A JOINT MOTION COMPUTATION AND SEGMENTATION ALGORITHM FOR VIDEO CODING

Sylvain BOLTZ, Éric DEBREUVE, Michel BARLAUD

Laboratory I3S, UMR 6070 of CNRS, University of Nice Sophia Antipolis
Bât Algorithmes/Euclide B, 2000 route des lucioles - BP121 - 06903 Sophia Antipolis Cedex, France
Tel : 33 (0)4.92.94.27.87 Fax : 33(0)4.92.94.24.98 - {boltz,debreuve,barlaud}@i3s.unice.fr

ABSTRACT

Motion compensation is an essential problem in video coding. The main drawback of the usual motion estimation methods is that they divide the images into blocks or patches which do not correspond to moving objects. In this paper, we propose a method to estimate the motion in regions instead of blocks. We define a cost functional to estimate simultaneously the segmentation and the motion of the regions. We introduce a joint motion estimation and segmentation algorithm based on the derivation of this cost functional. We show some encouraging results for video compression.

1. INTRODUCTION

The goal of this paper is to propose a method to estimate motion in video sequences for coding purposes. Most video coders, including MPEG-like coders and more recent wavelet-based coders [7], use simple motion estimators based on block matching algorithms. Even though these algorithms are fast and quite accurate, they still have some issues : indeed, since the subdivision into blocks does not match the positions of the moving objects, some blocks overlap regions with different motions, which creates blocking artifacts in the coded-decoded sequence.

To overcome this problem, one should consider a segmentation of the moving objects. Unfortunately, most of the image segmentation techniques based on active contours [4, 8, 9, 10, 12, 13] are not fully automatic, and they are too complex to be implemented within a video coder. For this reason, we propose a simplified active contour approach to estimate motion and segmentation simultaneously in a local context. More precisely, we divide the image into macroblocks in which segmentation is performed independently. We suppose that there are at most two regions with different motion in each macroblock.

In order to decide whether each macroblock should be split or not, a block selection process will be run. The selected blocks are now distinct joint motion segmentation problems and we will now define a cost functional to solve them simultaneously. Finally, we will show how this model allows to avoid occlusion.

This technique is unable to provide a global segmentation of the image, but this information is not mandatory for coding purposes.

2. BASIC IDEA AND MOTIVATIONS

2.1 Blocks versus Regions

In classical block-based motion estimation methods, when the prediction error in a block is above a threshold, the block is further divided into smaller blocks. We will show in this paper that we obtain better results by dividing the blocks into regions instead, using splines [15] or at least discontinuous straight lines.

This model might not perform better than the block-based model with small blocks in all cases. However it should be a good alternative specially on object borders, as shown in figure 1

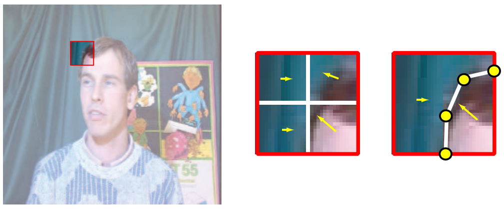

Figure 1: Splines region better than block-based region

2.2 Transmission cost

Let us compare the transmission cost between the two methods : the first method divides a block into 4 smaller blocks (4 motion vectors); whereas the second method divides it into 2 regions (4 control points and 2 motion vectors), figure 1. The precision of the motion vectors is typically 1/8 pixel precision, and the precision of control points is one pixel. Moreover 2 out of 4 control points are on the border of the block so these points are represented by only one parameter. The region-based representation can be coded using 5 vectors: 2 control points, 1 vector made of 2 parameters and 2 region motion vectors, versus 4 vectors with smaller blocks. The transmission cost in each macroblock is then : 2 motion vectors at 1/8 pixel precision plus 3 control points position at pixel precision. With the smaller blocks method it would be 4 motion vectors at 1/8 pixel precision. In a coder, this represents $64 bits/$macroblock for our method versus $64 bits/$macroblock for the block-based method. Assuming that not all blocks are divided by a spline, we even obtain a smaller motion information in the compressed video with the bonus of a greater spline precision Fig. 1.

3. SEGMENTATION AND MOTION ESTIMATION

3.1 Criterion definition

A region in the frame is defined by the optical flow. Let $I(m,i)$ be a video sequence, m the spatial coordinates, i the frame number, and v the optical flow between image i and image i+1. v is a vector field representing an apparent motion related to a local gray-scale coherence between two consecutive images [6, 5, 14, 16, 11] .

$$(I(m,i) - I(m+v,i+1))^2 = 0 \qquad (1)$$

In general, Eq.(1) has several solutions, since many points in an image have the same gray-scale value. Therefore, the problem of computing the motion of a point must be regularized. First, Eq.(1) can be extended to a domain surrounding this point, second, we assume that the optical flow v constant over a region . Therefore, region should be a minimizer of the following energy (2):

$$\begin{cases} J(\ ) & = \int (I(m,i) - I(m+v(\ ),i+1))^2 dm \\ v(\ ) & = arg\min_{v} \int (I(m,i) - I(m+v,i+1))^2 dm \end{cases} \qquad (2)$$

The cost functional J (2) is minimized to solve motion and segmentation problems simultaneously.

For higher robustness, the functional is defined on a set of two frames surrounding the image of interest, a forward and a backward frame (previous equation is forward only), and we constraint the motions v computed backward and forward to be equal i.e uniform motion assumption.

$$\begin{aligned} k(m,v) &= (I(m,i) - I(m+v,i+1))^2 \\ &+ (I(m,i) - I(m-v,i-1))^2 \end{aligned} \qquad (3)$$

$$\begin{cases} J(\ ) & = \int k(m,v(\ ))dm \\ v(\ ) & = arg\min_{v} \int k(m,v)dm \end{cases} \qquad (4)$$

Our cost functional (4) must be minimized in each macroblocks. An alternate minimization algorithm is then defined in each :

1. Estimate the motion v^* for a fixed segmentation (constant) in section **3.3.**
2. Estimate the segmentation $\ ^*$ for a fixed motion (v constant) in section **3.2.**
3. Iterate until convergence.

3.2 Segmentation

For a fixed motion, in order to find the region that minimizes the cost functional, we use a region competition algorithm. For instance, the functional J for two regions including a regularization term can be written as follows:

As the two regions form a partition of the block there is only one unknown , with $\ = \ _1$, $\overline{\ } = \ _2$ and $\ = \ $.

$$J(\ ) = \int k(m,v(\ ))dm + \int_{\overline{\ }} k(m,v(\overline{\ }))dm + \int \ dt \qquad (5)$$

The first and second term are the energy (2) applied on the two regions of the block, and the last term is the regularization term where is the contour between the two regions and a constant.

3.2.1 *Energy derivative using shape gradients*

Derivating this functionnal on a region is not easy, moreover when the criterion terms $k\ (m,v)$ are region-dependant. A shape gradient model [4, 12, 13] is used to make the energy depend on an evolution parameter .

$$J(\ (\ ),\ _1(\ ),\ _2(\ )) = J(\ ) \qquad (6)$$

Jehan Besson et al. [4, 13] then proposed a method to compute the eulerian derivative J' :

$$\begin{aligned} J'(\ ) &= \int_{(\ )} (k\ (m,v_2) - k\ (m,v_1)(V.N)ds \\ &+ \int_{(\ )} (-\ \ )(V.N)ds \end{aligned} \qquad (7)$$

V is the unknown local deformation of and N is the inward unit normal to .

3.2.2 *Evolution equation*

The derivative (7) must be negative to go towards to the minimum of the functionnal. The evolution equation $\frac{\ }{\ }$ is then:

$$V = \frac{\ }{\ } = (k(m,v_1) - k(m,v_2) + \ .\ ).\overrightarrow{N} \qquad (8)$$

We use active parametric contours to model the boundary . is represented by an open spline, the first and last control points of the splines are located on a block border and their evolution are also projected to stay on the border. An explicit parametrization of the active contour is performed by interpolating a spline between the control points.

3.3 Motion estimation

A classical matching method is used to compute motion. However the matching is performed with regions instead of blocks, and more precisely a fast suboptimal matching algorithm is used: the Diamond Search [17]. We recall that the matching criterion is (See Eq. (4)) for a fixed .

$$v(\ ) = arg\min_{v} \ _{x\in} k(m,v) \qquad (9)$$

To further improve the results, the motion is computed using the YUV color components using a color weighting as in [3].

In addition, as the alternate minimization process provides a slightly modified region from the last iteration, we can initiate the Diamond Search algorithm with the previously computed motion.

4. IMPLEMENTATION

4.1 Initialization and block selection

A block of homogeneous motion does not require to be split by a spline. The algorithm must select the blocks to be divided and, at the same time, must initialize a first spline in these blocks. This selection is a three-step procedure:

- First, every block is divided into 4 smaller blocks. Then are computed the motion vectors in these smaller blocks with a block matching algorithm. Finally we compute a normalized distance between the each pairs of vectors:

$$d = \max_{i=1..4, j>i} \frac{\|v_i - v_j\|}{min(\|v_i\|,\|v_j\|)} \qquad (10)$$

  and we threshold this value to choose if a block should be split by a spline or not.

- As a requirement for compression applications, we threshold the mean value of the prediction error at initialization, which helps to produce an effective segmentation for video coding rather than a regularized one.
- Finally, we threshold the same criterion (10) applied to the motion vectors of the two regions delimited by the spline

In addition, we use the first threshold to initialize a first spline in the block : the blocks (i, j) found by maximizing d in (10) defines the two classes of motions. The two other blocks are classified whether they are closer from the motion of i or the motion of j, closer in the sense of the same normalized distance. The motion classification leads to six different possible initializations made of control points splitting the blocks.

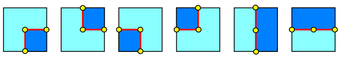

Figure 2: Six possible initializations

4.2 Topology management

We assume that a block is composed of at most two connex regions separated by a spline. However if the spline reaches a border, it splits the block in 3 regions; in this case the spline is cut into two parts and the shortest one is discarded, so that only two regions remain (See Fig. 3).

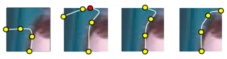

Figure 3: Topology Management

5. EXPERIMENTAL RESULTS

The proposed method was tested on the sequence ”Erik”, on the frames 18 to 24 which represent a quite uniform translation, needed by our bidirectional constraint.

5.1 Segmentation analysis

Let us analyze the result on the mid frame 21, the result is shown in Fig. 4. The method seems to perform well. Note that some blocks at the bottom of the frame were not divided by a spline because the background is quite homogeneous, so even with Erik’s motion, the prediction error is lower than the energy’s threshold.

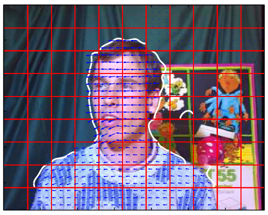

Figure 4: Macroblock spline segmentation

5.2 Prediction error

In order to estimate the performances in terms of prediction error the proposed segmentation method was applied to the 8 frames of sequence Erik. Each frame was processed using the next and previous frame as detailed above. The Tab. 1 presents the prediction error energy per frame (PEEF), averaged over the 8 frames. The PEEF is defined as follows:

- Case with 4 blocks: the PEEF is equal to the sum of the prediction errors (from (3)) of the 4 blocks composing a macroblock, summed over each macroblock in which inhomogeneous motion was detected.
- Case with 2 regions: the PEEF is equal to the sum of the prediction errors (from (3)) of the 2 regions composing a macroblock, summed over each macroblock in which inhomogeneous motion was detected, i.e split by a spline.

By definition, in both cases, the same macroblocks are considered. The proposed method leads to a decrease of about $1/3$ of the average PEEF on the 8 frames.

Table 1: Energy of prediction error on the split macro-blocks: spline method vs smaller blocks method

Macroblock division	4 blocks	2 regions	gain in %
Average PEEF	59.9	39.1	33.90

6. OCCLUSIONS MANAGEMENT

Figure 4 shows that segmentation splines are actually located a few pixels away from the object to be segmented. This is due to the background being occluded by Erik. Indeed, since criterion (3) is bidirectional, occluded background parts are on the both sides of Erik. The occluded background on the Erik’s sequence represents up to 10% of a block, which is much more important than in a classical algorithm on the whole image where occluded parts represent about 1% of the image. Thus we must take occlusion into account, as the problem is local we suppose there is only one kind of occlusion which happens only in backward or forward estimation. To correct this problem, a weighting between forward and backward estimation will be used. For the bidirectional prediction, the weighting is the same in both directions, we will adjust the weightings in forward prediction or backward prediction if occluded parts are detected. The new criterion with weightings is thus described in (11).

$$\begin{aligned} k(m,v) &= c_f * ((I(m,i) - I(m+v,i+1)))^2 \\ &+ c_b * ((I(m,i) - I(m-v,i-1)))^2 \end{aligned} \quad (11)$$

where c_f and c_b are respectively the forward and the backward weightings.

6.1 Occlusion detections

The occlusion detection method is now to be defined as well as the set of weightings. The constraint Block Matching algorithm gives us two values of the criterion (9): one forward and one backward. Comparing these two values, we can assume that if the forward (resp. backward) criterion value is some percentage higher than the other backward (resp. forward) criterion value there is an occlusion problem, so we set

c_f (resp. c_b) to 0 and the other to 2. Otherwise, we use the constraint bidirectional method, so c_f and c_b are set to 1.

6.2 Some results

We compare the segmentation results with and without occlusion management. We count the wrong-classified points in the two cases and we compare the results with a manual segmentation. There are 1400 wrong pixels using our method without occlusion management and only 1000 wrong pixels using our method with occlusion management, see figure 5. Visually, we observe an important diminution of the wrong pixels, the splines are much closer to Erik; we can also notice some improvements of the selection algorithm behavior; a spline at highlighted block wrongly removed by criterion (3), Fig. 4 are now back in the video, Fig. 5.

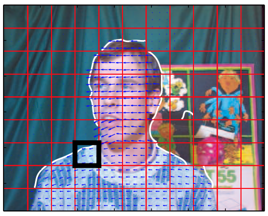

Figure 5: Macroblock spline segmentation with occlusions management

The results shown on the 8 frames in Tab. (2) in terms of PEEF (as defined in 5.2) are not better because our error criterion (3) does not take account on occlusions. However using an adaptative filtering, as presented in [1, 2], this accurate segmentation should provide better results.

Table 2: Energy of prediction error on the split macro-blocks, with occlusion management

Macroblock division	4 blocks	2 regions	gain in %
Average PEEF	57.9	42.8	25.83

7. CONCLUSION

We have described a joint motion segmentation and motion estimation algorithm. We adopted a simplified approach using macroblocks in which the problem is solved independently in each macroblock. We presented interesting first results on video coding. In future works, we intend to further improve the robustness and the efficiency of our algorithm by adding image gradient terms in our criterion. We will also integrate the proposed method into a full wavelet-based video coder and adapt our original optical flow error criterion to a wavelet subband error criterion.

REFERENCES

[1] T. Andre, M. Antonini, and Barlaud M. Full occlusion management for wavelet-based video coding. In *EUSIPCO*, 2005 to appear.

[2] T. Andre, M. Antonini, and Barlaud M. Puzzle temporal lifting for wavelet-based video coding. In *ICIP*, 2005 to appear.

[3] T. André, B. Pesquet-Popescu, M. Gastaud, M. Antonini, and M. Barlaud. Motion estimation using chrominance for wavelet-based video coding. In *Proc. IEEE Picture Coding Symposium*, San Francisco, USA, December 2004.

[4] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variations or shape gradients ? *SIAM Applied Mathematics*, 1(2):2128–2145, 2003.

[5] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. In *SIAM Journal of Applied Mathematics*, volume 60, pages 156–182, 1999.

[6] B.Horn and B.Schunck. Determining optical flow. In *Artificial Intelligence*, volume 17, pages 185–203, 1981.

[7] M. Cagnazzo, T. André, M. Antonini, and M. Barlaud. A model-based motion compensated video coder with JPEG2000 compatibility. In *IEEE Intern. Conf. on Image Processing*, pages 2255–2258, Singapore, October 2004.

[8] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[9] T. Chan and L. Vese. Active contours without edges. In *IEEE Transactions on Image Processing*, volume 10, pages 266–277, 2001.

[10] D. Cremers and S. Soatto. Variational space-time motion segmentation. In *Internation Conference of Computer Vision*, pages 886–893, 2003.

[11] E. Debreuve, M. Gastaud, M. Barlaud, and G. Aubert. A region-based joint motion computation and segmentation on a set of frames. In *WIAMIS*, Montreux, Swi, 2005.

[12] M. C. Delfour and J. P. Solezio. Shapes and geometries: Analysis, differential calculus and optimization. In *Advances in Design and Control. Society for Industrial and Applied Mathematics*, Philadelphia, 2001.

[13] S. Jehan-Besson, M. Barlaud, and G. Aubert. DREAM^{2}S: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. *IJCV*, 53(1):45–70, 2003.

[14] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 8, pages 565–593, 1986.

[15] F. Precioso, M. Barlaud, T. Blu, and M. Unser. Robust real-time segmentation of images and videos uisng a smoothing-spline snake-based algorithm. *IEEE Trans on IP*, 14(7), 2005.

[16] J. Weickert and C. Schnorr. Variational optic flow computation with a spatio-temporal smoothness constraint. In *Journal of Mathematical Imaging and Vision*, volume 14, pages 245–255, 2001.

[17] Shan Zhu and Kai-Kuang Ma. A new diamond search algorithm for fast block-matching motion estimation. In *IEEE Transactions on Image Processing*, February 2000.