OPTICAL FLOW DETECTION USING AN IMPROVED BLOCK MATCHING ALGORITHM

Remus BRAD

"Lucian Blaga" University of Sibiu
Computer Science Department
rbrad@ulbsibiu.ro

Abstract:
In the last few years, researchers shown a growing interest for computer detection and tracking of moving objects. The Block Matching Algorithm (BMA) approach is employed in the MPEG standard, as well as in a large variety of optical flow detection techniques. The BMA divides the current frame in a number of blocks and search for a matching in the next frame, in order to estimate the displacement of blocks between two successive frames. Due to large searching space, the classical BMA is a greedy algorithm and a number of versions where developed toward reducing the computational load. We present a new BMA by making a number of notes on the classical BMA implementation. We have improved the computation time by reducing the number of operations and loops required. We have also found an intrinsic optical flow regularization method and therefore obtain smoothed motion vectors.

Key words: computer vision, motion detection, optical flow, block matching algorithms

1. Introduction

Usually, in computer science, the motion term is associated with computer vision. In this special case, motion is restricted to a 2-dimensional projection of the moving objects 3-dimensional paths. Thus, the set of motion vectors obtained is named optical flow. In the last few years, a growing interest in motion detection and analysis has created a large number of applications, starting from video surveillance and ending with image compression.

All developed methods are based on a local estimation of linear motion and are assuming the preservation of local distribution of image intensity. For a given point, the motion vector can be recovered computing the displacement of the best matching of two regions in successive images. Matching is estimated using a particular measure. These special cases of methods are named Block Matching Algorithms (BMA).

2. Block Matching Algorithms

BMAs are the most common motion estimation methods and are widely adopted by image compression and codification standards (MPEG, CCIT H261/262/263). BMA are assuming the fact that a block of pixels has the same translation motion from one frame to another [1]. This is usually applied to a rectangular region of a fixed size. One can define a searching region for the motion vector, such as:

$$|u| \le \Delta_h \text{ and } |v| \le \Delta_v \tag{1}$$

If the block size is K lines and L columns, the searching space is $(K+2\Delta_h)(L+2\Delta_v)$ and the number of possible directions is $(2\Delta_h+1)(2\Delta_v+1)$. Usually, the matching criterion to be optimized is the absolute mean difference of displaced frames:

$$(\hat{u},\hat{v}) = \underset{\substack{(u,v)\in Z^2 \\ |u|\le\Delta_h, |v|\le\Delta_v}}{\arg\min} \sum_{i=0}^{K-1}\sum_{j=0}^{L-1} \left| I_{DFD}(i,j;u,v) \right| \tag{2}$$

where the frame difference is:

$$I_{DFD}(i,j;u,v) = I(i,j;k) - I(i-u,j-v;k-1) \,.$$

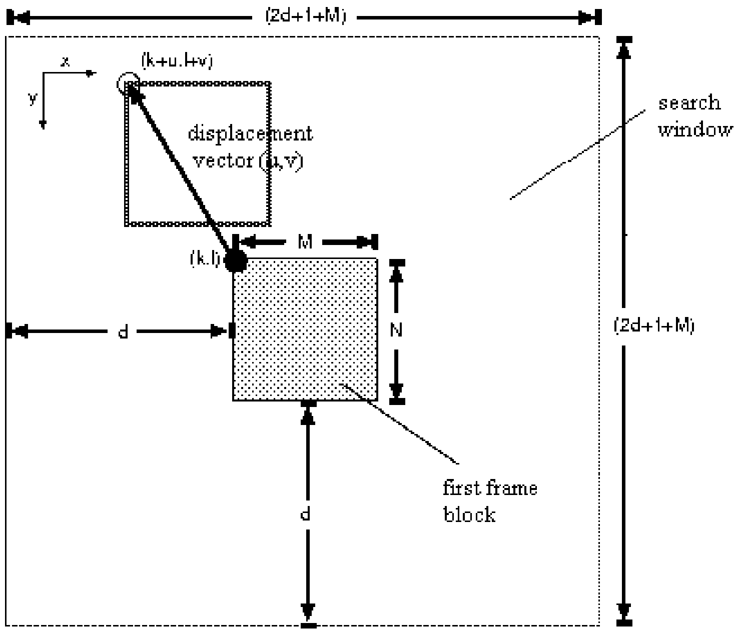


Figure1. The block matching model

Figure 1 describes the principles of BMA. The current window is divided in several rectangular blocks. For each block in the current frame, a displacement vector is obtained as coordinates of the best matching block in the search window of the next frame. The Full Search (FS) algorithm is the most common BMA. It requires $(2d+1)^2$ operations and it finds the optimal motion vector among all candidates. Thus, computing complexity makes it useless for real time applications. The aim of reducing the search complexity is achieved by several FS derivations such as: Three Steps Search, 2D Logarithmic Search [2], Cross Search, Orthogonal Search, Four Steps Search [3], Gradient Descend Search.

3. A new search algorithm

As previously mentioned BMA methods are searching for the global minimum of a given function. Thus, comparisons between blocks of two successive frames can exceed 10^5 and a number of speed-up methods have been proposed. After completing a study on most of them, we have observed the possibility of improving the convergence, by a series of additional tests as following.

3.1. Comments on search space reduction

3.1.1. First comment

Equation 2 represents the search criterion to be optimized (absolute difference between two regions of successive frames) and may be expressed as:

$$D(i,j)=\sum_{k=0}^{P-1}\sum_{l=0}^{Q-1}|I(k,l,t)-I(k+i,l+j,t-1)| \quad (3)$$

One can observe that all terms are positive and if at a certain moment of the search process the partial evaluated sum exceeds the previous computed minimum, then (i,j) is not the desired displacement and further estimation of the sum is pointless. Efficiency can be achieved by comparing the computed sum and the minimum for every row of tested block. In the case of FS, this strategy can decrease to half the time required to compute the minimum of the search criterion (2).

3.1.2. Second comment

In [4], similar observations are leading to a search reduction method. If equation 3 is expressed as an inequality, then

$$\left|\sum_{k,l}|I(k,l,t)|-\sum_{k,l}|I(k+i,j+l,t-1)|\right|\le \sum_{k,l}|I(k,l,t)-I(k+i,l+j,t-1)| \quad (4)$$

In FS algorithm, the absolute difference is computed for every (i,j) in the search window. At every increment, comparison between the computed value and the already estimated minimum is performed in the goal of convergence. Computing individually the sums in the left part of the inequality and supposing that in the current search window a minimum is found in (m,n), then (i,j) will be the next considered point. If

$$\left|\sum_{k,l}|I(k,l,t)|-\sum_{k,l}|I(k+i,j+l,t-1)|\right|> \sum_{k,l}|I(k,l,t)-I(k+m,l+n,t-1)| \quad (5)$$

then (i,j) will not lead to a successful minimum and the right part of (4) is not longer required to be computed. In opposite, for a small number of cases, (4) must be computed, but the left term sums are obtained incrementally. This observation reduced the search time to approximately. 1/3.

3.1.2. The third comment

If the pixels belonging to the current block didn't suffer any spatial transformation, then searching for a minimum of (2) is a waste of time. Therefore, at the beginning of the process, we will first compare the blocks at the central position of the two frames. A threshold must be used, due to the presence of noise and must be set according to its quantity.

3.2. Optical flow regularization

BMA are producing an irregular optical flow, especially due to pixels of moving objects gray level variation. Incoherence of optical flow may cause difficulty to subsequent processing stages, such as compression, segmentation or motion compensation. Di Stefano [5] is proposing a median filtering regularization method .

Our method is based on the observation that neighboring pixels usually have similar motion vectors. Optical flow smoothing can be achieved in a post-processing stage, as in [5] or during the flow determination, as we are suggesting in our method. Consequently, in the block-matching step, we have increased the size of the reference block, presented in figure 1, from M to $3M/2$. Even if the image is divided in M sized block, when computing the displacement, we will consider a larger area. Therefore, a smoothing effect will be observed. In figure 2, we presents a test sequence ("Hamburg Taxi") and the optical flow results without c) and with the smoothing d) effect due to our observation.

We have changed the FS algorithm according with the four previously presented observations and named it New Full Search (NFS). Even if the computing time and the total number of operations are larger than other methods [4], the proposed algorithm is based on a global search and doesn't fall in local minimum.

4. Experimental results

The test sequences employed in our experiments comprises the "Salesman" in 360x288, "Hamburg Taxi" in 256x190, "Yosemite" in 316x252 and "Translating Tree" in 150x150. In all cases presented in the paper, we have used an M=16 reference block size and a distance d=7 for the search window. The "Hamburg Taxi" contains 4 moving objects: the two cars entering from the left and right side of the image, the taxi in center and a pedestrian in upper left corner. Figure 2 presents the first and the third image from the sequence and the optical flow without and with smoothing using the NFS algorithm.

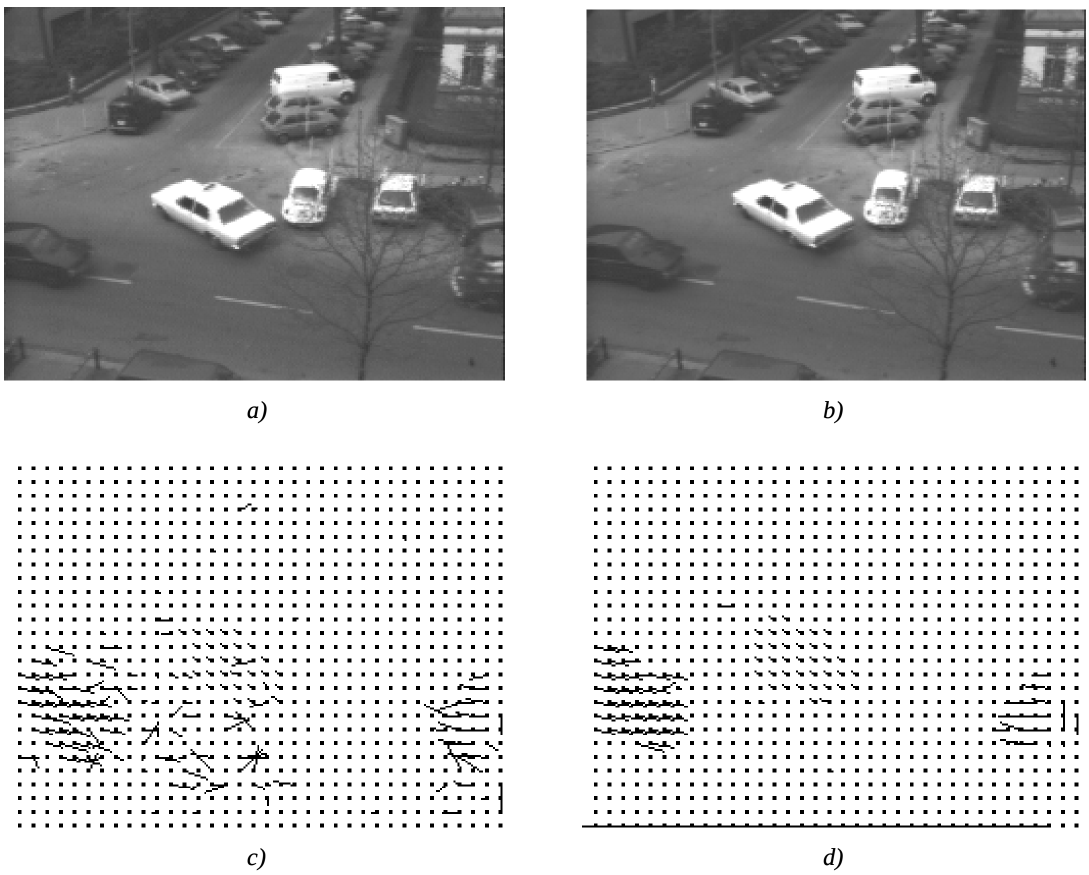
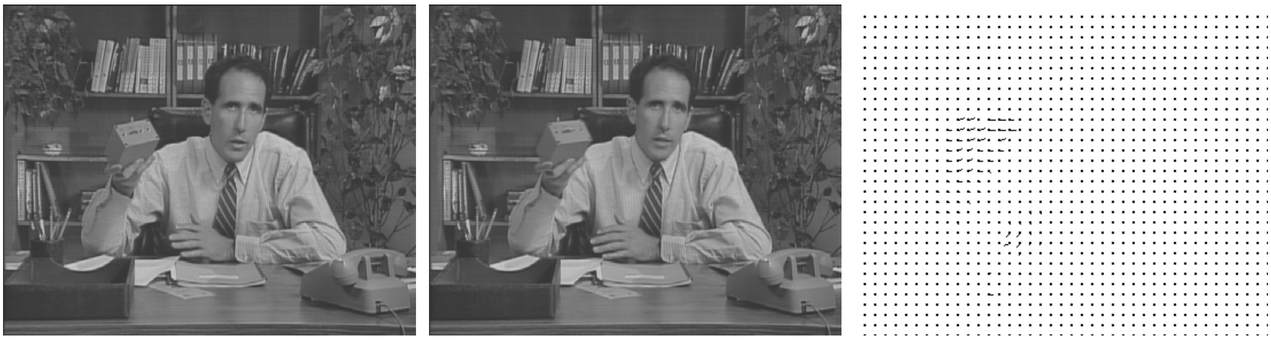


Figure 2. "Hamburg Taxi" sequence. a) first and b) third image; optical flow c) without and d) with smoothing.

For the "Salesman" sequence, we have employed the second and fourth image. The sequence contains two types of motions: one of the left arm, in the right direction and one of the right arm's fingers. Figure 3 presents the optical flow computed by the NFS method. Figure 4 and 5 shows results for the "Translating tree" and "Yosemite" test sequences.

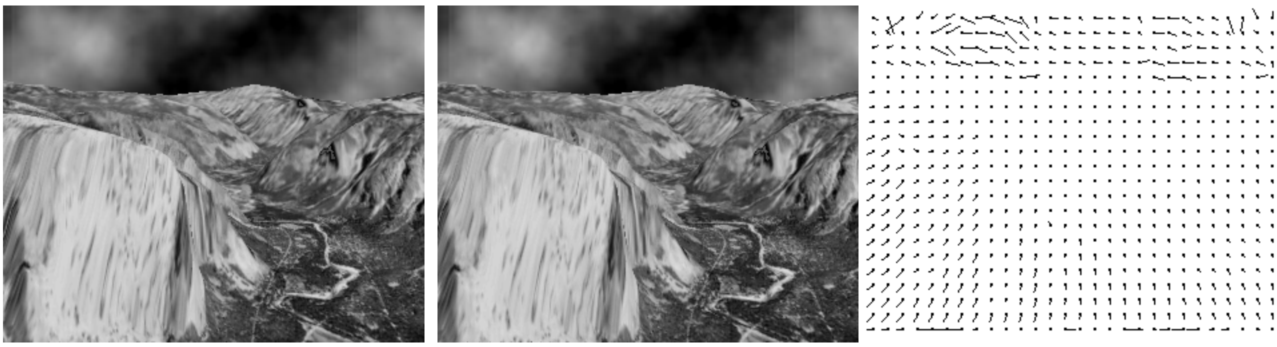


a) b) c)

Figure 3. "Salesman" sequence. a) second and b) fourth image; c) optical flow

a) b) c)

Figure 4. "Translating tree" sequence. a) 14-th and b) 15-th image; c) optical flow



a) b) c)

Figure 5. "Yosemite" sequence. a) 5-th and b) 6-th image; c) optical flow

5. Conclusions

We have accomplished a comparative study of the block matching algorithms and based on the comments we have made, we have found possibility to improve performances. The NFS algorithm proved good results in optical flow detection and in computing complexity. Future work will include a labeling algorithm for moving objects and an application in meteorological sequence processing.

References

[1] C.K. Cheung, L.M. Po - "*Hybrid Search Algorithm for Block Motion Estimation*", Proc. IEEE Int.Symp.Circ.and Syst., vol.IV, pp.297-300, May 1998

[2] C.H. Lee and L.H. Chen - "*A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid*", IEEE Transactions on Image Proc., vol.6, no.11, pp.1587-1591, November 1997

[3] L.M. Po and W.C. Ma – "*A Novel Four-Step Seach Algorithm for Fast Block Motion Estimation*", IEEE Transactions on Circuit and Systems for Video Technology, vol.6, no.3, pp.313-317, June 1998

[4] R. Brad - "*Detecția mişcării din secvențe de imagini*", Referat de doctorat III, Universitatea Tehnica Cluj-Napoca, 2000

[5] L. Di Stefano and E. Viarani - "*Vehicle Detection and Tracking Using the Block Matching Algorithm*", Proc. of 3rd IMACS/IEEE, Athens, Greece, vol.1, pp. 4491-4496, July 1999