Martin Čapka *

THE BLOCK MATCHING ALGORITHM AND ITS USE FOR VEHICLE MOTION DESCRIPTION

This paper deals with principles of Block Matching Algorithm (BMA) and its use for vehicle detection and tracking. The Block matching plays essential role in motion estimation employed in the MPEG compression standard. The current frame is divided by BMA in smaller, fixed size blocks. These blocks are then matched in the previous frame in order to estimate their displacement between two successive frames. BMA supplies us with information referred to as motion vectors. Motion vectors are next processed to gain relevant data for vehicle detection and tracking.

1. Introduction

The present expansion of automated systems for everyday life applications is caused mostly by the growth of computer memory and processor speed. Computer vision area can be seen as a good example of such intensive computations systems, whose use becomes more feasible with the advances in computing power. Object tracking is a key computer vision topic aiming at detecting the position of a moving object from a video sequence. Motion detection and estimation is very important for video encoding. The Block matching principle, which is used in MPEG compression standard to estimate motion between images in sequence, provides so for the mentioned goal very interesting information – motion vectors associated with image points.

2. The Block Matching Algorithm

The BMA is based on partitioning each frame in a given sequence into square blocks of a predefined size (in pixel: 8×8 or 16×16 ...) and detecting their displacement between the *anchor* (actual) frame and the previous one called target frame as well. For each block of the target frame the searching inside a given scan area is performed. It provides a matrix of displacement vectors (DVM) associated with it. Each block encloses a part of the image and defines in the previous frame, a "*scan area*", centred in the block centre. The block is shifted pixel-by-pixel inside the scan area, calculating the match value at each shift position. This is shown in Fig.1.

The similarity between blocks can be evaluated on the basis of several matching measures. The mostly used criteria are the Normalised Cross- Correlation Function (NCCF) (1), the minimum of Mean Square Error (MSE) (2) or the minimum of Mean Absolute Difference (MAD):

$$NCCF = \frac{\sum_{i,j} [A(i,j) \times (B(i,j)]}{\sqrt{\left[\sum_{i,j} A^2(i,j)\right] \times \left[\sum_{i,j} B^2(i,j)\right]}} \qquad (1)$$

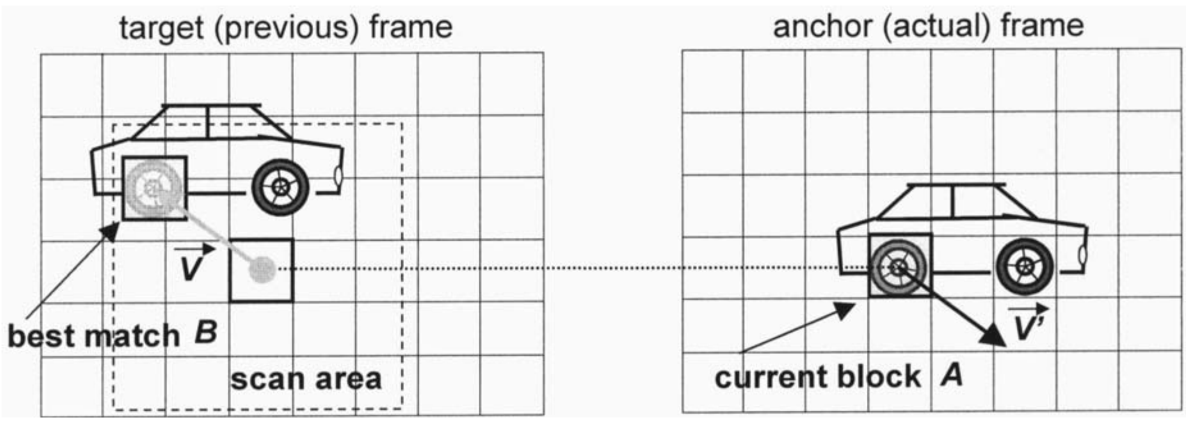

Fig. 1. The Block Matching Principle

* **Martin Čapka**
Department of Transportation Networks, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, SLOVAKIA, E-mail: martin.capka@fri.utc.sk

$$MSE = \frac{1}{N_1 N_2} \sum_{i,j} [A(i,j) - (B(i,j)]^2 \quad (2)$$

$$MAD = \frac{1}{N_1 N_2} \sum_{i,j} |A(i,j) - (B(i,j)| \quad (3)$$

where " $\times$ " represents the product between the corresponding element's value of matrix "A" and matrix "B". Indexes i, j roll all over the matrix size N1, N2. The size of the scan area has also impact on time consumption, mainly in case of exhaustive search. Therefore, the highest possible block displacement between two frames should be taken into account by setting the size of the scan area. Though there are many improvements [1] on searching strategies, it is convenient to keep it as small as possible.

The matching process produces a displacement vector V' between the position of the block in the anchor frame and its best match in the previous frame. In the actual frame, the reflected vector $V' = -V$, applied in the block centre represents the block displacement as well as its tracking information.

3. Noise elimination

Unfortunately, mostly due to noisy tone fluctuations, the BMA generates many wrong vectors over static blocks located on the background. A small modification to the conventional block matching algorithm can be made in order to lower the effect of this kind of noise – the threshold which defines the maximum difference between two corresponding blocks under which they are considered matching.

The threshold value is applied first on the computed matching criterion between the current block and the corresponding block without displacement in the target frame. If the block similarity does not exceed the threshold value, the current block is considered then as motionless and no more matches are evaluated. This technique results in a significant computational saving [3].

The unavoidable tone fluctuations of moving objects between successive frames cause that the BMA usually produces also a noisy DVMs. Noisy vectors may have a significant influence on segmentation of vehicles. As mentioned in [7], the noise found in the DVMs may be modelled as the presence of outliers in a rather regular vector field. A vector median filter is often used as a reliable tool to smooth these differences. The median value of a set of n scalars is defined as $(n/2)$th element of the ordered set. For the two-dimensional vectors the output of the vector median filter by [2] is obtained by taking the element of a set of vectors which minimises the sum of the distances from all the other elements, with distances evaluated on the basis of the *L2-norm.*, in the $R \times R$ space. The distance between two vectors, $u[a,b]$ and $v[c,d]$ is given by:

$$\| \vec{u} - \vec{v} \|_{L_2} = \sqrt{(a-c)^2 + (b-d)^2} \quad (4)$$

If we simply use the median filter on DVM, most of no null vector entries are changed into null-ones since the neighbourhood tends to be dominated by null vectors. Due to this inadmissible erosion of the connected components not only the splitting of several vehicles occurs, but also they may be recovered very poorly or even completely lost.

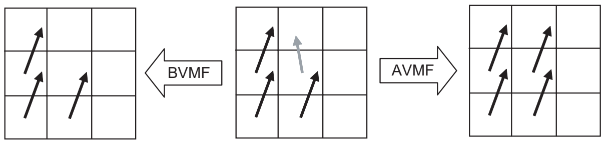

Fig. 2. The application of basic (left) and adaptive (right) vector median filter on DVM (middle)

Fig. 2 left shows an example how the vector in the centre of the "8- neighbourhood" is going to be deleted rather than made similar to its no null neighbours because it has more null neighbours than no null ones.

To overcome this, an "adaptive" filter has been devised in which only no null vectors are taken into account during the smoothing process (Fig. 2 right). That is, each no null vector is turned into the vector median computed on the basis of its no null 8-neighbours. This operator may be viewed as an adaptive statistical order (or rank) filter in which the order changes as a function of the local characteristics of the signal [6].

4. Detecting of moving objects

After the restored DVM are obtained, the next step is to detect vehicles by grouping together connected clusters of blocks having similar (and nonzero) displacement vectors. The grouping process is basically a sequential, iterative connected components labelling algorithm ([4]) in which the blocks having no null displacement play the role of "foreground points" and two blocks are considered neighbours if they are 8-connected and the *L2-norm* of their displacement vectors is less than a threshold *Th*:

$$\| \vec{u} - \vec{v} \|_{L_2} = \sqrt{(a-c)^2 + (b-d)^2} \leq \text{Th} \quad (5)$$

This process is sometimes referred to as *δ-labelling*. Moreover, in [7] it was decided to filter out the groups consisting of very few blocks in order to increase robustness with respect to noise. The value of the threshold on the group size depends on the size of the block with respect to the size of vehicles. The labelling is run on every frame and produces a matrix, called *label_map*, having the same size as the DVM. After obtaining the *label_map*, another matrix, named *resized_map*, is produced. It is obtained by enlarging the *label_map* to the size of the original frame so as to have a matrix with labelled objects at the same resolution as the image. This matrix will be used in the tracking algorithm.

5. Tracking of vehicles

The tracking algorithm relies on the actual frame's block matching and labelling outputs. The labelling output consists indeed of a set of temporary labels which will be updated by the tracking step according to the block-level tracking information provided by the BMA and embedded into the DVM. Given the set of the blocks belonging to a labelled object in the actual frame, we translate every block in the previous-frame's *resized_map* by a vector equal to its reflected DVM entry ($V' = -V$) and evaluate the amount of overlap of the temporary label assigned to the object with the labels in the previous frame.

The process is described in Fig.3 where the middle part shows the temporary labelling of an object in the actual frame and the left one the established labelling of objects in the previous frame. To track the object "4" we reflect the DVM and we shift back the blocks by the reflected vectors over the previous frame's *resized_map*. Each shifted block will overlap *NxN* pixels of the previous *resized_map* pixels: these pixels represent the label entries for the considered block which will be used to determine the final global label value. As shown in Fig. 3 left the block may overlap labelled pixels (grey-hatched) as well as un-labelled ones (white). Scanning all the pixels of the shifted label "4" we calculate the number of entries for each different pixel label. After repeating these operations for each block we finally obtain an array of label values, where each cell contains the number of the entries for that label value. The number of the cell that contains the maximum number of entries will be the new label for all the blocks of the actual object. With reference to Fig. 3 left, it is evident that the maximum number of entries will be found at the location number "5" of the mentioned array. Consequently, Fig. 3 right shows the result of tracking the temporary label "4", which has been turned into label "5".

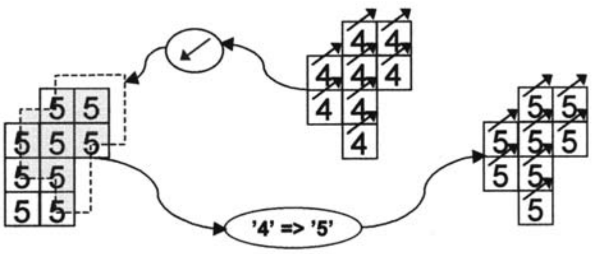

Fig. 3. Tracking of vehicle.

If for a certain temporary label we find no overlap with the *resized_map* of the previous frame, then a new unused label value will be given to this label (it is a "new born").

6. Conclusion

Moving objects evolve in the scene. Frame by frame we can recover their shape at the resolution of the BMA's blocks grid (for example, see Fig. 4). Objects motion, together with low-resolution shape recovery, noise and consequent filtering operations, may cause substantial changes in a shape and size of vehicles. For these reasons it may happen that the maximum number of overlapped pixels corresponds to the "unlabelled" label. The problem arises especially in case of objects made out of a few blocks. If we simply assigned the object the label corresponding to the maximum number of overlapped pixels we might label as a "new born" an object which conversely could be tracked. Hence, in this situation we look for the presence of labelled pixels and choose as the new label for our object the label value with the maximum number of pixel entries.

Fig. 4. Vehicle's shapes at the resolution of the BMA's blocks grid.

Experimental results in [7] and [3] show that the block matching principle with its enhancements as correlation threshold, adaptive filtering, grouping and tracking is quite effective in detecting and tracking vehicles. The worst problem with the BMA seems to be its computational load. For this reason, every preselection of "blocks of interest" may help to reduce the amount of data to be processed. Very useful pre-selection is provided by threshold differencing between the actual frame and background. As for a vehicle detection, there can be still some over-segmentation effect when vehicles are too big with respect to block size. The correction of perspective influence can in many cases bring also more accuracy on motion description.

Acknowledgment:
This work has been supported by the institutional grant FRI 06/2004.

References:

[1] ALKANHAL, M., TURAGA, D.: *Search Algorithms for Block-Matching in Motion Estimation,* Technical report on Mid-Term project, No.18-899, Carnegie Mellon University, Pittsburgh, Spring 1998.
[2] ASTOLA, J., HAAVISTO, P., NEUVO, Y.: *Vector Median Filters,* Proceedings of IEEE, vol. 78, No. 4, April, 1990.

[3] CHEUNG, S.-C., GYAOUROVA, A., KAMATH, C.: *Block matching for object tracking,* Technical report of University of Nevada, No. UCRL-TR-200271, October 2003.
[4] HARALICK, R. M., SHAPIRO, L. G.: *Computer and Robot Vision,* Reading MA: Addison-Wesley, 1992, ISBN 0-201-10877-1.
[5] HOANG, D., T., VITTER, J., S.: *Efficient Algorithms for MPEG Video Compression,* New York: John Wiley & Sons Inc., 2002, ISBN: 0471379425
[6] SHALKOFF, R. J.: D*igital Image Processing and Computer Vision,* New York: John Wiley & Sons Inc., 1989, ISBN: 0471857181.
[7] DI STEFANO, L., VIARANI, E.: *Vehicle Detection in Traffic Images,* In: Proceedings of ICEIS'99, 1st Int'l Conference on Enterprise Information Systems, Portugal: Set_bal, Vol. I, pp. 308–313, March, 1999.