Kiran al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 1, Issue 2, September 2014, pp. 149-153

A Modified Full Search Block Matching Algorithm with Reduced Number of Search Positions per Block per Frame

Kiran[1], Dr. Kamal Sharma[2], Sharad Chauhan[3]

[1]Student, M. Tech, ESEAR, Ambala
[2]Professor, Dept. of ECE, E-Max group of Institutions, Ambala
[3]Asisstant Professor, Dept. of CSE, E-Max group of Institutions, Ambala

***Abstract*— Full Search Block Matching algorithm (FSBM) which is one of the primitive algorithm for video compression and motion detection is being implemented in the current work. Further, FSBM has been modified (MBM) by computing Mean square Error (MSE) between consecutive frames and avoiding a wide number of iterations required by checking where this value comes out to be zero . Experimentally, the performance of Modified Block Matching(MBM) for two videos is evaluated .First video is traffic situation in day time and the other one for night. Following two parameters are chosen:**

1. **Execution time per frame.**
2. **Number of iterations per frame.**

Selections were made at thresholds 4500,5000,6000 .It is the value that is need to be set to get accurate results for different cameras like daylight and night used under different circumstances.

***Keywords*— Motion Detection, FSBM, MBM.**

I. Introduction

The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame. The idea behind block matching is to divide the current frame into a matrix of 'macro blocks' that are then compared with corresponding block and its adjacent 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 calculated for all the macro blocks comprising a frame, 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. This 'p' is called as the search parameter. Larger motions require a larger p, and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macro block is taken as a square of side 16 pixels, and the search parameter p is 7 pixels. The idea is represented in fig 1. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block.There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (i). Another cost function is Mean Squared Error (MSE) given by equation (ii).

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

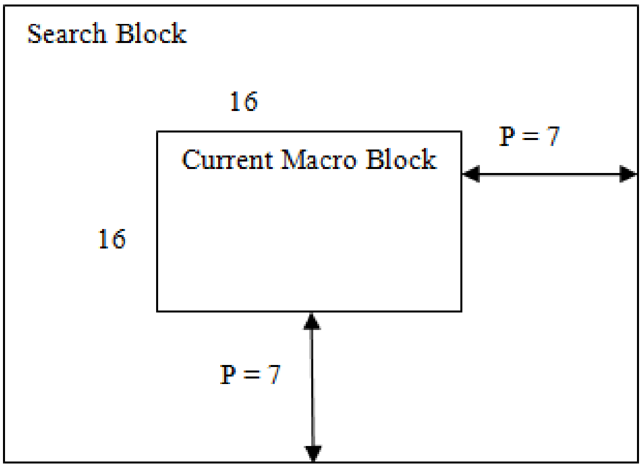

Fig 1 Macro Block with Result Window

$$\text{MSE} = \frac{1}{N^2}\Sigma_{i=0}^{N-1}\Sigma_{j=0}^{N-1}(C_{ij} - R_{ij})^2 \qquad \text{(ii)}$$

Where N is the side of the macro bock, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively. [7]

1.1 OBJECT TRACKING

Object tracking is an important task within the field of computer vision. The proliferation of high powered computers, the availability of high quality and inexpensive video cameras, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. There are three key steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior.

It is being applied for

Video indexing, traffic monitoring, automated surveillance, motion based recognition. In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as orientation, area, or shape of an object. Tracking objects can be complex due to partial and full object occlusions, noise in images, complex object motion, scene illumination changes.

One can simplify tracking by imposing constraints on the motion and/or appearance of objects. For example, almost all tracking algorithms assume that the object motion is smooth with no abrupt changes. One can further constrain the object motion to be of constant velocity or constant acceleration based on a priori information. Prior knowledge about the number and the size of objects, or the object appearance and shape, can also be used to simplify the problem. Numerous approaches for object tracking have been proposed.

These primarily differ from each other based on the way they approach the following questions: Which object representation is suitable for tracking. Which image features should be used. How should the motion, appearance, and shape of the object be modeled.

The answers to these questions depend on the context/environment in which the tracking is performed and the end use for which the tracking information is being sought. A large number of tracking methods have been proposed which attempt to answer these questions for a variety of scenarios. [15]

1.2 Motion Detection

Motion detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. Motion detection can be achieved by both mechanical and electronic methods. When motion detection is accomplished by natural organisms, it is called motion perception.

Usually detection is achieved by

Infrared (Passive and active sensors),Optics (video and camera systems) ,Radio Frequency Energy (radar, microwave and tomographic motion detection),Sound (microphones and acoustic sensors),Vibration (triboelectric, seismic, and inertia-switch sensors),Magnetism (magnetic sensors and magnetometers).

A large proportion of research efforts of object detection and tracking focused on this problem in last decade. Compared with object detection without motion, on one hand, motion detection complicates the object detection problem by adding objects temporal change requirements, on the other hand, it also provides another information source for detection and tracking. [9]

1.3 Block Matching

A Block Matching Algorithm (BMA) is a way of locating matching blocks in a sequence of digital Video frames for the purposes of motion estimation. The purpose of a block matching algorithm is to find a matching block from a frame in some other frame, which may appear before or after. This can be used to discover effectiveness of inter frame video

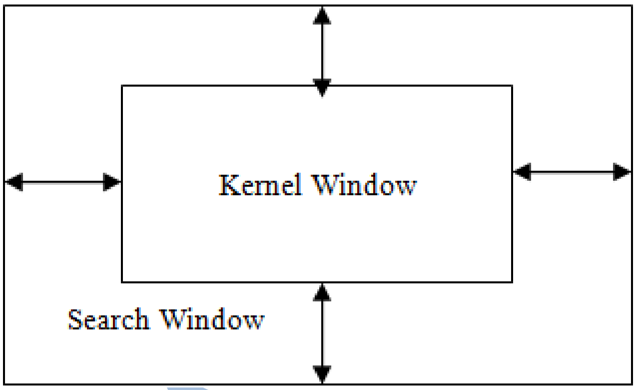

Fig 2 Block Matching Concept

Compression and motion detection. Block matching algorithms make use of an evaluation metric to determine whether a given block in frame matches the search block in frame. [7]

1.4 Problem Formulation

We have chosen FSBM algorithm for motion detection of pixels. While we use FSBM we visit each and every pixel whether it is in motion or not. Due to this there is lot of wastage of time and it also increase the number of iterations. So we focus on To minimize the time wastage and number of iterations so that we can reduce the processing time for frame as well as for whole video.

II. EXISTING SOLUTION

According to Love et.al [1] Basic need for event detection and tracking application is detection of moving object in complex scenes and these scene are difficult to analyze because of camera noise and lighting condition. Background subtraction is used for it. Other approach is Block matching. It consist of three components Block determination, Search method and matching criteria .It can be used in several videos where difficult traffic and weather are there.

Olivares et al [3] Alternatives In FPGA Block Matching Motion estimation takes a great part of processing time for video encoding .Best motion vector is obtained by full search algorithm. FPGA based design are used because it support high number of process elements in parallel mode. FPGA implementation of FSBMA for motion estimation video coding is analyzed.

Zhu et al [5] On the basis of study of motion vector distribution from commonly used test image sequences a new diamond search algorithm for fast block matching is proposed. Simulation results shows that proposed DS algorithm greatly perform than well known TSS three step .Experiment shows that it is better than recently proposed 4SS and block based gradient descent search, in terms of mean square error and search points. In this window size is not restricted.DS is implemented in MPEG video coding environment. randomization to rotate the cluster heads and achieves a factor of 8 improvement compared to the direct approach, before the first node dies. Further improvements can be obtained if each node communicates only with close neighbors, and only one designated node sends the combined data to the BS in each round.

Hunag et.al [6] Block matching motion estimation is heart of video coding system. Main concept of fast algorithm can be classified into categories like Reduction search points ,Simplification of matching criterion and many more. Main idea is quick checking of entire search range with simplified criterion to globally eliminate impossible candidates. Motion estimation engine is usually most important module in typical video encoder. This is used in VLSI architecture.

Pandian et al[7] presented that block matching motion estimation is essence of video coding system .In this they have studied different block matching algorithm ,for video compression. ME consumes 80% of computational power of if full search is applied.

Barjatya et.al [8] presented block matching algorithms used for motion estimation in video compression..

Gyaourova et al[9] presented Block matching is a standard technique for encoding motion in video compression algorithms. Goal of this (1) BMA is explored on low resolution and low frame rate (2) Improve the motion detection performance by the use of different search points ,during block matching. Block matching proved to be reasonably successful technique for object tracking.

Ahmed et al [13] presented a new technique called edge detection for fast block-matching motion estimation. For matching the macro block feature like shape and edge is used. Shade macro block has probability to move in the same direction as its neighboring macro block. So this property is used to reduce the number of average search points.

III. IMPEMENTATION

The basic idea behind object tracking is the videos being consecutive image frames changing rapidly. The scheme here is applied on the frames, the technique to be used is block matching algorithm that follows the under given scheme.

Step 1: Divide current frame to small rectangular blocks

Step 2: Motion of each block is assumed to be uniform

Step 3: Find the best match for each block in previous frame

Step 4: Calculate motion vector (MV) between current block and its counterpart in previous frame under the condition step 4 where intensities of pixels in current and previous frames are equal (MAE =0). (Including this condition prevents many search position whereby improving the algorithm.)

- Typical size for blocks: 16x16 pixels
- Maximum movement: w: typically 8, 16 or 32
- Matching Criteria:
- Mean Absolute Error (MAE)
- Mean Square Error (MSE)
- Sum of the Squared Error (SSE)
- MAE is preferred due to its simplicity
- Search Window (in previous frame)
- Rectangle with the same coordinates as current block in current frame, extended by w pixels in each directions
- Full Search

  All candidates within search widow are examined.

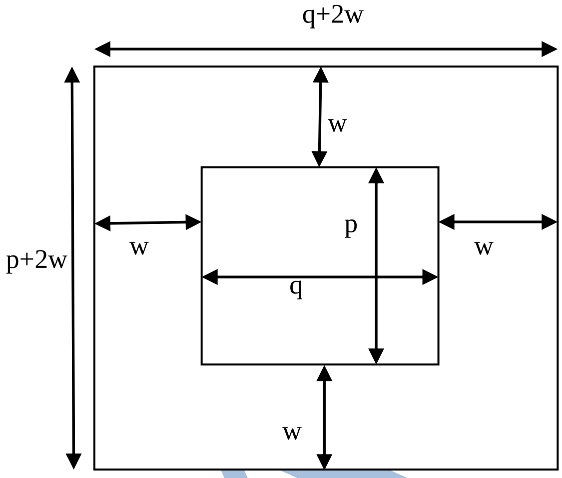

Fig 3 Search Window

- $(2w+1)^2$ positions should be examined
- Advantage: Good accuracy, Finds best match
- Disadvantage: Large amount of computation: $(2w+1)^2$ matches, 16x16 MAE for each match.

IV. RESULTS

Two videos are taken, one in broad daylight and the other one at night. Frames are captured in order of numbering 70,100,150 250 and 350 which is arbitrary in nature .While motion detection for a video, at initial step of algorithm false detections are prominent. In order to optimize this number, each frame is thresholded (cut-off) at threshold which depends upon the set of camera and optical conditions in scene. These videos are thresholded at 4500, 5000, 6000 respectively.

5.1 Parameters chosen for objective evaluation

a).Number of Iterations

b).Execution time per frame

c).Execution time for full video

5.2 Day Light Video at Threshold 5000

Table 5.2 for Day Light Video at Threshold 5000

FULL STANDARD BLOCK MATCHING (FSBM)					MODIFIED BLOCK MATCHING (MBM)						
SN	Frame Num	Green count	Clock Time in sec	Total processing time in sec	Green count	Red Count	Reduction in %age	Clock Time in sec	Total processing time in sec	Reduced Time in sec	Average Reduced Time For Each Frame in sec
1	70	3241	0.2867		3241	873	73%	0.2754		0.013	
2	100	4889	0.2857		4889	1287	79%	0.2780		0.007	
3	150	2858	0.4684	154.333	2858	880	69%	0.2848	118.606	0.1836	0.04712
4	250	1661	0.2782		1661	684	58%	0.2648		0.0134	
5	350	2601	0.2955		2601	925	64%	0.2759		0.0196	

Processing time for full video1 using FSBM= 154.333 sec

Processing time for full video1 using MBM= 118.606 sec

Reduced Execution Time =154.333 - 118.606 = 35.727 sec

5.2.1 Comparison of Iterations in FSBM and MBM for Day Light Video at Threshold 5000

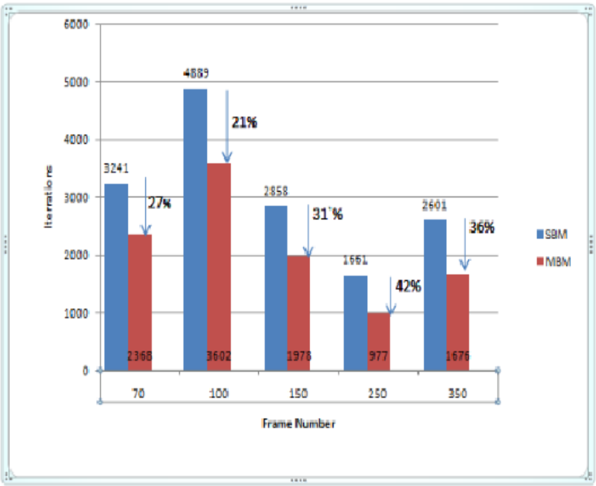

Fig 5.4 Comparison of Iterations in FSBM and MBM for Day Light Video at Threshold 5000

5.3 Night Video at Threshold 5000

Table 5.5 for Night Video at Threshold 5000

FULL STANDARD BLOCK MATCHING (FSBM)					MODIFIED BLOCK MATCHING (MBM)						
SN	Frame Num	Green count	Clock Time in sec	Total processing time in sec	Green count	Red Count	Reduction in %age	Clock Time in sec	Total processing time in sec	Reduced Time in sec	Average Reduced Time For Each Frame in sec
1	70	7303	0.3913		7303	1059	85%	0.3706		0.0207	
2	100	7293	0.4331		7293	1139	84%	0.3631		0.07	
3	150	7439	0.3829	852.4357	7439	1139	84%	0.3568	809.5088	0.0261	0.0760
4	250	7713	0.3920		7713	1102	85%	0.3647		0.0273	
5	350	8086	0.3920		8086	1237	84%	0.3684		0.0236	

Processing time for full video1 using FSBM= 852.43573sec

Processing time for full video1 using MBM= 809.5088 sec

Reduced execution Time = 852.4357- 809.5088 = 42.9269 sec

5.3.1 Comparison of Iterations in FSBM and MBM for Night Video at Threshold 5000

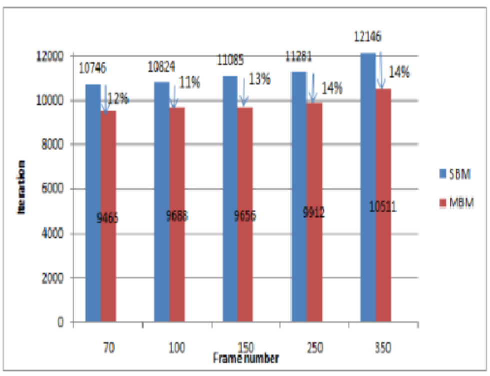

Fig 5.8 Comparison of Iterations in FSBM and MBM for Night Video at Threshold 4500

IV. CONCLUSIONS

In both the videos as threshold is increased, the percentage reduction is also increased and is almost linear. This indicates that for a given set of camera and visual scene. If number of motion detection is more, the threshold needed will also be on higher side whereby more saving in number of iteration it usually happens for the scene where intensity variation is more likely. Reduction in execution time at optimal value of threshold is ranging from 20 -40 sec for full length video in both cases. The new approach leads to reduction in execution time per frame but nearly about 0.02 sec for resolution of 320x120.

REFERENCES

[1] Nicole S. Love and Chandrika Kamath ," An Empirical Study of Block Matching Techniques for the Detection of Moving Objects" , Center for Applied Scientific Computing Lawrence Livermore National Laboratory 2005.

[2] Y.W. Huang, B.Y. Hsieh, T.C.Wang, S.Y. Chien, S.Y.Ma, C.F. Shen, and L.G. Chen, "Analysis and Reduction of Reference Frames for Motion Estimation in MPEG-4 AVC/JVT/H.264," in *Proc. of IEEE Int.* Conf. Acoust., Speech, and Signal Processing (ICASSP'03), 2003, pp. 145–148.

[3] Joaqu´ın Olivares, Ignacio Benavides, Javier Hormigo, Julio Villalba, Emilio Zapata "Fast Full-Search Block Matching algorithm Motion Estimation" University of C´ordoba Spain and University of Malaga Spain2006.

[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] S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," in Proc. of IEEE Int. Conf. Image Processing *(ICIP'97*), 1997, pp. 292–296

[6] S. Immanuel Alex Pandian, Dr.G. Josemin Bala and Becky Alma George "A Study on Block Matching Algorithms for Motion estimation" Karunya University, Coimbatore,India 2011.

[7] Aroh Barjatya, Student Member" Block Matching Algorithms For Motion Estimation" 20004 IEEE.

[8] Aglika Gyaourova and Chandrika Kamath and Sen-ching Cheung "Block matching for object tracking" Department of Computer Science, University of Nevada, Reno and Center for Applied Computing Lawrence Livermore National Laboratory 2003.

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

[10] Z. Ahmed, et al., "Mean Predictive Block Matching (MPBM) for fastlock-matching motion estimation," in Visual Information Processing (EUVIP), 2011 3rd European Workshop on, pp. 67-72.

[11] Zaynab Ahmed , Abir Jaafar Hussain and Dhiya Al-umeily "Edge Detection for Fast Block-Matching Motion Estimation to Enhance Mean Predictive Block Matching Algorithm" .2012 IEEE.

[12] Luigi Di Stefano, Enrico Viarani "Vehicle Detection and Tracking Using the Block Matching Algorithm" Department of Electronics, Computer Science and Systems (DEIS) University of Bologna 2000.
[13] Ismail Oner Sebe "Object-Tracking Using Multiple Constraints" DIGITAL VIDEO PROCESSING, WINTER 2002 Stanford University, EE Department E-mail: iosebe@stanford.edu 2002.
[14] Arnold W. M. Smeulders, Dung M. Chu, Simone Calderara and Afshin Dehghan "Visual Tracking: an Experimental Survey" student members IEEE 2013.
[15] Alper Yilmaz Ohio State University, Omar Javed ObjectVideo, Inc. and Mubarak Shah "Object Tracking: A Survey" University of Central Florida