A REGION-BASED JOINT MOTION COMPUTATION AND SEGMENTATION ON A SET OF FRAMES

É. Debreuve[†], *M. Gastaud*[†], *M. Barlaud*[†], *G. Aubert*[‡]

[†]Laboratoire I3S, UMR CNRS 6070,
2000, route des lucioles, BP 121,
06903 Sophia Antipolis Cedex, France

[‡]Laboratoire Dieudonné, UMR CNRS 6621,
Université de Nice-Sophia Antipolis, Parc Valrose,
06108 Nice Cedex 2, France

ABSTRACT

Classically, motion between two consecutive frames of a video sequence is computed by solving the optical flow equation or by block matching. In terms of data consistency, as opposed to regularization, the first approach is local and, therefore, it is ill-posed. Moreover, it follows from a linear approximation of the assumption of brightness constancy. The second approach is region-based, allowing computation of a motion more coherent with the data. However, it is computationally intensive and it sets a limit on the amplitude of motion. The computed motion can be used to segment moving objects. Conversely, motion can be computed more reliably if the moving objects are known. We propose a method of simultaneous motion computation and segmentation by minimization of an energy based on the assumption of brightness constancy. The proposed energy is defined on a region and relies on a general parametric motion model. Minimization is performed by a steepest descent algorithm for motion computation and active contour for segmentation.

1. INTRODUCTION

A classical method of motion computation between two consecutive frames of a video sequence is to solve the optical flow equation [1, 2, 3, 4, 5, 6]. Aside from regularization, this approach is local (or pixel-based) whereas motion is bound to objects. Moreover, if no parametric motion model is used, the problem is ill-posed. Finally, the optical flow equation is a linear approximation of the assumption of brightness constancy.

The standard method of motion computation in video codecs is block matching. This method is region-based, or rather, block-based. It allows to get a motion more coherent with the data although the blocks are independent of the objects. The motion model is classically translation. Sub-optimal search methods [7, 8] (as opposed to an exhaustive search method) can be used to achieve sub-pixel accuracy with a low computation time. However, there is a user-defined limit on the amplitude of motion (usually +15/-15 pixels in both horizontal and vertical directions).

We propose a method of motion computation by minimization of an energy based on the assumption of brightness constancy, *i.e.*, without linear approximation as in the case of optical flow. The proposed energy is defined on a region and relies on a general parametric motion model. Minimization is performed by a steepest descent algorithm [9]. We also propose to use this same energy to perform a spatiotemporal segmentation of the sequence using an active contour method. Actually, considering the motion as an unknown parameter of the segmentation problem, the proposed method jointly performs motion computation and segmentation. The energy is defined on a portion of the sequence composed of at least two frames and minimization is done on a set of frames.

2. COUPLED ENERGY

Let f be a video sequence composed of gray-scale images f_t. Mathematically, f is a function from $S \equiv D \times [0, T]$ into $\mathcal{R}^+$ where D is a bounded connected open set of $\mathcal{R}^2$ and T is a positive integer. For any m and t, $f(m, t)$ is equivalent to $f_t(m)$.

Let v be the optical flow between image t and image $t+1$: A vector field representing an apparent motion related to a local gray-scale coherence between two consecutive images [1, 2, 3, 4, 5, 6]. If the local gray-scale coherence is chosen to be the (assumption of) brightness constancy, then

$$(f_t(m) - f_{t+1}(m + v))^2 = 0\ . \qquad (1)$$

In general, Eq. (1) has several solutions v since many points in an image have the same gray-scale value. Therefore, the problem of computing the motion of a point must be constrained. First, Eq. (1) can be extended to a domain surrounding the point. This implies that the motion of a point is coherent with the motion of neighboring points. Second, the motion can be restricted to a class of motion described by a limited number of parameters. Taking a parametric model approach [10, 11], the motion within a domain

Ω can be defined as follows

$$v(m) = M(m,t)\, p(\Omega),\ m \in \Omega \tag{2}$$

where M is a $2 \times n$ matrix representing a given motion model and p is a vector of $\mathcal{R}^n$ composed of the motion parameters (see Subsections 3.2.2). Combining Eqs. (1) and (2), we define the following energy

$$E(\Omega, p) = \int_\Omega (f_t(m) - f_{t+1}(m + M(m,t)\, p))^2 \, \mathrm{d}m \tag{3}$$

where Ω is a subset of D in f_t. Regarding motion (domain Ω known), Eq. (3) looks similar to the energy given in [12]. However, the work of [12] is a pixel-based approach since each pixel has its own independent motion as far as data consistency is concerned (although regularization makes the computed motion less local). On the contrary, Eq. (3) describes a global motion within domain Ω such as proposed in [9] using the absolute value of the difference instead of the squared difference (which implies a problem of differentiation when the difference is equal to zero). Moreover, since Ω is actually unknown, the problem of segmenting f_t is also considered [9, 11]. If energy (3) is iteratively and alternately minimized with respect to p and Ω, then motion computation and segmentation are coupled but not joint. In section 3, we propose a joint energy by explicitly writing p as a function of Ω.

3. JOINT MOTION SEGMENTATION / COMPUTATION ON TWO FRAMES

Equation (3) is rewritten as a joint motion segmentation / computation problem

$$E(\Omega) = \int_\Omega (f_t(m) - f_{t+1}(m + M(m,t)\, p(\Omega)))^2 \, \mathrm{d}m \tag{4}$$

where $p(\Omega)$ is given by

$$p(\Omega) = \mathrm{argmin}_p \int_\Omega (f_t(m) - f_{t+1}(m + M(m,t)\, p))^2 \, \mathrm{d}m \ . \tag{5}$$

Motion parameters $p(\Omega)$ appear as intermediate auxiliary (but valuable) variables.

3.1. Motion segmentation

3.1.1. Derivative of the energy

Minimization of energy (4) requires computation of its derivative with respect to Ω. This is not a straightforward development. Domain Ω is made dependent of a dynamic scheme variable, τ. Derivative is computed with respect to τ using the shape gradient technique [13, 14, 15]

$$\frac{\mathrm{d}E}{\mathrm{d}\tau}(\tau) = -\int_{\partial\Omega(\tau)} (f_t(s) - f_{t+1}(s + M\, p(\tau)))^2 \, V_\tau(s) N_\tau(s) \, \mathrm{d}s \ . \tag{6}$$

where $\partial\Omega$ is the oriented boundary of Ω, s is the curve parameter of $\partial\Omega(\tau)$, $p(\tau)$ is obtained by Eq. (5), $s + M\, p(\tau)$ is a shortcut for $\partial\Omega(\tau)(s) + M\, p(\tau)$, V_τ is the (unknown) local deformation (or velocity) of $\partial\Omega$, and N_τ is the inward unit normal to $\partial\Omega$.

3.1.2. Minimization of the energy

Since a dynamic scheme was introduced to determine the derivative of (4), the minimization was naturally performed using the active contour technique [16, 17, 18]. An initial contour is iteratively deformed according to V_τ chosen such that derivative (6) is negative or equal to zero at each iteration. The minimum is reached when the derivative is equal to zero. The corresponding shape of the active contour represents the segmentation. This process is described by the following evolution equation

$$\frac{\partial\Gamma}{\partial\tau}(s,\tau) = V_\tau(s) = (f_t(s) - f_{t+1}(s + M\, p(\tau)))^2 N_\tau(s) \tag{7}$$

where Γ is a short notation for $\partial\Omega$. Note that the motion parameters are required to compute the deformation at each iteration.

3.2. Motion computation

3.2.1. Definition of an auxiliary energy

For now, we consider the problem of minimizing energy (4) with respect to $p(\tau)$ at Ω fixed. The value of $p(\Omega)$ which minimizes energy (4), if the minimum exists and is unique, represents the average motion parameter of Ω between images f_t and f_{t+1} for motion model M. Note that this motion should be interpreted in a general sense (as opposed to rigid motion) since, depending on M, it might include non-linear deformation.

3.2.2. Derivative of an auxiliary energy

Minimization of energy (4) considered as a function of $p(\Omega)$ requires computation of its derivative with respect to $p(\Omega)$. It can be shown that (for simplicity, $M(m,t)$ is denoted by M)

$$\frac{\mathrm{d}E}{\mathrm{d}p}(p) = -2 \int_\Omega (f_t(m) - f_{t+1}(m + Mp)) \, M^{\mathrm{T}} \, \nabla f_{t+1}(m + Mp) \, \mathrm{d}m \tag{8}$$

where M^{T} is the transpose of matrix M. The simplest motion is translation. In this case, we have

$$M(m,t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad p = a \tag{9}$$

where a is the translation vector. Therefore derivative (8) is

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}p}(p) &= -2 \int_{\Omega} (f_t(m) - f_{t+1}(m+p)) \\ &\qquad \nabla f_{t+1}(m+p) \, \mathrm{d}m \, . \end{aligned} \tag{10}$$

3.2.3. Minimization of the auxiliary energy

Energy (4) is not globally convex in p if functions f_t and f_{t+1} are, *e.g.*, locally periodic. As a consequence, a steepest descent algorithm with successive line minimizations in the direction of the (conjugate) gradient cannot be used. However, it is considered locally convex near $p(\Omega)$. Therefore, we developed an ad hoc steepest descent algorithm with a variable step.

4. PROCESSING OF A SET OF FRAMES

Processing two frames at a time might not guarantee enough stability and accuracy to use complex motion models. The development of the previous sections is extended by writing an energy for a set of frames, taking the first frame as the reference for all the subsequent frames. If the object to be segmented is such that its boundary changes consistently with its motion then writing an energy which takes the first frame as a reference might increase the robustness of segmentation. We propose the following joint motion segmentation / computation problem

$$E(\Omega) = \sum_{t=1}^{T} \int_{\Omega} (f_0(m) - f_t(m_t(\Omega)))^2 \, \mathrm{d}m \tag{11}$$

where $p(\Omega)$ is given by

$$p(\Omega) = \operatorname{argmin}_p \sum_{t=1}^{T} \int_{\Omega} (f_0(m) - f_t(m_t(\Omega)))^2 \, \mathrm{d}m \, . \tag{12}$$

where domain Ω is the segmentation domain in the first frame and $m_t(\Omega)$ is the transformation of point m by the motion described by parameters $p(\Omega)$ repeated t times.

4.1. Motion segmentation

Minimization of energy (11) required computation of its derivative with respect to Ω. As previously mentioned, we use the shape gradient method to obtain the following expression for the derivative

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}\tau}(\tau) &= -\sum_{t=1}^{T} \int_{\partial\Omega(\tau)} (f_0(s) - f_t(m_t(\tau,s)))^2 \\ &\qquad V_\tau(s) N_\tau(s) \, \mathrm{d}s \, . \end{aligned} \tag{13}$$

Minimization of energy (11) is performed by iterating the following evolution equation

$$\frac{\partial \Gamma}{\partial \tau}(s,\tau) = V_\tau(s) = \sum_{t=1}^{T} (f_0(s) - f_t(m_t(\tau,s)))^2 N_\tau(s). \tag{14}$$

4.2. Motion computation

The parametric motion model described by Eq. (2) is convenient because it isolates the motion parameters in a vector. Thus, the derivative of energy (4) with respect to these parameters has a simple form. In order to keep this property for energy (11), the expression of transformed point $m_t(\Omega)$ must be restricted.

4.2.1. Isotropic motion model

Let us recall that matrix M is a shortcut for $M(m,t)$ (see Eq. (2) and Section 3.2.2). If M is independent of m and t (*e.g.*, for a translation), we propose to call the motion model isotropic. In this case it can be shown that

$$m_t(\Omega) = m + t \, M \, p(\Omega) \, . \tag{15}$$

Therefore, the derivative of energy (11) with respect to p is

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}p}(p) &= -2 \sum_{t=1}^{T} \int_{\Omega} (f_0(m) - f_t(m + t \, Mp)) \\ &\qquad t \, M^{\mathrm{T}} \, \nabla f_t(m + t \, Mp) \, \mathrm{d}m \, . \end{aligned} \tag{16}$$

The derivative of energy (11) with respect to Ω is straightforward.

4.2.2. Anisotropic motion model

If matrix M depends on m and/or t, we propose to call the motion model anisotropic (either in space, or time, or both). In this case, for t greater or equal to 1, we have

$$\begin{aligned} m_t(\Omega) &= m_{t-1}(\Omega) + M(m_{t-1}, t-1) \, p(\Omega) && (17) \\ &= (m_{t-2}(\Omega) + M(m_{t-2}, t-2) \, p(\Omega)) \\ &\quad + M(m_{t-1}, t-1) \, p(\Omega) && (18) \\ &= \cdots && (19) \end{aligned}$$

This expression is not readily usable. However, if the motion is restricted to a time-independent translation and scaling, then it can be shown that

$$m_t(\Omega) = k^t \, m + b \sum_{j=0}^{t-1} k^j \tag{20}$$

In this case, the derivative with respect to the motion parameters will have a relatively simple expression

$$\frac{\mathrm{d}E}{\mathrm{d}p}(p) = -2\sum_{t=1}^{T}\int_{\Omega}(f_0(m) - f_t(m_t(\Omega)))$$
$$\left(\begin{array}{cc} tk^{t-1}\,m + b\sum_{j=0}^{t-1} jk^{j-1} & \sum_{j=0}^{t-1} k^j \end{array}\right)^{\mathrm{T}} \nabla f_t(m_t)\,\mathrm{d}m..$$

5. EXPERIMENTAL RESULTS

5.1. Motion computation

In order to evaluate the accuracy of the motion computation of the proposed method, we performed tests with the "Translating Tree" sequence of reference for which motion is known. We compared our results with the study in [19]. Since our method is region-based and the study is pixel-based, we computed motion in 15x15 blocks and we assigned the motion of a block to all of its pixels. Although this approach allows a comparison with the known motion field, it probably gives our results a slight disadvantage than comparing the motion of a block with the average on the block of the known motion field. We used energy (11) with a set of 3 frames and a translation motion model. The numbers that are not in bold face in table 1 come from [19].

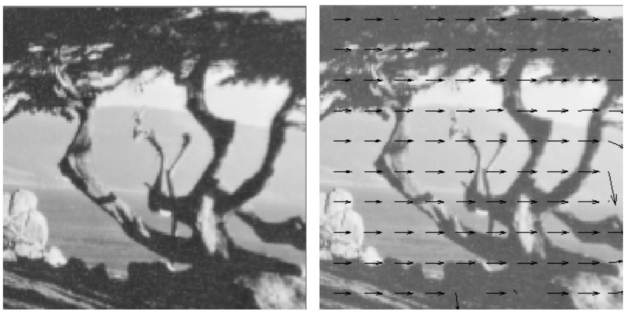

Fig. 1. On left: "translating tree" original image; On right: vectors obtained using motion estimation of a set of three frames on 15*15 blocks.

Technique	ϵ	σ	δ
Horn and Schunck (original)	38.72°	27.67°	100%
Uras *et al.* (unthresholded)	0.62°	0.52°	100%
Fleet and Jepson ($\tau = 1.25$)	0.23°	0.19°	49.7%
Proposed method	**0.33°**	**0.21°**	**81%**

Table 1. Comparison of the motion computation with the results in [19]. Notations: ϵ is the average angular error, σ is the standard deviation of the error, and δ is the density of pixels taken into account in the calculation of ϵ and σ.

Table 1 does not provide a strict comparison of the proposed method with existing methods. It only gives a hint on its accuracy. Note that the study in [19] includes much more results. We extracted the results for the original Horn and Schunck technique, for the best technique with a density higher than 75%, and for the best technique regardless of the density. As presented in [19], the error measures are not always performed on the whole image but on a subsets of the image and this rate is reported in the results as density. Figure 2 presents results on sequence "Foreman" obtained with the same energy, number of frames, and motion model as above. However, the motion was computed on 2 regions instead of blocks: The foreman's head and the background. These regions were obtained by manual segmentation and the initial guess for our steepest descent algorithm was provided by a block matching with a pixel accuracy on 16x16 blocks. For clarity, the motion of each region is represented by several, identical vectors instead of just one.

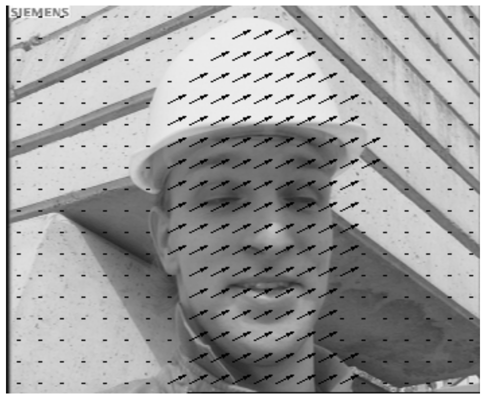

Fig. 2. Motion computation on a set of 3 frames of sequence "Foreman".

5.2. Joint motion segmentation / computation

We performed a joint motion segmentation / computation of a set of 3 frames of sequence "Coastguard". Although motion is computed using the 3 frames as described in section 5.1, only 2 frames are segmented since the different proposed energies make use of image intensity up to the last frame but they make use of the object domains up to the last but one frame. Figure 3 presents these results. Note that the front antenna (almost invisible on the images due to their small size here) had stopped the active contour in frame $t + 1$, preventing it to move closer to the other, bigger antenna behind. This is also the case in frame t but to a smaller extent.

6. CONCLUSION

The motion computation aspect of the proposed method proves to be efficient. The preliminary results on joint motion segmentation / computation are encouraging. We currently work

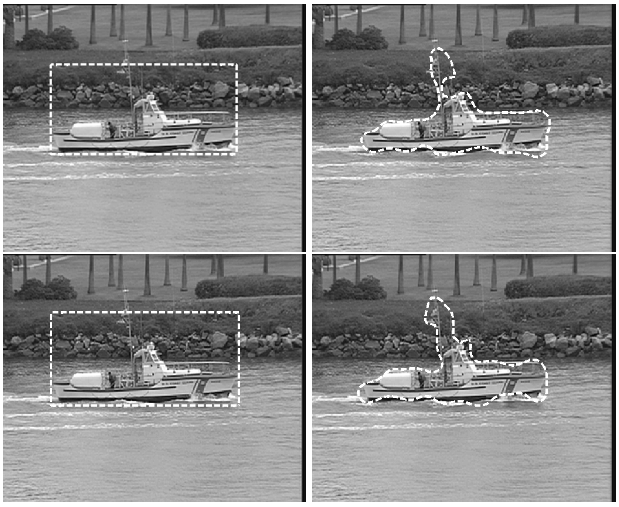

Fig. 3. Joint motion segmentation / computation of a set of 3 frames of sequence “Coastguard”; First line: User defined initial contour and segmentation result for frame t; Second line: *Idem* for frame $t + 1$.

on another criterion of temporal coherence defined successively between two frames. Future works include an extensive study on the relative benefits of the different proposed approaches: Energy based on a temporal coherence 2-by-2 or with a reference.

7. REFERENCES

[1] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[2] H. H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences,” *IEEE Tr. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 565 – 593, 1986.

[3] C. Schnörr, “Computation of discontinuous optical flow by domain decomposition and shape optimization,” *Int. Journal of Computer Vision*, vol. 8, pp. 153–165, 1992.

[4] L. Blanc-Féraud, M. Barlaud, and T. Gaidon, “Motion estimation involving discontinuities in a multiresolution scheme,” *Optical Engineering*, vol. 32, pp. 1475–1482, 1993.

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

[6] J. Weickert and C. Schnörr, “Variational optic flow computation with a spatio-temporal smoothness constraint,” *J. of Math. Imaging and Vision*, vol. 14, pp. 245–255, 2001.

[7] R. Li, B. Zeng, and M. L. Liou, “A new threestep search algorithm for block estimation,” *IEEE Transactions on Circuits Systems and Video Technology*, vol. 4, pp. 438–442, 1994.

[8] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” *IEEE Transactions on Image Processing*, vol. 9, pp. 287–290, 2000.

[9] S. F. Wu and J. Kittler, “A gradient-based method for general motion estimation and segmentation,” *J. of Visual Communication and Image Representation*, vol. 4, pp. 25–38, 1993.

[10] J.-M. Odobez and P. Bouthemy, “Robust multiresolution estimation of parametric motion models,” *J. of Visual Communication and Image Representation*, vol. 6, pp. 348–365, 1995.

[11] D. Cremers and S. Soatto, “Variational space-time motion segmentation,” in *International Conference on Computer Vision*, 2003, pp. 886–893.

[12] L. Alvarez, J. Weickert, and J. Sánchez, “A scale-space approach to nonlocal optical flow calculations,” in *International Conference on Scale-Space Theories in Computer Vision*, 1999, pp. 235–246.

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

[14] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, “Image segmentation using active contours: Calculus of variations or shape gradients?,” *SIAM Journal of Applied Mathematics*, vol. 63, pp. 2128–2154, 2003.

[15] 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,” *International Journal of Computer Vision*, vol. 53, pp. 45–70, 2003.

[16] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, pp. 321–332, 1988.

[17] V. Caselles, F. Catté, T. Coll, and F. Dibos, “A geometric model for active contours,” *Nümerische Mathematik*, vol. 66, pp. 1–31, 1993.

[18] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *Int. J. of Computer Vision*, vol. 22, pp. 61–79, 1997.

[19] J. L. Barron, D. J. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” *International Journal of Computer Vision*, vol. 12, pp. 43–77, 1994.