Real Time Video Sequences

S. S. Abdelaal[a],
[a]Computer Engineering Department
College of Engineering and Technology
Arab Academy of Science and Technology and Maritime Transport-
Heliopolis - Cairo - Egypt

***Abstract* - The paper proposed an evaluation of adaptation of motion estimation algorithms, to the type of motion detected in real time video sequences. Motion speed variations in frames occur during real time video streaming. Certain algorithms can be be applied on the video frames according to the type of motion detected in these frames. So in case fast motion is detected , fast motion estimation algorithms that are suitable to that type of motion like the fast adaptive rood pattern search algorithm arps are applied, and in case slow motion is detected in the frames ,slow motion estimation algorithms that are suitable to that type of motion like the three step search algorithm tss are used. So adaptation of motion estimation algorithms to the type of motion detected in the video scenes is required to be applied till the end of real time video streaming.this adaptation process is done continuously eight methods have been discussed and implemented ranging from the very basic exhaustive search to the fast adaptive algorithms, and they are evaluated and compared together in terms of psnr and computational complexity**

Keywords--- Hybridization; block matching; motion estimation; tradeoff.

1. INTRODUCTION

Performing motion estimation in real time with acceptable computational time and a high image quality is a major challenge, because of the intensive computations and the large amount of resources required by motion estimation. It has been an active research field in the past two decades and various algorithms have been developed to reduce the computation and improve the performance. Most of the algorithms developed for motion estimation so far are block-based techniques, called block-matching algorithms (BMA). Block matching algorithms for motion estimation have been widely adopted due to their effectiveness and simplicity for implementation.

The full search algorithm (FS) [1] is the simplest block matching algorithm that can deliver the optimal estimation solution, however it yields an extremely computational expensive BM method that seriously constraints its use for real-time video applications. Several BM algorithms have been proposed considering the following two techniques:

(1) Using a fixed pattern search strategy: the search operation is conducted over a fixed subset of the total search window. The Three Step Search (TSS) [2], the New Three Step Search (NTSS) , the Simple and Efficient TSS (SES) [3], the Four Step Search (4SS) [4] and the Diamond Search (DS) [5], all represent some of its well-known examples. They are not able to eventually match the dynamic motion-content, sometimes deliver false motion vectors resulting in image distortion.

(2) Using adaptive search strategy: the algorithm chooses as search points only those locations that iteratively minimize the error-function. This category includes the Adaptive Rood Pattern Search (ARPS) [6], which is BMA based on motion vector predictor and has a superior performance.

In fact, motion speed in real time videos can be slow, fast or a mix of both. For example, we cannot know exactly what the behavior of motion speed is in a broadcasting real-time video sequence. It is a challenge to select the most suitable algorithm to be applied on the video sequence, this selection is according to the type of motion speed which changes many times in the same video sequence, so as to achieve the best reasonable results in accuracy and computational speed.

The tradeoff between complexity in computations and accuracy is the most important issue in selecting the proper BMA. The algorithm which gives higher accuracy will need high computations and vice versa. This is the motivation of hybridization [7,8] between algorithms, to gain low computational complexity keeping reasonable accuracy.
in the same video sequence.
So the most important idea to be discussed is the selection between algorithms to be applied on real time video sequences according to the type of motion speed in these sequences.

The remaining paper is organized as follows: in section 2, we give an overview of the implemented techniques involved in block-matching algorithms, discussions about the approaches of scene detection , the algorithms to be applied for slow motion and those that can be applied for fast motion

Section 3 presents the simulation results for evaluations and comparisons between the different block matching algorithms . Finally, the conclusions are drawn in section 4.

2. LITERATURE REVIEW

The idea behind block matching [9] as shown in fig. 1 is to divide the current frame into a matrix of ‘macro blocks’ that are then compared with corresponding block and its neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement is calculated for all the macro blocks comprising a frame; it constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to *p* pixels on all fours sides of the corresponding macro block in previous frame. . In this section, we describe several block matching algorithms which are commonly used:

2.1. Full Search (FS)

This algorithm, also known as Exhaustive Search, it is the most computationally expensive block matching algorithm of all. It calculates the cost function at each possible location in the search window. It finds the best possible match and gives the highest PSNR amongst any block matching algorithms. Fast block matching algorithms try to achieve the same PSNR doing as little computations as possible. The obvious disadvantage of FS is that the larger the search window gets the more computations it requires. Computation points in FS always equal$(2 * p + 1)^2$.FS is not suitable for real time applications.

2.2. Fast BMA using a fixed set of search patterns

Such approaches assume that the error-function behaves monotonically; holding well for slow-motion sequences but failing for other kinds of movements in video sequences, however all above methods rely on the unimodal error surface assumption (UESA) which is usually not established for real video sequences. Therefore, all these algorithms may get trapped into local optimum easily, especially under the condition of complex movement making the algorithm prone to get trapped into local minima.

Simplicity and regularity are the most important advantages of these methods, which make them attractive for implementation. However, they have less adaptability and search efficiency in tracking large motions. Some examples of these algorithms are implemented here as:

2.2.1. Three Step Search(TSS)

This is one of the earliest attempts of fast block matching algorithms and dates back to mid-1980. The general idea is represented in Fig. 2. It gives a flat reduction in computation by a factor of 9. So that for p = 7, FS will compute cost for 225 macro blocks whereas TSS computes cost for 25 macro blocks. The idea behind TSS is that the error surface due to motion in every macro block is unimodal. This is the main disadvantage of TSS as it may be trapped into a local minimum as the error surface is not restricted to be unimodal.

2.2.2. New Three Step Search (NTSS)

NTSS improves on TSS results by providing a center biased searching scheme and having provisions for half way stop to reduce computational cost. It was one of the first widely accepted fast algorithms and frequently used. The NTSS has the same disadvantage like TSS as it assumes unimodal error surface.

2.2.3. Simple and Effecient Search(SES)

SES is another extension to TSS and exploits the assumption of unimodal error surface. The main idea behind the algorithm is that for a unimodal surface there cannot be two minimums in opposite directions. Although this algorithm saves a lot of computation as compared to TSS, it was not widely accepted for two reasons. Firstly, in reality the error surfaces are not strictly unimodal and hence the PSNR achieved is poor compared to TSS. Secondly, there was another algorithm, Four Step Search, that had been published a year before that presented low computational cost compared to TSS and gave significantly better PSNR.

2.2.4. Four Step Search(4SS)

Similar to NTSS, 4SS also employs center biased searching and has a halfway stop provision. This search algorithm has a best case scenario of 17 checking points and worst case of 27 checking points. But also inherits the same accuracy features from TSS.

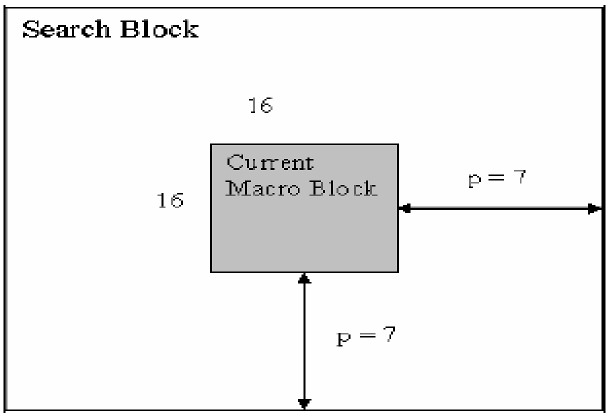

Fig. 1.Block matching a macro block of side 16 pixels and a search parameter p of size 7 pixels.

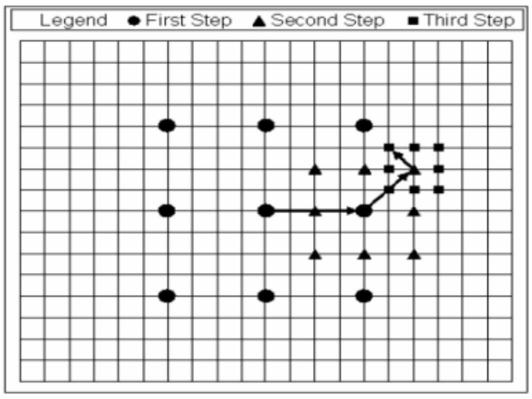

Fig. 2. Three Step Search procedures. The motion vector is (5, -3).

2.2.5. Diamond Search(DS)

DS algorithm is similar to T SS, but the search pattern is changed from a square to a diamond. Moreover, there is no limit to the number of steps that the algorithm can take. DS uses two different types of fixed patterns, namely, large diamond search pattern (LDSP) and small diamond search pattern (SDSP), as shown in Fig. 3. LDSP is utilized first until the least cost is found at the center of the pattern. Then SDSP is employed in the last step. The resultant accuracy was better than any other fixed pattern algorithm.

2.3. Adaptive search patterns

The adaptive search patterns [10] try to solve the above mentioned issues depending on that in most cases, adjacent MBs belonging to the same moving object will almost have similar motions. Therefore, the current block’s motion behavior can be reasonably predicted by referring to its neighboring blocks’ MVs in the spatial and/or temporal domains.

2.3.1 The adaptive rood pattern search ARPS

The adaptive rood pattern search ARPS is an example of this category. As for the second issue, the adaptive rood pattern (ARP) with adjustable rood arm (thus, the pattern size), that is dynamically determined for each MB according to its predicted motion behavior is used.

Adaptive rood pattern search (ARPS), consists of two sequential search stages, as shown in fig.4 : 1) initial search and 2) refined local search. For each macro block (MB), the initial search is performed only once at the beginning.

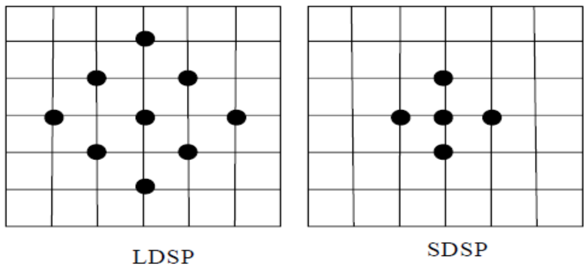

Fig. 3. LDSP and SDSP search patterns.

Extensive experiments show that the search speed of ARPS is better than the Fast BMA Using a Fixed Set of Search Patterns and achieves reasonable peak signal-to-noise ratio (PSNR) particularly for those video sequences containing large and/or complex motion contents. By finding a good starting point for the follow-up refined local search, unnecessary intermediate search and the risk of being trapped into local minimum matching error points could be greatly reduced in long search case. 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. Fig. 4 shows how the ARPS finds the step size used in second stage based on four arms and a predicted vector.

The main advantage of this algorithm over DS is that if the predicted motion vector is (0, 0), it does not waste computational time in doing LDSP, it rather directly starts using SDSP. Furthermore, if the predicted motion vector is far away from the center, then again ARPS save on computations by directly jumping to that vicinity and uses SDSP, whereas DS takes its time doing LDSP.

2.3.2 motion speed-based adaptive algorithm (MSBAA)

In real sequences, the kinds of motion vary so many times and the motion speed also differs. They may have scene changes as they consist of many concatenated small scenes. As real time videos should have a mix of all kinds of motion speeds, so the motion speed-based adaptive algorithm (MSBAA) determines the type of motion speed in the video sequence in advance, and then it selects the suitable algorithm to be applied to the number of tested frames according to the type of motion speed in them. This mechanism is applied to the video, frame by frame till the video streaming ends, so the first step in (MSBAA)[11] is to use a simple detector to detect the difference between the two successive frames. The detector is pixel based as it uses the mean absolute frame difference (MAFD)[12] shown in equation (1). MAFD gives an indication of the type of motion speed . This detector is well known in the researches in the field of scene change detection .The main advantage of this detector is that it has extremely low computational cost.

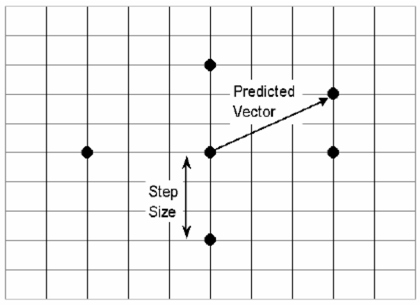

Fig. 4. Adaptive Rood Pattern: The predicted motion vector is (3,-2), and the step size S = Max (|3|, |-2|) = 3.

$$MAFD_n = \frac{1}{MN}\sum_i^{M-1}\sum_j^{N-1}|f_n(i,j) - f_{n-1}(i,j)| \qquad (1)$$

Where M and N are the width and height of the frames, $f_n(i, j)$ is the pixel intensity at position (i, j) and n is the frame number. MAFD corresponds to the first-order derivative of f_n and it measures the degree of dissimilarity at every frame transition.

A threshold value is suggested for this detector which is equal to 14. This detector is familiar and was previously used in many researches for scene change detection. It is used in the (MSBAA) algorithm as an important tool to determine the type of motion speed in the streaming video frames.

The algorithm illustrated as followed: **firstly**, zero motion prejudgment (ZMP) is used to stop computations in case of static blocks using a threshold T=2 when using mean absolute difference MAD. **Secondly**, performing full search algorithm in all blocks in the first row and column. **Thirdly**, performing ARPS in the rest of the blocks of the frame. The search structure of the slow motion algorithm is shown in fig. 5

The reason for using FS here in the slow motion hybrid algorithm is that, the ARP's size is equal to the length of the predicted MV (i.e., the MV of the immediate left block of the current block). That is, the predicted motion vector is of greatly high importance here to achieve accuracy. So the motion vectors of the initial blocks (blocks of first row and column) are so important that it will be appreciated to use FS to get optimal motion vectors at the starting point for ARPS that should be applied to the rest of the blocks in the frame

The complexity of computations increased because of using FS in many blocks (depending on the image size) but not much, because at least half of the search window of the blocks in first row and column is outside the frame. But that yields higher PSNR.

In videos that have small motion the total number of static blocks per frame could be easily as high. Thus, significant additional reduction in computational cost is possible if we perform a zero-motion prejudgment (ZMP) at the beginning of ME.

This algorithm is suitable for video sequences having slow motion as shown in next section. It shows less computational complexity on the average due to the effect of using ZMP, also it gives higher accuracy. On testing video sequences having fast/complex motion, we found that ARPS is better as it gives better computation with reasonable accuracy.

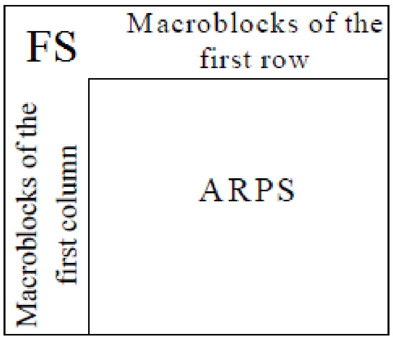

Fig. 5. Search structure of the slow motion hybrid FS&ARPS algorithm.

3. SIMULATION RESULTS AND DISCUSSIONS

We have implemented eight algorithms in this paper to compare them and show how the tradeoff is handled between computational speed and PSNR.

Peak-Signal-to-Noise-Ratio (PSNR) is used as the parameter referring to accuracy of the algorithm. It is given by equation (2), it characterizes the motion compensated image that is created by using motion vectors and macro blocks from the reference frame.

$$PSNR = 10Log_{10}\left[\frac{(Peak\ to\ peak\ value\ of\ original\ data)^2}{MSE}\right] \qquad (2)$$

Eight block matching algorithms consisting of eight well known algorithms which are (full search (FS),three step search(TSS), simple and efficient three step search (SETSS), new three step search (NTSS), four step search(FSS), diamond search(DS),adaptive rood pattern search (ARPS))and the motion speed-based adaptive algorithm (MSBAA) are applied on three video sequences respectively so as to compare the results of every applied algorithm with respect to PSNR and the number of searching points per frame which reveals the computation metric.

Table 1 shows a comparison in computation points and PSNR when applying the algorithms on “Ship” video sequence. Ship sequence has 300 frames having slow motion and a lot of static blocks. According to table 1, the MSBAA algorithm has the least average computation points, and it gets better results. All algorithms have PSNR values near to that of FS; this assures that the MSBAA algorithm is the most suitable for slow motion video sequences.

Table 1 shows that the motion speed-based adaptive (MSBAA) algorithm saved a lot of computations in real time and kept the PSNR. The other center based algorithms (TSS, SESTSS, NTSS, 4SS, DS) have nearby values of PSNR but higher computation points, so they are not attractive to real time applications.

All algorithms have been tested against two very big sequences of more than 2600 frames which contain different scenes with different kinds of motion speed. The results are attached in table 2 which shows the comparison between all algorithms with respect to the number of searching points per frame and PSNR respectively. In first 30 frames which have no motion, the fixed pattern algorithms waste so much time as they have fixed values of computation points, but the (MSBAA) algorithm makes zero computations as the detector will detect in this case that (MAFD = 0).i.e. there is no motion.

When the motion is slow with immobile blocks in the frame, the fixed pattern search algorithms have to pay their fixed computation point per each block within the frame, but the (MSBAA) algorithm pay only one computation point as it uses ZMP property for early stop to the search.

When the motion is detected to be fast or complex, the motion speed-based adaptive (MSBAA) algorithm will select ARPS to be applied on these fast frames. As table 2 the average searching points/frame for all the frames will be so near to ARPS algorithm. This means that the (MSBAA) algorithm has detected that most of the motion speeds in the frames of this video sequence is fast and accordingly the (MSBAA) algorithm selects ARPS to be applied on these fast motion frames, so we get number of search points very close to that of ARPS algorithm.

Table 1. Average PSNR and average computation points per frame of “ship” video sequence

Algorithm	PSNR	Average no. of searching points/frame
FS	36.64110849	204.2828283
TSS	36.64087692	23.22512438
SETSS	36.64071148	17.08462737
NTSS	36.64106525	15.96114126
4SS	36.6408834	15.87300869
DS	36.64102387	12.30519122
ARPS	36.64181829	5.095180662
MSBAA algo.	36.64049221	4.1315644

Table 2. Average PSNR and average computation points per frame of a big mixed video sequence (2600 frames)

Algorithm	PSNR	Average no. of searching points/frame
FS	30.10403959	204.2042882
TSS	29.73211626	23.35164408
SES	29.51807144	16.38235682
NTSS	29.96400838	19.44239201
4SS	29.8054743	17.92645894
DS	29.86203986	15.80063321
ARPS	29.96704198	7.747574748
MSBAA algo.	29.9679944	7.580641026

4. CONCLUSIONS

The tradeoff between computational complexity and PSNR is the most important parameter in choosing which algorithm to be implemented in real-time applications to achieve good reduction in computational speed with reasonable psnr

The used fixed pattern algorithms works well for slow motion frames, only the adaptive rood pattern algorithm is suitable for fast motion

For future work it can be thought that of new algorithms that can handle fast motion putting camera movements and scene detection parameters into consideration

REFERENCES

[1] Aroh Barjatya, “Block Matching Algorithms For Motion Estimation,” *ECE 6620, Digital Image Processing,Utah State University, USA, 2004.*

[2] 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.

[3] Jianhua Lu, and Ming L. Liou, “A Simple and Effecient Search Algorithm for Block-Matching Motion Estimation”, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 7, No. 2, pp. 429-433, April 1997.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 10, October - 2017

[4] 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.

[5] 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.

[6] 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.

[7] Jidong Ma, Zhaowei Gu, Fuming Qu, Gongjian Shen, Anna Wang, "A novel block-matching algorithm for motion estimation", College of Information Science and Engineering, Northeastern University , 2004.

[8] Santosh ku. Shhotray, Dheeraj Kannoujia and Samir Hu Jha, "an efficient block matching algorithm for fast motion estimation using combined three step search and diamond search algorithm", *International Journal of Computer & Communication*, Vol. 3, No. 6-8, 2012.

[9] M. R. Khammar, "Evaluation of different block matching algorithms to motion estimation", *International Journal of VLSI and Embedded Systems-IJVES* , Vol. 03, No. 03, July-August 2012

[10] Ravindra Kr Purwar, Navin Rajpal, "A fast block motion estimation algorithm using dynamic pattern search", *Signal, Image and Video Processing*, Vol. 7, No. 1 , pp 151-161. January 2013.

[11] S. S. Abdelaal, M. B. Abdelhalim, A. A. Hegazyb. " A New Speed-Based Adaptive Motion Estimation Algorithm For Video Sequences", in 12th international *conference in signal processing,* ICSP2014 Proceedings, pp. 637-642. IEEE,2014.China,Hangzu

[12] Xiaoquan Yi and Nam Ling, "Fast Pixel-Based Video Scene Change Detection", *IEEE International Symposium on Circuits and Systems, ISCAS'05*, Kobe, Japan, Vol. 4, 3443 – 3446, May 2005.