MOTION AND REGION OVERLAPPING ESTIMATION FOR SEGMENTATION-BASED VIDEO CODING

Montse Pardàs, Philippe Salembier, Benigno González
Dept. Teoria del Senyal i Comunicacions
E.T.S.E.T.B - U.P.C.
Apdo. 30002, 08080 Barcelona, Spain
Tel: (343) 401 6454, E-Mail: pardas@tsc.upc.es

ABSTRACT

In object-based video coding systems, the scenes are described in terms of three dimensional objects, which can be coded as textures and contours. However, in order to achieve high compression ratios, the redundancy in the temporal dimension must be exploited using motion compensation.

In this context, motion compensation can take advantage of the description of the scene as three dimensional regions. In this paper we present a motion estimation method which finds, in a fast and computationally simple manner, the affine parameters for every region of the scene. These parameters correspond to the motion of the texture. Following, an ordering between regions is established in order to have the right compensation of contours.

1. INTRODUCTION

In the framework of very low bit rate video coding, there is an increasing interest in second generation image compression methods [1]. These techniques aim to eliminate the redundant information within and between frames taking advantage of the properties of the human visual system. In particular, segmentation-based compression methods describe the scenes in terms of textured regions surrounded by contours, in such a way that the regions correspond, as faithfully as possible, to the objects in the scene. These techniques have recently been used for very low bit rate video coding [2],[3]. They describe the scene in terms of objects in a 3D space (2 spatial dimensions plus the time dimension). However, contours and textures are not efficiently coded in a three dimensional space: the discrete nature of the time dimension leads to great discontinuities (which are even stronger if low transmission frame rates are used). To reach very low bit rates, motion compensation has to be used.

Motion estimation within a contour/texture coding approach presents some specific issues. For example, if a 3D segmentation is available, the region correspondence problem is solved. Therefore, a simple approach consists of using a model of motion and of estimating the model parameters for each region. This method is used in [4] for a purely translational model and results in a simple region matching. In [5], a similar approach relying on an affine model is taken. However, these methods only take into account the shape of the regions. As consequence, they give good results for contour compensation but not for texture compensation. Moreover, the problems of texture or of contour motion estimation are not equivalent because of the presence of background regions and foreground regions. Indeed, if a foreground region is moving in front of a background region, the motion of the transition (contour) between the two regions does not say anything about the motion of the background region. An efficient motion compensation technique should rely on both the motion of the texture and an ordering relation between regions. In this way, an ordered motion compensation of the regions will result in a good compensation of contours and of textures.

In this paper, a method for estimating the texture motion of regions and for computing the ordering relation between them is presented. The method proposed for estimating the texture motion finds the affine parameters of the regions. A dense motion field is first computed using a block matching algorithm. From this motion field the aforementioned parameters are extracted. The block-matching is confined to the inside of the regions in order to have an estimation which is not affected by the contours. Following, by estimating the ordering relation between neighboring regions the right compensation of both, contours and textures, can be performed.

In section 2 of this paper we comment the conflict between contours and textures in this motion estimation problem. In section 3 the method used for the texture motion estimation is described. Following, in section 4 it is explained how the order relation between regions is computed. Section 5 describes the compensation process. Finally, some results of this methods are shown and conclusions are reported.

2. CONTOUR VS TEXTURE MOTION ESTIMATION

In an object-based video coding technique the problem of motion estimation has different characteristics than in a general context. In these techniques we are usually dealing with three dimensional segmentations [2], [3], [6] which describe the scene as 3D regions and so intrinsically solve the region correspondence problem. So, the motion estimation problem is only a problem of choosing the right model for the regions motion and estimate the parameters of this model for every region. However, when estimating these parameters the difference between contours and textures has to be taken into account. Take for instance the example of Figure 1. We have in this figure two balls A and B. Ball A is moving with a translational motion towards B, while B is fixed.

Furthermore, the relative position between them places A in the foreground and B in the background. Suppose that these pictures have been correctly segmented as 3D regions and so we know the correspondence between the spatial regions in the two pictures. It can be observed that, if we estimate the motion of A, we can consider both its texture and its contours, because they are suffering exactly the same motion. However, this is not the case for ball B. The texture of B is static, because the ball is not moving. By contrast, its contours are shrinking, due to the motion of A on its foreground. This implies that a distinction has to be made between contours and textures in the motion estimation problem.

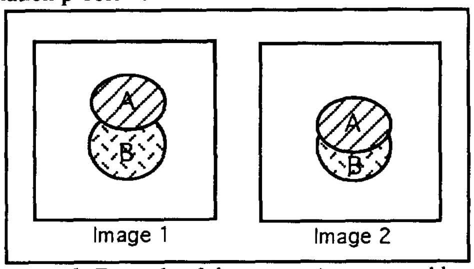

Figure 1. Example of the contour/texture problem

Actually, the right motion of the object is defined by the motion of its texture, and when it does not coincide with the contour motion it is because we are dealing with an object in the background. However, in video coding systems where contours and textures are coded by motion compensation, we might be interested in both results. That is, when compensating contours for coding them, we are interested in having the best compensation of the contours, even though the motion estimation does not coincide with the real motion that the object has suffered. This is the approach taken in [4] where a region matching is performed based exclusively on the contours. In [5] it was proposed to perform two different motion estimation, in order to have different compensations for contours and textures. The first one would be used for contour coding and the second one for texture coding.

However, if the ordering between neighboring regions is known it is not necessary to compute the motion for the contours. It is enough with performing in the right order the compensation of every region using the texture parameters. For instance, in Figure 1, we could first make the compensation of the background. Following, the balls A and B could be compensated, and in the pixels which could belong to both compensations, the values of A would be assigned, as it is in the foreground of B. This process would lead to the right contours for ball B.

3. TEXTURE MOTION ESTIMATION

For determining the texture motion of each region, different methods can be used. However, due to the fact that the motion will be estimated in a region basis, and regions may be large, it is advisable to use a model which can cope with different kinds of motion. For instance, using only translational motion would produce a too poor estimate. For this reason, in [5] it was proposed to use an affine motion model, which can deal with translation, zoom and rotations. The problem of this approach is that it has a rather high computational cost. The parameters of the affine model for every region have to be estimated with an iterative procedure.

Furthermore, special care has to be taken in order not to let the contours have an influence on this estimation. This implies that the motion estimation will have to be performed in a sub-region contained inside the region of interest.

The method which we propose in this paper finds the parameters of the affine model with a fast algorithm which does not require iterations. These parameters are computed with a closed formula from a dense motion field which is previously computed inside the regions.

3.1. Dense motion field

The dense motion field is obtained with a block-matching algorithm. Suppose we wish to obtain the motion field between frames t and t+1. Every region in the frame t is divided in small square blocks, avoiding the blocks which include parts of the contours. Inside these blocks it is assumed that the motion is uniform and translational. Thus, the best matching block in frame t+1 is searched for every block in t. The position of this best matching block gives us the displacement vector for every pixel inside the block of frame t. The size of the blocks is adapted to the regions shape. The problem of the blocks size in a block-matching algorithm comes from the fact that in large blocks the assumption of uniform translational motion inside the block is farther from reality than for small blocks. However, small blocks can produce noisy estimates. In our algorithm we use a block size of 8X8 pixels, which is small enough to guarantee that the hypothesis made is not far from reality. Although some noisy estimations appear, they are compensated in the next step, when the global parameters for the entire region are obtained. In order to get enough information for small regions, a minimum size of 4X4 is allowed. Blocks of 8X8 are divided in smaller blocks when they do not completely belong to the same region. Blocks of 4X4 which do not completely belong to a region are discarded. For the remaining blocks, the block-matching algorithm is applied to obtain a displacement vector for every block. The blocks which best prediction for the position in the next frame does not belong to the same region, or falls on a contour are also discarded. Thus, we ensure that all the blocks which are taken into account in frame t belong to the texture and are not affected by the contours.

3.2. Global region motion parameters

From the displacement vectors obtained in the previous step we wish to obtain a more compact representation of the motion, due to the fact that the division in blocks of 8x8 and 4x4 produces too many information to be coded in a very low bit rate coding system. Moreover, the blocks belonging to the same region will usually have a coherent motion. We will use an affine model for describing this motion, which can take into account translation, rotation and expansion. This model uses six different parameters However, it can be simplified in order to have only four parameters (excluding rotation) or three (making the expansion in the horizontal axis equal to the expansion in the vertical axis).

In [7] the model with three parameters was derived to compute the global motion of an image. We will follow a similar development for the six parameters model to estimate the model in every region.

Let us call u=(x,y) and u'=(x',y') the space image coordinates of a given point, in two successive frames, before and after motion respectively.

Assuming that the motion for every region can be defined with two parameters of expansion, two of rotation and two of translation, the relation between the positions u and u' can be expressed as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (1)$$

or, in matricial form,

$$u' = Au + b \quad (2)$$

Then, these parameters have to be found for every region, using the information obtained in the block matching algorithm.

The block matching algorithm has measured the displacement vector for the pixels of the regions:

$$u_i' = u_i + d_i \quad (3)$$

u_i' being the positions of the center of the blocks belonging to the regions.

Hence, the elements of the matrices A and b have to be computed in such a way that the estimated displacement vectors with these parameters best match the measured ones. The estimated displacement vectors can be computed as:

$$\hat{d}_i = \hat{u}_i' - u_i = Au_i + b - u_i = (A - I)u_i + b \quad (4)$$

Taking into account that every computed displacement d_i is applied to a number s_i of pixels belonging to the same block (s_i depending on the size of the block) and taking as measure of distortion the l_2 norm of the displacement error, we obtain:

$$E(A,b) = \sum_i s_i \left\| \hat{d}_i - d_i \right\|^2 = \sum_i s_i \left\| \hat{u}_i' - u_i' \right\|^2 \quad (5)$$

where the summation is over the displacement vectors belonging to the regions.

So, the estimation of the parameters consists in minimizing

$$\min_{Ab} E(A,b) \quad (6)$$

By differentiating and setting the derivatives to zero we obtain the optimal solution:

$$A = \left(\sum_i s_i u_i' u_i^T - \frac{1}{\sum_i s_i} \left[\sum_i s_i u_i' \sum_i s_i u_i^T \right] \right) \cdot \left(\sum_i s_i u_i u_i^T - \frac{1}{\sum_i s_i} \left[\sum_i s_i u_i \sum_i s_i u_i^T \right] \right)^{-1} \quad (7)$$

$$b = \frac{1}{\sum_i s_i} \left[\sum_i s_i u_i' - \sum_i s_i A u_i \right] \quad (8)$$

These matrices define the forward motion parameters for every region. An equivalent relationship can be found for the backward motion parameters:

$$u = A^{-1}u' - A^{-1}b \quad (9)$$

$$A_B = A^{-1} \quad (10) \qquad b_B = -A^{-1}b \quad (11)$$

$$u = A_B u' + b_B \quad (12)$$

4. ORDERING OF THE REGIONS

The motion parameters computed in the previous section can be used for compensating the texture of the regions. However, its contours have to be defined. As explained in Section 2, the contours will be defined establishing an order between neighboring regions which will be used in the compensation process.

The order relation will be defined for every couple of neighboring regions. That is, for every frontier between two regions a two hypothesis test will be performed in order to determine which of the two regions is on the foreground. The parameters computed for every region are used to calculate the prediction error for the two possible order relations. The order producing the lowest prediction error is assumed to be the correct one.

5. CONTOUR AND TEXTURE COMPENSATION

The compensation of the contours will be performed through the compensation of the segmented image or partition. Every region in frame t will produce its corresponding region in frame t+1, using the computed parameters. When there is a problem because a pixel in frame t+1 corresponds to more than one compensated region, the ordering relation will be used for deciding which one is on the foreground. This will produce a unique partition or segmented image in frame t+1. Let us explain in more detail how this process is performed:

First, all the pixels in every region of frame t are motion compensated and so projected in the next frame. However, due to the fact that we use parameters of expansion for the motion, the position of the projected pixels will not always precisely coincide with the position of a pixel in t+1. So, every projection will be approximated to its nearest pixel position. Nevertheless, this process will not fill every pixel of the inside of the regions (in case of expansion). For this reason, it is not enough to check the order relation for the pixels where two or more regions produce a compensated pixel (if the first region filled one pixel directly and the second one only by interpolation it would get the label of the first region). So, at the same time that a region is compensated it will be checked for every pixel that none of the regions which are on the foreground of the first one can produce a projection at the same position. This is performed applying the backward parameters of the new region to the pixel under consideration and checking if the corresponding pixel is actually of this new region.

Observe that new neighboring regions can appear in this process. That is, two regions which were not neighbors in frame t can become neighbors in frame t+1 due to the compensation process. In this case, this new relation will have to be studied and added to the list of neighboring regions.

Finally, it is possible to have pixels which have not been filled by any region (pixels between regions). These pixels will be left without any label and it will be in the contour coding part where they will be filled. In this procedure, the error between the contours obtained and the computed segmentation of this frame has to be coded.

Once the compensated partition has been defined, the texture inside every region can be computed using the same parameters and the residue can be coded.

6. RESULTS

In Figure 2 we present an example of the motion estimation process. It corresponds to the sequence Table Tennis. In first row we show the original images. The first image of the second row corresponds to the motion vectors estimated with the block matching algorithm. The block sizes which have been used are 8 and 4. As it can be seen, there are many noisy vectors. The second image of this row presents the motion vectors obtained using the method described in this paper to obtain four parameters for every region. The model has been simplified and the rotation parameters are not used. It can be observed how the motion field obtained in the block matching has been correctly approximated and how the noisy vectors have been eliminated.

In Figure 3 we present the segmentation which has been used for the images in Figure 1, and how it has been compensated. The first row presents the segmentation for the two original frames. The compensation of the first frame using the parameters obtained and the ordering relation which has been described is shown in the second row.

7. CONCLUSIONS

In this paper we present a method to perform the motion estimation in an object based video coding system. This method has the following advantages with respect to previously used techniques:

- It solves the problem of motion estimation of contours and textures together, without having to perform two different estimations and to send two sets of motion parameters, only the order relation between regions is needed additionally.
- It allows the introduction of a more complex model than the translational one [4] (which is not adequate for large regions), while avoiding the high computational cost required to estimate a complex model directly on the original signal [5].

ACKNOWLEDGEMENTS

This work has been performed within the framework of the RACE MORPHECO project and has been partially supported by a grant of the Comissionat per Universitats i Recerca, Generalitat de Catalunya.

REFERENCES

[1] M. Kunt, A. Ikonomopoulos, M. Kocher, "Second generation image-coding techniques", *Proceedings of the IEEE*, Vol. 73, 1985, pp. 549-574.

[2] S. Rajala, M. Civanlar and W. Lee, "Video data compression using three-dimensional segmentation based on HVS properties", *Proc. of IEEE International Conference on Acoustics Speech and Signal Processing'88, ICASSP'88*, New York, NY, USA, 1988, pp.1092-1095.

[3] P. Salembier, M. Pardàs, "Hierarchical morphological segmentation and coding for image sequences", to be published in IEEE Transactions on Image Processing, Special Issue on Image Sequence Compression, Vol. 3, nº5, September 1994.

[4] C. Gu, M. Kunt, "3D Contour image sequence coding based on morphological filters and motion compensation", *in Proc. of the international Workshop on Coding Techniques for Very Low Bit-rate Video*, Colchester, UK, 7-8 April 1994.

[5] M. Pardàs, P. Salembier, "3D Morphological Segmentation and Motion Estimation for Image Sequences", to be published in *Signal Processing* , EURASIP, Special Isssue on Mathematical Morphology, Vol. 38, Nº 2, September 1994.

[6] P. Willemin, T. Reed and M. Kunt, Image sequences coding by split and merge, *IEEE Transactions on Communications*, Vol. 39, No12, December 1991, pp 1845-1855.

[7] Y. Tong Tse, R. L. Baker, "Global zoom/pan estimation and compensation for video compression", in Proceedings of the ICASSP-91, Toronto, USA, pp. 2725-2728.

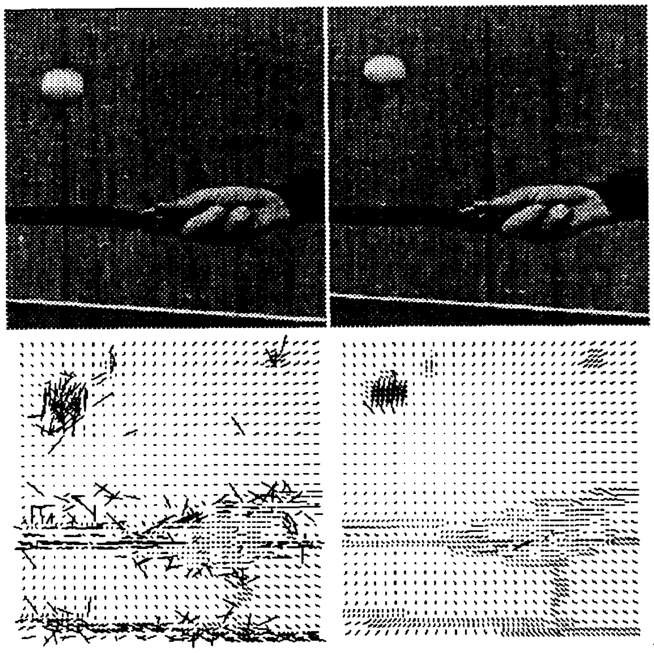

Figure 2. Example of the motion estimation process

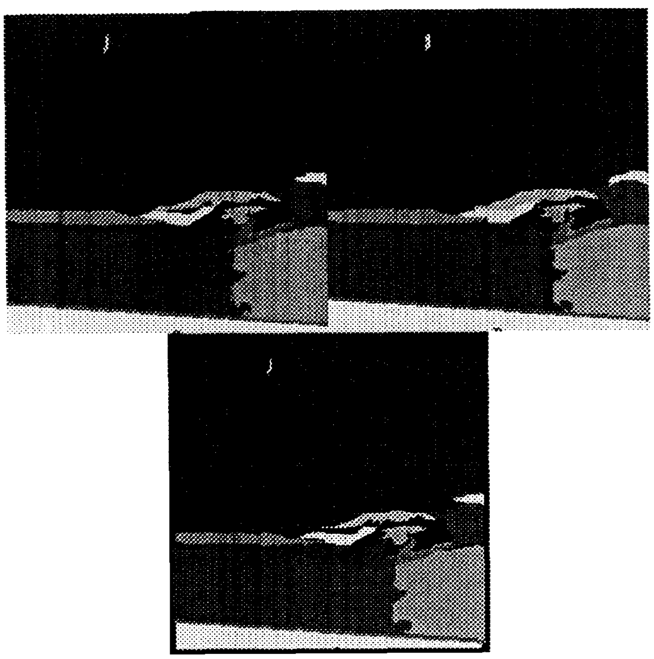

Figure 3. Example of the compensation of the segmentation.