MOTION IN IMAGES, BLOCK MATCHING ALGORITHMS FOR OBJECTS SEARCHING

Ing. Oscar Ivan Higuera Martinez, Ing. Leila Natalia Diaz Salcedo

Grupo de Investigación en Procesamiento de Señales DSP-UPTC,
Universidad Pedagógica y Tecnológica de Colombia, Sogamoso, Boyacá, Colombia.
Tel-Fax: (+57) 8-7706896 Ext. 214,
E-mail: {ohiguera, lenadisa}@gmail.com

Abstract: The present work is framed in the titled project of Investigation: "Design of an algorithm for the determination of apparent motion due to the displacement of the camera code DIN SGI 359". This text describes some algorithms of search to determine the correspondence among blocks in motion estimate on images sequences. This text gives an initial reference mark, where the used algorithms are described, as well as the realized measurement, and finally the results obtained in the implementation are discussed.

Keywords: Block Matching, Motion Estimate, Search Algorithm, Image Processing.

1. INTRODUCTION

Increases of processing speed and computation memory capacity have caused automated systems expansion for applications of everyday life, computer vision systems or artificial vision is a good example. These systems point to process visual information that in general is characterized by their great size and complex structure.

Artificial vision systems have been applying in surveillance systems, medical diagnostic, robots' navigation, systems of confirmation of identity, structure recovery starting from images sequences, compression and reconstruction of images sequences, tracking and dynamic characterization of objects in motion, among others.

Two or more images in the same scene constitute a sequence. Due to objects in a sequence experience moderate motions, a great likeness will exist among successive images (except in case of plane camera changes). This principle is used to estimate displacement, that is if motion increases, temporary correlation among pixels in the same space position decreases.

Objects' tracking is an application of artificial vision that looks for discovering an object position in a video or images sequence, where motion vectors are used to predict changes in the scene, and to reduce files and images sizes by coding only the current frame.

The paper includes a study of some of the main BM algorithms applied to objects tracking. In this case, the camera was immobile, and for further efficiency in processing time, sequences were used in grayscale.

2. MOTION ESTIMATION

There are diverse methods to estimate the displacement of objects in an images sequence, as optical flow based ones, frecuential methods, and block matching, among others. This article will show only block matching (BM) methods.

BM methods are based on dividing image in small not overlapped blocks of MXN pixels (typically 16X16 or 8X8) denominated sub-images and to estimate each one of these blocks displacement, assuming that all pixels in the same block experiences the same displacement. This supposition will be as correct as smaller sub-images are considered, which involves calculating a bigger sub-images number displacement. With these methods, there is a big error if displacement is not constant in a pixels block, which happens when image contains multiple objects in motion, objects which suffer a deformation, or other kind of motions like zoom, rotations, etc.

The displacement that a sub-image experiences; from a frame to the following one can be represented by a vector that will match displacement direction and magnitude. It is obtained looking for the block maximum correlation position inside the following image, unless sub-pixels refinement is used; in theory, motion vectors are obtained with a precision of ± 0.5 pixels.

2.1 Search region

$$Z = (M + 2dm)(N + 2dm) \qquad (1)$$

To find the displacement of a block S of MXN pixels between an image and the following one; it is necessary to define a maximum displacement (dm) [1], that defines a search area size (1). To carry out BM consists on displacing the sub-image S of the previous image for the whole search area in current image and then choose maximum correlation position between both [Figure 1].

This search involves an excessive number of operations that increase calculation time, which can be reduced using an intelligent search. BM algorithms entail a reduction for operations regarding to total search that would be $(2dm+1)^2$ numbers for evaluate.

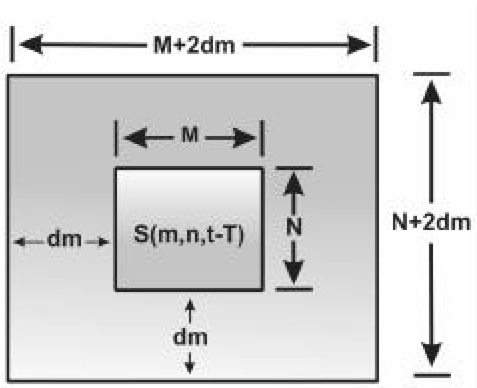

Fig. 1: Search Region, around the block S(m,n, t-T)

3. BLOCK MATCHING ALGORITHM

In an ideal case, two blocks are corresponding if they have the exactly same pixels. This is rarely true, because objects change in relation to the observer's point of view, reflected light on object surface also changes, and finally in real world there is always noise in images. Usually in scenes that contain motion there are occlusions, that is to say, new objects disappear and new ones appear. Besides images present false motion, which appears in situations where objects under interest have an uniform color. Blocks inside objects do not experience motion because all the pixels around them have a smooth color. When smooth regions are small, there is a bigger possibility of discover their motion because there is a bigger overlapping likelihood with non uniform color region, a larger block size can be used to overcome this problem.

Because of this, choosing right block size is not a trivial task, biggest blocks are less sensitive to noise, while smaller blocks produce good contours. In deep the main factor to choose block size; is the size of the object that needs to being search. The next factors are the amount of noise in the video frames, object’s texture and background.

Unfortunately, computational load grows quickly with the increase of the search area. In its totality, there are a big number of BM search window algorithms to determine motion. The implemented algorithms are Three Step Search (TSS), Two Dimensional Logarithmic Search (TDL), Four Step Search (FSS), and Orthogonal Search Algorithm (OSA), although there is great variety, it is observed these are characteristic and carrying out variations on them would give new methods.

In spite of the problems before mentioned, the calculation is fast and is apply to find likeness among regions [2]. Some of the matching approaches that are frequently used are based on the pixels difference; among them we have mean absolute distance (MAD), mean square distance (MSD), normalized cross-correlation (NCC), etc.

One of the BM algorithms problems is that they can converge to a local minimum instead of converge to the global minimum, for example the rate of reduction of search area can be done in function of the two smaller correlations of previous step, instead of the smallest correlation. Such algorithms are name search window dynamic algorithms.

There is another kind of algorithms; this one tries to exploit the natural space dependence (homogeneity) that exists in most of the images, therefore; surrounding blocks' motion vectors can be using to predict a block's motion vector. You can also exploit the temporary dependence, trying to predict the motion vectors for the current block; based on its motion vectors in the previous frame.

3.1 Three Step Search (TSS)

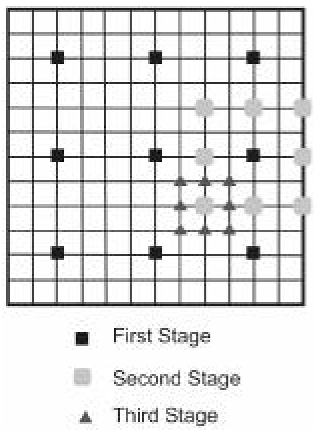

Fig. 2: Selected Block with TSS Search.

Koga *et. al.* [1-5] introduced this algorithm in 1981, this is maybe the most popular method for simplicity and robustness, its looks for the best motion vector with a fine search pattern, and the following three steps describe the algorithm.

Step 1: A size of the initial step is chosen (2^n), later 8 blocks at the step distance are chosen and to matching is performed, together with the center.

Step 2: The step size is divided by 2, and the new center is the block that registered the biggest correlation from the previous step.

Step 3: To repeat the steps 1 and 2 until get a size step equal to 1, the point with the biggest correlation will be the result.

In this algorithm, it is observed that the repeating number of steps 1 and 2 is equal to n, and the maximum displacement that would be achieved is 2^n-1. An example is shown in the figure 2, n=3.

3.2 Two Dimensional Logarithmic Search (TDL)

Introduced by Jain and Jain at the same time that the TSS algorithm, and shows certain relation with this one [5-6], TDL algorithm requires a bigger number of steps that TSS algorithm, especially when the search window is bigger. Following steps describe the TDL algorithm.

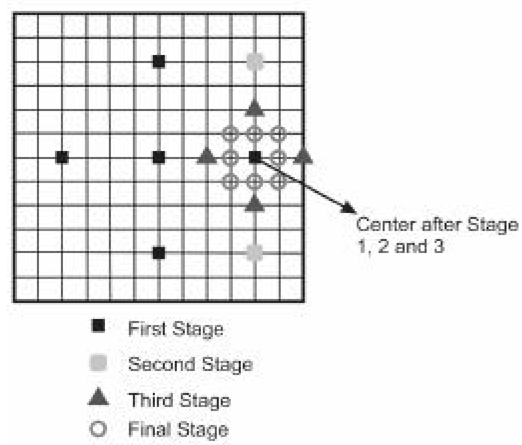

Fig. 3: Selected Block with TDL Search.

Step 1: An initial step is chosen, four blocks are chosen around the center, but in this case, they are shaping a cross, and matching is performed in this five blocks.

Step 2: If the position with smaller error is in the center, the step size is divided in two, and then go back to the firs step. However, if the position with smaller error is on one of the other points, the center is moved to that position and goes to the firs step

Step 3: When the step size is equal to 1, the search is carry out on all the nine points around the center and the best correlation determines the required block.

The algorithm is observed in the figure 3. There are many variations of this algorithm, and there are different ways of changing the step size, some authors suggest that this must be changed in each step, maybe the bigger inconvenience is that the algorithm may not converging if the initial step is very small or the motion is very big.

3.3 Four Step Search (FSS)

This algorithm was proposed by Lai-Man Po and Wing-Chung MA in 1996 [5], [7]. It is based on real images sequences characteristics. The following steps describe the FSS algorithm:

Step 1: The initial step is equal to 2, then choose nine point around it (figure 4a), the matching performance is calculated, if the point with the bigger correlation is in the center go to the step 4, otherwise go to the step 2.

Step 2: The center is moved to the position with the biggest correlation, the step size stays equal to 2, but in this case the search parameter depends on the place with the biggest correlation in the previous step, the taken points are shown in the figure 4b and 4c, and matching is performed. If it is in the center, go to the step 4 in another case go to the step 3.

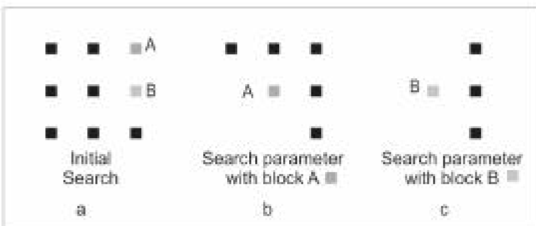

Fig. 4: Search Parameters (FSS).

Step 3: To repeat the search strategy until be able to go to the step 4.

Step 4: The step size is reduced to be equal to 1, the search is carry out on all the nine points around the center and the best correlation determines the required block. Figure 5 shows a FSS algorithm example.

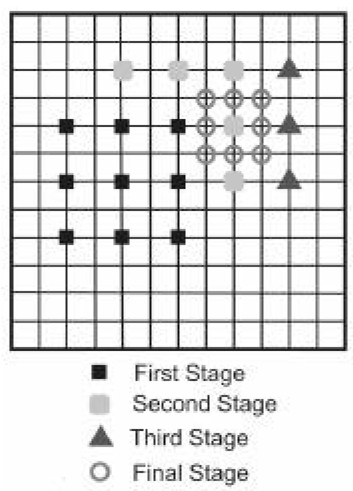

Fig. 5: Selected Block with FSS Search.

This algorithm is less complex than TSS, its evaluation in terms of performance is good, it is more robust than TSS, and maintains its operation for image sequences with complex motions like zoom and fast motions; therefore, it is a very attractive strategy for the estimation of the motion.

3.4 Orthogonal Search Algorithm (OSA)

Introduced by Puri (1987), this algorithm is a TSS and TDL hybrid [5], it is based on a horizontal and vertical search to find the match. The following steps describe the OSA algorithm:

Step 1: A size step is chosen (usually half of the maximum displacement in the search window). To take two points horizontally at that distance of the window center and to obtain matching performance in the three points, and the center is moved to the position of the match.

Step 2: To take vertically two points at the step distance of the new center and calculated the maximum correlation, this it will be the new center.

Step 3: The step size is divided in a half and if this it is smaller than 1 end the process. Otherwise carried out steps 1 and 2 again. Figure 6 shows an example.

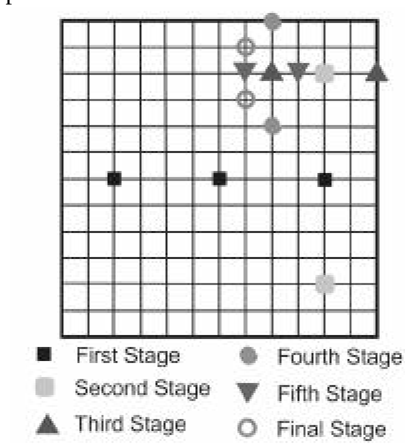

Fig. 6: Selected Block with OSA Search.

4. RESULTS

To measure algorithms performance were used synthetic sequences with known motion, and the results were confirmed with real image sequences. To measure it was used correlation between two blocks; using the mean absolute distance (MAD), as is observe in the equation 2 the MAD is a measuring of difference between blocks A and B [4], consequently a lower MAD means a bigger correlation between blocks A and B.

$$MAD = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|A(i,j) - B(i,j)\right| \quad (2)$$

To consider the motion existence was defined a threshold for minimum difference between two blocks, which name as the zero motion. With this threshold the correlation is: the MAD is calculated between the current block and the corresponding block (without motion), if the MAD is below the threshold, is considered that there is not motion and any other block is not evaluated, if it is not this way, the search must be carry out with the algorithms before mentioned. This technique minimizes to the maximum the unnecessary search of false motion, this helps to eliminate the effect caused by the noise, and it reduces the time of final processing.

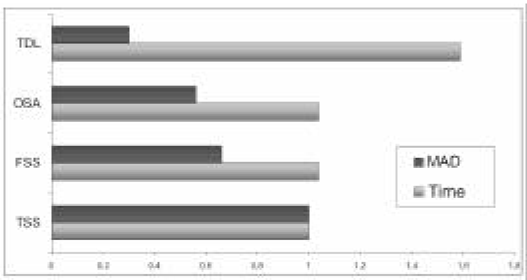

Fig. 7: Results.

The results for different sequences used to carry measures out; were normalized to produced ones by the TSS algorithm in order to comparing it (Figure 7). The most efficient methods with regard to MAD are TSS, OSA, and FSS. As for time TDL presents the smallest time and TSS presents the high one. With these results it is recommended to use OSA and FSS methods.

5. CONCLUSIONS

Although implemented algorithms are not optimums, they show an important processing time reduction. This leads us to continue testing similar algorithms, to be able to reduce MAD and searching time. For example to test intelligent search strategies, among these, a search based on statistics of previously calculated motions.

To sacrifice in excess the MAD to reduce searching time can produce big errors that are not useful for some applications (like in Medicine), because they can be derived in mistaken interpretations.

REFERENCES

[1] Faundez M. *Tratamiento digital de voz e imagen y aplicación a la multimedia*. México DF: Alfaomega Grupo Editor, 2001.

[2] Koga, T. [et al]. Motion-compensated interframe coding for video conferencing. Proceedings NTC'81 (IEEE), G5.3.1-5, New Orleans, LA. 1981

[3] Po, L-M. y Ma, W-C. A Novel Four Step Search Algorithm For Fast Block Motion Estimation, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no.3, pp 313-7,June 1996.

[4] Gyaourova, A.; Kamath, C. y Cheung, S.-C. Block Matching for Object Tracking, Department of Computer Science, University of Nevada, Lawrence Livermore National Laboratory. October 14, 2003, URL-TR-200271, Consult February 2006. Available on: http://www.llnl.gov/CASC/sapphire/pubs/UCRL-TR-200271.pdf.

[5] Turaga, D. y Alkanhal, M. Search Algorithms for Block-Matching in Motion Estimation, Mid-Term project, 18-899, Spring, 1998, Consult February 2006. Available on: http://www.ece.cmu.edu/~ee899/project/deepak_mid.htm

[6] Jain, J. R. y Jain, A. K. Displacement measurement and its application in interframe image coding, IEEE Trans. Commun., vol. COM-29, Dec. P. 1799-1808. 1981

[7] Zahariadis, T. y Kalivas, D. A Spiral Search Algorithm for Fast Estimation of Block Motion Vectors, Signal Processing VIII, theories and applications. Proceedings of the EUSIPCO 96. Eighth European Signal Processing Conference.