EFFECTIVE MOTION FIELD DESCRIPTION BASED ON AFFINE MODELS AND GLOBAL MOTION INFORMATION

Marco Barbieri$^{(\dagger)}$, *Rosa Lancini*$^{(\ddagger)}$
$^{(\dagger)}$ Signal Processing Laboratory,
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
Tel: (+39 2) 661 612 40; Fax: (+39 2) 661 004 48

$^{(\ddagger)}$ CEFRIEL, Politecnico di Milano
Via Emanueli 15, I-20216 Milano
Tel: (+39 2) 661 612 09; Fax: (+39 2) 661 004 48
e-mail: rosa@mailer.cefriel.it

ABSTRACT

In this paper we study the possibility to estimate reliable motion field by considering both a global motion, due to camera parameters changes and a local motion, due to the displacement of the image objects. By considering two images I_n and I_{n-k} a first motion field is estimated using a block matching algorithm. Thanks to this information, the global motion parameters (horizontal/vertical pan and zoom factor) are estimated. One of the two images is then compensated by the estimated global motion. A combination of a block matching and a differential algorithm is used to obtain a dense local motion field. Simulation results indicate that the detection and compensation of the global motion are essential for good motion filed estimation and motion compensated prediction. Moreover the local motion field is used as input for a segmentation algorithm based on affine model, in order to detect the moving object present in the scene.

1 Motion estimation

Motion estimation and prediction are used in the state of the art moving image coders like MPEG-1/2 [1]. Generally, the block matching algorithms are able to estimate only translation displacements and therefore their performance fails when zoom and warping (Global motion) are present. To overcome this problem we apply a technique that is able to find both global motion (due to camera parameter changes) and local one (due to the displacement of the image objects).

1.1 Global motion (Pan/Zoom) Estimation

When we allow the change of camera parameters during sequence acquisition, global motion is present The motion of the image points depends not only on the spatial displacements of the imaged objects, but also on the camera parameters. In particular only the change of pan (along xy-plane) and zoom (along z-axis) are allowed in our camera model. The three parameters derived from the described model are: pan along x-axis p_x, pan along y-axis p_y and z as the zoom along z-axis. Considering a couple of images I_n and I_{n-k}, the global motion parameters are detected by starting from a first rough motion field obtained by a block matching algorithm [2]. In fact when a zooming occurs the motion field is generally very noisy, due to the fact that only translational motion field can be correctly estimated by block matching algorithm. The motion field obtained by a set of global motion parameters would be described by:

$$\Delta_{xi} = z x_i + p_x , \qquad \Delta_{yi} = z y_i + p_y . \tag{1}$$

To obtain the unknown global motion parameters, we minimize a certain error function, chosen to be:

$$E\left(p_x, p_y, z\right) = \sum_{i=1}^{N} s_i \left[\left(\Delta_{xi} - \alpha_i\right)^2 + \left(\Delta_{yi} - \beta_i\right)^2 \right] , \tag{2}$$

where p_x, p_y, z are respectively pan on xy-plane and zoom along z-axis, $\{s_i\}$ the array of selected displacement (initially all "1"), Δ_{xi} and Δ_{yi} are field distortions due to global unknown parameters (see (1)), (α_i, β_i) is the displacement estimated for each block with block matching algorithm; the sum is over the all $i = 1, \ldots, N$ image blocks. As suggested in [2], global parameter estimation is performed with two iterations. A first estimation is carried out using all the available motion vectors, then the displacements that don't match (over a given threshold) the recovered global motion field are discarded and the estimation procedure runs another time. In the discard procedure, the value s_i of $\{s_i\}$ array corresponding to the discarded (α_i, β_i) displacement is set to "0". The image I_{n-k} is compensated obtaining I^*_{n-k}, by using this first estimated global motion parameters. The compensated image I^*_{n-k} has the characteristic to be shot with the same global camera parameters as image I_n. A further motion estimation using block matching algorithm is carried out between I_n and I^*_{n-k} in order to evaluate the residual camera parameters. These to sets of global parameters are combined to obtain the final one, that define the camera motion.

1.2 Local Motion Estimation

The aim of this step is twofold: 1) detect only object displacements, reducing the noisy effect due to the change of camera parameters and 2) obtain a dense motion field. Applying both block matching and differential techniques, it's possible to estimate large displacements with good accuracy and high spatial resolution. The idea is that block matching gives a first displacement field which is further refined by the differential algorithm [3]. In fact differential algorithm has been applied using a recursive procedure starting from the block matching motion field. Improvement is obtained considering the case of different time subsampling factor (k) used in sequence analysis. If the sequence under analysis is no subsampled in time ($k = 1$), then an estimation using block matching algorithm followed by differential algorithm estimation is performed. Application of differential algorithm steps allow us to obtain a dense local motion field useful for the successive segmentation phase. If the sequence under analysis is subsampled in time ($k > 1$), then block matching algorithm works with I_n and I^*_{n-k} images by using all images between I_n and I^*_{n-k}. The intermediate frames are compensated considering a linear variation of global motion parameters through I_n and I^*_{n-k} images (obtaining $I^*_{n-1}, \dots, I^*_{n-k+1}$). Block matching algorithm is applied with a correlation measure that takes into account a global distortion, considering temporal and spatial constrains. This measure evaluates and indicates, on all the images $(I_n, I^*_{n-1}, \dots, I^*_{n-k+1}, I^*_{n-k})$, the way the considered block has been matched. After this step, differential algorithm is applied to obtain a more dense local motion field. Fig. 1 shortly shows the process used to obtain local displacements passing through an original frame sequence (I_n and I_{n-k}) and a global parameter estimation.

2 Segmentation

Given a dense motion field, the task of segmentation is to identify coherent motion regions. The previous presented approach (global and local motion description) makes attractive an image segmentation based on motion field information. In fact after global motion compensation, all the background part of the scene will be stationary and the motion of the imaged objects will depend only by their 3D displacements in the space in front of the camera. Segmented regions are not overlapped in our algorithm. It means that if we call L the set of all image pixels, R_i $(i = 1, \dots, N)$ the obtained regions, then the N regions are a possible segmentation of the image L if and only if: 1) $\bigcup_i R_i = L, i = 1, \dots, N$, and 2) $R_i \bigcap R_j = \emptyset, \forall i \neq j$. For the velocity vectors we use an algorithm based on affine models [4]. Affine model can be used to describe congruent motion vectors in a compact way and its expression is:

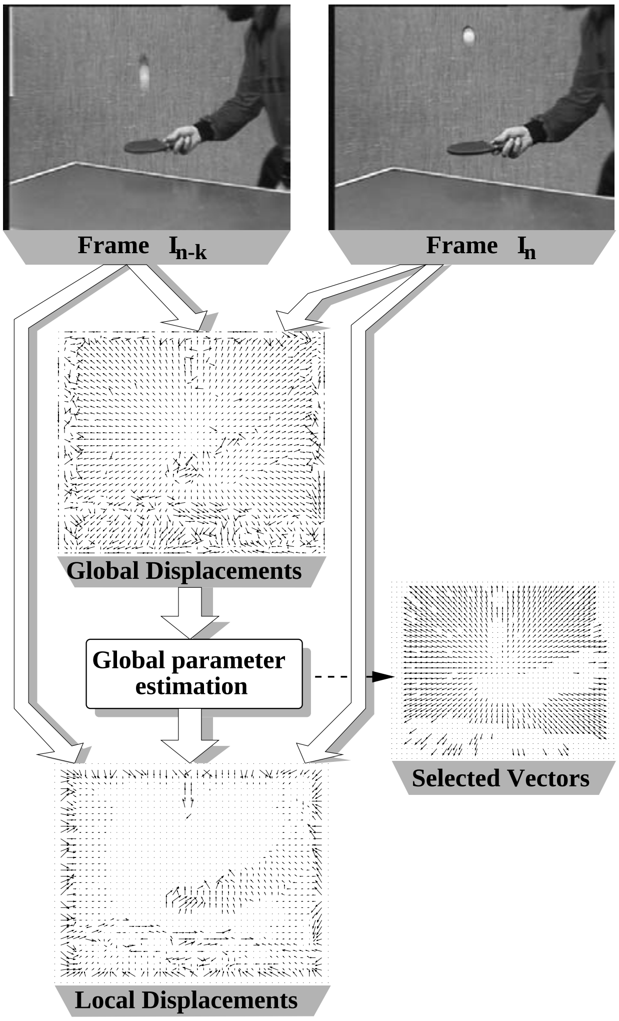

Figure 1: Local motion estimation steps.

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_{x0} \\ a_{y0} \end{bmatrix}, \quad (3)$$

where V_x and V_y are the velocity components of the point with (x, y) coordinates and the $\{a_{ij}\}$ are the affine model coefficients.

An iterative process is applied to the image in order to obtain the segmented image as result. In the clustering process for the objects, we assume two basic hypothesis: 1) they cannot be connected, and 2) they cannot appear and disappear in the analysed sequence. The first hypothesis takes into account that an object can be occluded by other objects, the second one takes into account that objects detected in the scene can: warp, stretch, zoom. They must be present on each frame of the sequence. This constrain is necessary to preserve stability of the algorithm and to ensure convergence. We start segmentation with arbitrary regions that are merged with affine criteria based on model coefficients. Merging methods, applied to generic regions R_i, are based on four criteria: 1) the region is deleted if all its points migrate into other regions, so that $card(R_i) = 0$; 2) the affine model of region R_i is similar to model of region R_j, so that $\{a_{kl}^{(i)}\} = \{a_{kl}^{(j)}\}$; 3) if the region becomes to small its points are labelled as *"no attributed"*; 4) if the MSE between affine model and vectors is too big, the region R_i and its points, are also labelled as *"no attributed"*. After a distance has been introduced, the two first merging criteria are spontaneous, while the last two are hard merging criteria based on thresholds. Initial merging phase is fragile especially if you apply hard merging criteria. Note that in our work it is not considered a region splitting procedure, that means that merged regions can not be further splitted. For this purpose hard thresholds vary with strictly decreasing functions. These threshold functions have been experimentally defined. At the end of each iteration a new compute of affine models is done, and all doubtful points, even those labelled as *"no attributed"*, have the possibility to stay in the same region or migrate into another one. At each iteration, the segmentation becomes more accurate because the parameter estimation of each affine motion model is calculated within a more coherent motion region. At the end of the iteration process, each segmented region has specific local affine model coefficients that make possible the reconstruction of motion vectors for every point without taking memory of the entire motion field. The affine parameters within regions are estimated at each iteration by standard linear regression techniques. Since the affine model is a linear model of the local motion, this estimation can be seen as a plane-fitting algorithm in velocity space. The analysis maintains temporal coherence and stability of segmentation by using the current motion segmentation results to initialize the segmentation for the next pair of frames. As above mentioned, segmentation process works on the dense motion field obtained between image I_n and I^*_{n-k}, so in order to follow objects through different frame sequence global motion parameters have to be considered. Furthermore, besides segmentation aim, the affine coefficients could be used to predict the motion information in an efficient way. In fact, in the prediction case instead of transmitting a dense motion field, we can only transmit the knowledge of the affine model for each segmented region.

3 Results

All our algorithms work on image luminance, without taking in account any colour image. The proposed technique described below has been tested on several sequences. Some results relative to the CIF (image characteristic: dimension 360×288 and 8 bit/pel) sequence *Table Tennis* are given. A subsampling factor in time of $k = 2$ is considered. Each image of the sequence is predicted from the previous one using only motion information.

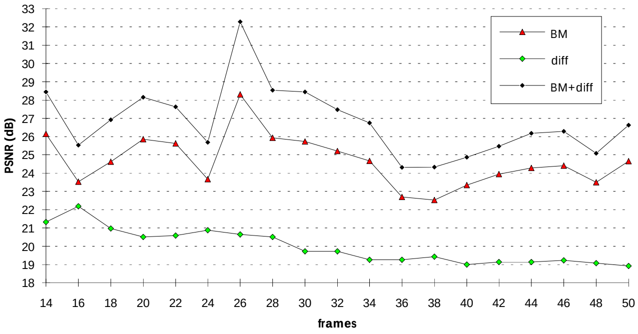

Figure 2: *Table Tennis sequence*: comparison results by using the different motion estimation algorithms.

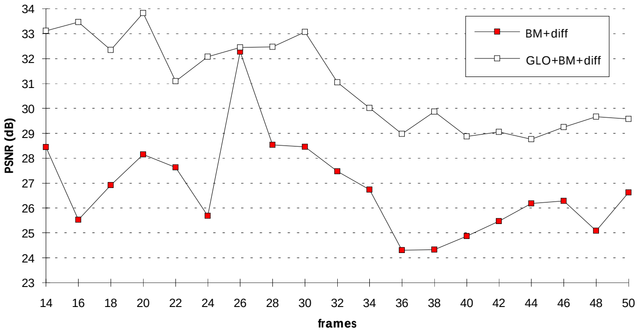

Figure 3: *Table Tennis sequence*: comparison results with and without global motion.

3.1 Motion field

A full-search block matching algorithm- with block dimension of 8×8 and with research windows of 40×40 as indicate in [1]- and differential algorithm -recursive until 2×2 block size with a ring zone around a considered block to avoid blocking effects on motion field- are implemented. The ring zone is chosen to be one pixel for 4×4 block size, and two pixels for 2×2 block size. Recursive depth of differential algorithm until 1 × 1 block dimension (pixel dimension) has been verified to be without a relevant improvement and time consuming, so it has been discarded. Fig. 2 gives the *PSNR* of predicted images obtained by using the presented algorithms: block matching (*BM*), differential (*diff*), combination of block matching and differential (*BM+diff*), without global motion compensation. In this result, the strong subsampling factor ($k = 2$) reduces performance of differential algorithm which bases its work hypothesis on small displacements of image objects. The combination of both block matching and differential (*BM+diff*) always offer the best result. The improvement of *PSNR* is in average 2.12dB and 6.82dB respectively to (*BM*) and (*diff*). The introduction of the global motion estimation brings to better performance, especially when a zoom occurs in the scene. Fig. 3 shows *PSNR* results obtained by introducing global motion and combination of both block matching and differential (*GLO+BM+diff*)and as comparison the best result obtained in previous step (*BM+diff*). In average the (*GLO+BM+diff*) overcome 4.22dB the (*BM+diff*) one. Note that in fig. 3 frames with high *PSNR* difference (frames 14, 20 and 36) are those where the highest zoom values have been estimated.

3.2 Segmentation

Fig. 4 shows the segmentation map obtained for a frame of the sequence *Table Tennis*: each affine model region is depicted by a different grey level. As we have already said, some frames of *Table Tennis* sequence contain a strong zoom factor, the main responsible of object deformation. In order to merge regions, we used as threshold law that decreases in passing from a couple of images to another. The segmentation map is not accurate on object borders, that because we segment with motion field information without taking into account any luminance information. Algorithm performance is currently implemented to improve the segmentation process with initial regions extracted from luminance border and add constrains in point assignments based on luminance edges.

4 Conclusion

In this work, we have studied the possibility to estimate reliable motion field by considering both a global motion, due to camera parameters changes, and a local motion, due to the displacement of the image objects. A combination of a block matching and differential algorithm has been used to obtain a dense local motion field. Simulation results proved that the detection and compensation of the global motion are essential for a good motion filed estimation and a motion compensated prediction. Furthermore, in order to detect the moving objects present in the scene, the obtained local motion field has been used as input for a segmentation algorithm based on affine model. In this case too, the estimation of the global motion is very useful to increase the quality of the obtained results.

Acknowledgements

The authors thank Prof. Stefano Tubaro for his useful theoretical support.

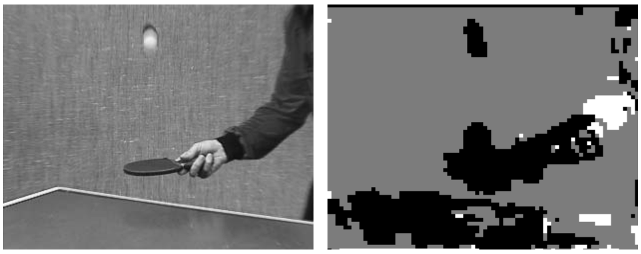

Figure 4: *Table Tennis*: Original frame (left); Segmented frame (right).

References

[1] D. LeGall, "Mpeg: A video compression standard for multimedia," *Signal Processing: Commun. ACM*, 34:47–58, 1991.

[2] P Migliorati, S. Tubaro, "Multistage motion estimation for image interpolation," *Image Communication*, vol. 7, June 1995.

[3] C. Cafforio and F. Rocca, "The differential method for image motion estimation," *NATO ASI Series*, F2, 1983.

[4] John Y. A. Wang and Edward H. Adelson, "Representing Moving Images with Layer," *IEEE Transaction on Image Processing*, 3(5), September 1994.