Vector Field Segmentation Using Active Contours: Regions Of Vectors With The Same Direction

[&]Tristan Roy [&]Michel Barlaud [&]Éric Debreuve [+] Gilles Aubert

[&]Laboratoire I3S, UMR CNRS 6070
Les Algorithmes, Bât. Euclide B
2000, route des lucioles
06903 Sophia Antipolis Cedex, France
e-mail: {troy, barlaud, debreuve}@i3s.unice.fr

[+]Laboratoire Dieudonné, UMR CNRS 6621
Université de Nice-Sophia Antipolis
Parc Valrose
06108 Nice Cedex 2, France
e-mail: gaubert@math.unice.fr

Abstract

We developed a method to segment a vector field and retrieve parameters of the segmented region simultaneoulsy with a vector homogeneity defined by an identical direction criterion. The method was applied to motion segmentation for objects in translation. The proposed criterion is based on the notion of dominant direction of a set of vectors. It states that every vector within a region must have a direction as close as possible to the dominant direction of the region. A maximal area constraint was added since the criterion is minimized on any subset of a valid segmentation. Differentiation of the criterion in a dynamic scheme framework led to a region-based active contour segmentation method. The dominant direction of the active region represents the parameters of the region. It is continuously recomputed as the region evolves while the region evolution depends on its value. The algorithm was successfully applied to a synthetic sequence and is being applied to real data.

1 Introduction

Vector field segmentation can be applied to optical flow segmentation. Actually optical flow calculation and segmentation can be performed simultaneously [12] by modifying the optical flow equation [6]: The authors represented the edges between two regions with a different motion by a contour. Then they use an active contour technique to segment and compute simultaneously the optical flow. This segmentation does not use any motion model. In order to be more robust a motion model can be introduced. In [4] the authors proposed a method to segment motion using a piecewise constant motion model and a piecewise affine motion model. Their modelization can be interpreted as an extension of the Mumford-Shah model [8] where the image intensity homogeneity criterion is replaced with a motion homogeneity according to a particular model. The piecewise constant model allows to segment translating objects. The piecewise affine model allows to segment rotating objects.

Generally motion segmentation algorithms often follow the same methodology: *(i)* Given an estimation of motion parameters, segmentation is performed based on static analysis; *(ii)* Then given an estimation of the segmentation, motion parameters are estimated according to a motion model; *(iii)* Steps *(i)* and *(ii)* are alternated until convergence. The Expectation-Maximization (EM) algorithm fits in this description. A modified version of the classical EM method was developed by [14] for motion segmentation. It uses a so-called perceptual organization of data for static analysis in step *(i)*.

We propose a vector field segmentation method using a vector field model based on the notion of dominant direction of a set of vectors [11]. It allows to segment regions containing vectors with the same direction (translating objects). It was developed in an energy minimization framework using active contours and shape optimization theory. Segmentation and estimation of the vector field model parameters (the dominant direction) are performed simultaneously.

2 Dominant direction of a set of vectors

Let us consider a set of n non-zero vectors, $V_1, V_2, \ldots, V_n$, of $\mathbb{R}^2$. Let us find a vector, v, corresponding to the dominant direction among the n vectors. The following function was introduced in [11]:

$$\begin{array}{rccl} J: & \mathbf{B} & \rightarrow & \mathbb{R} \\ & u & \mapsto & \sum_{i=1}^{n}(u \cdot V_i)^2 = u^t \sum_{i=1}^{n}(V_i V_i^t)u \end{array} \quad (1)$$

where $\mathbf{B}$ is the unit sphere in $\mathbb{R}^2$ ($\{u \in \mathbb{R}^2, |u| = 1\}$), $a \cdot b$ is the inner product, and u^t is the transpose of u. It must be assumed that the following symmetric, positive, 2×2 matrix

$$M = \sum_{i=1}^{n} V_i V_i^t \quad (2)$$

has two distinct eigenvalues. Function J is maximized by v and $-v$ where v is a normalized eigenvector corresponding to the highest eigenvalue of M.

Now let us consider a vector field

$$\begin{array}{rcl} V: \quad \mathbf{D} & \rightarrow & \mathbb{R}^2 \\ m & \mapsto & V(m) = \left[\begin{array}{c} V_1(m) \\ V_2(m) \end{array} \right] \end{array} \tag{3}$$

where $\mathbf{D}$ is a bounded domain of $\mathbb{R}^2$. We defined the dominant direction, v, of vector field V within domain Ω ($\Omega \subset \mathbb{R}^2$, bounded) as a normalized eigenvector corresponding to the highest eigenvalue of the following (symmetric, positive, 2×2) matrix

$$M(\Omega) = \int_\Omega V(m)V^t(m)dm \tag{4}$$

It must be assumed that $M(\Omega)$ has two distinct eigenvalues. By extension v is called the dominant direction of Ω.

3 Energy of a vector field within a domain

Let us consider a vector field, V, such that for all m in $\mathbf{D}$, $V(m)$ is different from zero. Let us consider all domains Ω included in $\mathbf{D}$ verifying the following property: $M(\Omega)$ has two distinct eigenvalues. We look for domains containing vectors with the same direction. Similarly we can say that vectors contained by such a domain must have the same direction as the dominant direction of the domain. We defined the energy of domain Ω, or equivalently the energy of its boundary $\partial\Omega$, as follows:

$$J_1(\Omega) = \int_\Omega |w(m) \times w_{\mathrm{dom}}(\Omega)|^2 \, dm \tag{5}$$

where $a \times b$ is the cross product, $w(m)$ is equal to $V(m)$ divided by its norm, and $w_{\mathrm{dom}}(\Omega)$ is the dominant direction of Ω. If Ω contains vectors with the same direction, any vector w within Ω can represent the dominant direction and J_1 is equal to zero. Note that in this case any subset of Ω has an energy equal to zero. Since our purpose was to find the largest domain, Ω_{hom}, containing vectors with the same direction (see Fig. 1), we added a maximal area constraint

$$\begin{aligned} J(\Omega) \quad = \quad & \int_\Omega |w(m) \times w_{\mathrm{dom}}(\Omega)|^2 \, dm \\ & + \lambda \int_{\mathbf{D}/\Omega} dm \end{aligned} \tag{6}$$

where λ is a real number and $\mathbf{D}/\Omega$ is the region outside Ω included in $\mathbf{D}$. Therefore minimizing J should allow to find a domain as large as possible containing vectors with directions as close as possible with each other. Eq. (6) can be rewritten as

$$\begin{aligned} J(\Omega) \quad = \quad & \int_\Omega \sin^2(\theta(m,\Omega))dm \\ & + \lambda \int_{\mathbf{D}/\Omega} dm \end{aligned} \tag{7}$$

where (see Fig. 2)

$$\begin{aligned} \theta(m,\Omega) &= \widehat{w(m), w_{\mathrm{dom}}}(\Omega) && (8) \\ &= \theta_2(\Omega) - \theta_1(m) && (9) \\ \theta_1(m) &= \widehat{e_1, w(m)} && (10) \\ \theta_2(\Omega) &= \widehat{e_1, w_{\mathrm{dom}}}(\Omega) && (11) \end{aligned}$$

where $\widehat{a,b}$ is the oriented angle between vector a and vector b, and (e_1, e_2) is the canonical basis of $\mathbb{R}^2$. Note that expression (7) is correct only because there are no zero vectors.

4 Energy minimization

Minimization of energy (7) can be done using a steepest descent algorithm. In this case computation of the energy derivative is required. However this energy derivative with respect to domain Ω cannot be determined directly because the set of domains included in $\mathbf{D}$ does not have a structure of vectorial space. Therefore the differentiation with respect to Ω is replaced with a differentiation with respect to variable τ where τ is the second argument of the following function

$$\begin{array}{rcl} T: \quad \mathbb{R}^2 \times \mathbb{R}^+ & \rightarrow & \mathbb{R}^2 \\ (m,\tau) & \mapsto & T(m,\tau) \end{array} \tag{12}$$

with the following properties

$$\begin{aligned} & T(\Omega_0, 0) = \Omega_0 && (13) \\ & T(\Omega_0, \tau) = \Omega(\tau) && (14) \\ & T_\tau(m) \text{ is a } C^1\text{-diffeomorphism} && (15) \\ & T_m(\tau) \text{ is a } C^1\text{-function on } \mathbb{R}^+ && (16) \end{aligned}$$

where Ω_0 is an initial domain with an energy usually not minimal. More information on that differentiation framework can be found in [5, 7, 13]. In particular the following proposition was proven:

Proposition 1 Let us consider the following function

$$F(\tau) = \int_{\Omega(\tau)} k(m, \Omega(\tau))dm \tag{17}$$

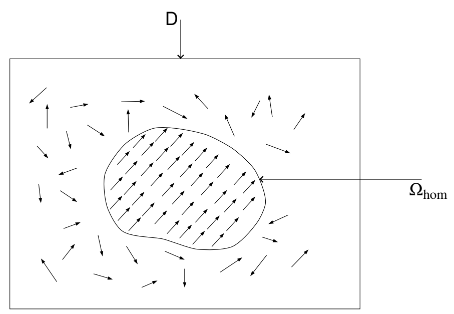


Figure 1: Domain containing vectors with the same direction.

where $\Omega(\tau)$ is equal to $T(\Omega_0, \tau)$ with Ω_0 being a domain included in $\mathbf{D}$ and k is a smooth function. We have

$$\begin{aligned} F'(\tau) &= \int_{\Omega(\tau)} \frac{\partial k}{\partial \tau}(m, \Omega(\tau)) dm \\ & \quad - \int_{\partial\Omega(\tau)} k(x_1(s), x_2(s), \Omega(\tau))(v \cdot N) ds \end{aligned} \tag{18}$$

where F' is the shape derivative of F, $\frac{\partial k}{\partial \tau}$ is the shape derivative of k with x_1 and x_2 fixed, s is the curvilinear abscissa on $\partial\Omega(\tau)$, v is the velocity of $\partial\Omega(\tau)$ (v is a short notation for $v(m, \Omega(\tau))$), and N is the inward unit normal to $\partial\Omega(\tau)$ (N is a short notation for $N(m, \Omega(\tau))$). The mathematical definitions of the shape derivatives are as follows

$$F'(\tau) = \lim_{d\tau \to 0} \frac{F(\Omega(\tau) + v d\tau) - F(\Omega(\tau))}{d\tau} \tag{19}$$

$$\frac{\partial k}{\partial \tau}(m, \Omega(\tau)) = \lim_{d\tau \to 0} \frac{k(m, \Omega(\tau) + v d\tau) - k(m, \Omega(\tau))}{d\tau} \tag{20}$$

$$v(m, \Omega(\tau)) = \frac{\partial T}{\partial \tau}(m, \tau) \tag{21}$$

Applying this result to our minimization problem, the purpose is to choose v in order to ensure the fastest decrease of J.

4.1 Energy derivative

We used Eq. (18) to compute the derivative of (7). First Eq. (7) is rewritten using variable τ.

$$\begin{aligned} J(\tau) &= \int_{\Omega(\tau)} \sin^2(\theta(m, \Omega(\tau))) dm \\ & \quad + \lambda \int_{\mathbf{D}/\Omega(\tau)} dm \end{aligned} \tag{22}$$

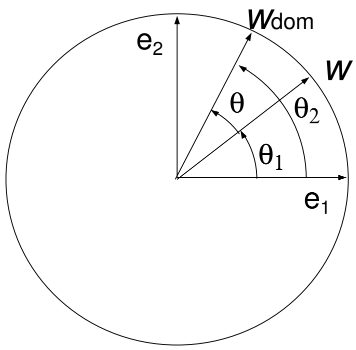


Figure 2: Angles θ, θ_1, and θ_2 on the unit circle.

Then the result of proposition 1 is applied:

$$\begin{aligned} J'(\tau) &= \int_{\Omega(\tau)} 2 \sin\theta \cos\theta \frac{\partial \theta}{\partial \tau} dm \\ & \quad - \int_{\partial\Omega(\tau)} \sin^2\theta (v \cdot N) ds \\ & \quad + \lambda \int_{\partial\Omega(\tau)} (v \cdot N) ds \end{aligned} \tag{23}$$

$$\begin{aligned} &= \int_{\Omega(\tau)} \sin 2\theta \frac{\partial \theta}{\partial \tau} dm \\ & \quad - \int_{\partial\Omega(\tau)} \sin^2\theta (v \cdot N) ds \\ & \quad + \lambda \int_{\partial\Omega(\tau)} (v \cdot N) ds \end{aligned} \tag{24}$$

where θ is a short notation for $\theta(m, \Omega(\tau))$.

4.1.1 Computation of $\frac{\partial \theta}{\partial \tau}$

The purpose is to express $\frac{\partial \theta}{\partial \tau}$ as a function of v.

$$\frac{\partial \theta}{\partial \tau} = \frac{\partial(\theta_2 - \theta_1)}{\partial \tau} \tag{25}$$

$$= \frac{\partial \theta_2}{\partial \tau} - \frac{\partial \theta_1}{\partial \tau} \tag{26}$$

$$= \frac{\partial \theta_2}{\partial \tau} \tag{27}$$

since θ_1 does not depend on τ. Furthermore we have (by definition)

$$w_{\text{dom}} = \begin{bmatrix} \cos\theta_2 \\ \sin\theta_2 \end{bmatrix} \tag{28}$$

Therefore

$$w'_{\text{dom}} = \frac{\partial w_{\text{dom}}}{\partial \tau} = \frac{\partial \theta_2}{\partial \tau} \vec{t} \tag{29}$$

where $\vec{t}$ is the vector tangent to the unit, counterclockwise-oriented circle:

$$\vec{t} = \begin{bmatrix} -\sin\theta_2 \\ \cos\theta_2 \end{bmatrix} \tag{30}$$

Taking the inner product of (29) with $\vec{t}$ leads to

$$\frac{\partial \theta_2}{\partial \tau} = w'_{\text{dom}} \cdot \begin{bmatrix} -\sin\theta_2 \\ \cos\theta_2 \end{bmatrix} \tag{31}$$

Let us recall the dominant direction matrix (using the normalized vector field, w, instead of the original vector field, V)

$$M(\tau) = \int_{\Omega(\tau)} w(m) w^t(m) dm \tag{32}$$

where w is

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{33}$$

Therefore

$$M(\tau) = \begin{bmatrix} a & c \\ c & b \end{bmatrix} \tag{34}$$

where

$$a = \int_{\Omega(\tau)} w_1^2 dm \quad (35)$$

$$b = \int_{\Omega(\tau)} w_2^2 dm \quad (36)$$

$$c = \int_{\Omega(\tau)} w_1 w_2 dm \quad (37)$$

According to section 2, a possible choice for w_{dom} is (the other choice being the opposite vector)

$$w_{\mathrm{dom}} = \frac{1}{\sqrt{(\lambda_{\max} - a)^2 + c^2}} \begin{bmatrix} c \\ \lambda_{\max} - a \end{bmatrix} \quad (38)$$

where $\lambda_{\max}$ is the highest eigenvalue of M and is equal to

$$\lambda_{\max} = \frac{a + b + \sqrt{(a-b)^2 + 4c^2}}{2} \quad (39)$$

Note that if $\lambda_{\max}$ is equal to a and c is equal to zero then expression (38) is not valid. In this case the following expression is used

$$w_{\mathrm{dom}} = \frac{1}{\sqrt{(\lambda_{\max} - b)^2 + c^2}} \begin{bmatrix} \lambda_{\max} - b \\ c \end{bmatrix} \quad (40)$$

Since we assumed that matrix M has two distinct eigenvalues it is not possible to have $\lambda_{\max}$ equal to a and equal to b at the same time. Therefore there is always a valid expression for w_{dom}.

Proposition 2 There exists $\alpha(m, \Omega(\tau))$ and $\beta(m, \Omega(\tau))$, two real numbers, such that

$$w'_{\mathrm{dom}} = \begin{bmatrix} \int_{\partial\Omega(\tau)} \alpha(v \cdot N) ds \\ \int_{\partial\Omega(\tau)} \beta(v \cdot N) ds \end{bmatrix} \quad (41)$$

A proof can be found in the appendix.

Using proposition 2 Eq. (31) can be rewritten as

$$\begin{aligned} \frac{\partial \theta_2}{\partial \tau} &= -\sin\theta_2 \int_{\partial\Omega(\tau)} \alpha(v \cdot N) ds \\ &\quad + \cos\theta_2 \int_{\partial\Omega(\tau)} \beta(v \cdot N) ds \end{aligned} \quad (42)$$

4.1.2 Expression of J' as a function of v

By combining (24), (27), and (42), J' can be expressed as a function of v

$$\begin{aligned} J'(\tau) &= \int_{\Omega(\tau)} \sin 2\theta \Bigg(-\sin\theta_2 \int_{\partial\Omega(\tau)} \alpha(v \cdot N) ds \\ &\quad + \cos\theta_2 \int_{\partial\Omega(\tau)} \beta(v \cdot N) ds \Bigg) dm \\ &\quad - \int_{\partial\Omega(\tau)} \sin^2\theta (v \cdot N) ds \\ &\quad + \int_{\partial\Omega(\tau)} \lambda (v \cdot N) ds \qquad (43) \\ &= \int_{\partial\Omega(\tau)} (-\alpha \sin\theta_2 + \beta\cos\theta_2)(v \cdot N) ds \\ &\quad \times \int_{\Omega(\tau)} \sin 2\theta dm - \int_{\partial\Omega(\tau)} \sin^2\theta (v \cdot N) ds \\ &\quad + \int_{\partial\Omega(\tau)} \lambda (v \cdot N) ds \qquad (44) \\ &= \int_{\partial\Omega(\tau)} v \cdot \Bigg[\left(\int_{\Omega(\tau)} \sin 2\theta dm \right) \\ &\quad (-\alpha\sin\theta_2 + \beta\cos\theta_2) - \sin^2\theta + \lambda \Bigg] N ds \end{aligned}$$

(45)

5 Evolution equation

The purpose being to minimize J velocity v must be chosen such that the energy decreases with maximum efficiency (J' in (45) must be negative with the highest possible absolute value). According to the Cauchy-Schwarz inequality this value is

$$\begin{aligned} v &= \Bigg[\left(\int_{\Omega(\tau)} \sin 2\theta dm \right) (\alpha\sin\theta_2 - \beta\cos\theta_2) \\ &\quad + \sin^2\theta - \lambda \Bigg] N = \mu N \end{aligned} \quad (46)$$

where α and β depend on vector field w and domain $\Omega(\tau)$ (Their expressions are not given here because they are quite complicated and hard to interpret. They can be deduced from Eq. (35), (36), (37), and the equations of the Appendix). Therefore we have the following evolution equation

$$\frac{\partial \Gamma}{\partial \tau} = v = \mu N \quad (47)$$

where $\Gamma(\tau)$ is a short notation for $\partial\Omega(\tau)$. Eq. (47) is the active contour ([2, 3, 9, 10]) evolution equation of the proposed segmentation method which segmentation criterion is the homogeneity of a vector field in term of direction. Given a value of parameter λ, the segmentation algorithm alternately solves for the contour (Eq. (47)) and for the dominant direction within the contour (Eq. (38) or (40)) until mutual convergence.

6 Application to the segmentation of a translating object

6.1 Context

In a greyscale video sequence $(I_t, t \geq 0)$ let us suppose that an object has a motion of translation. Let us suppose that the optical flow was estimated

$$\begin{array}{rccl} V_{t\to t+1}: & \mathbf{D} & \to & \mathbb{R}^2 \\ & m & \mapsto & \left[\begin{array}{c} V_{t\to t+1,1}(m) \\ V_{t\to t+1,2}(m) \end{array} \right] \end{array} \tag{48}$$

We assume that $V_{t\to t+1}(m)$ is a non-zero vector for all m and all t. Therefore the object domain should minimize criterion (7) using the proposed active contour method. Note that the initial contour must include an area containing vectors with the same direction wide enough with respect to the contour area so that the dominant direction is meaningful.

6.2 Optical flow estimation

In the following examples the optical flow was estimated by a block matching method with a block size of 4×4 pixels and a maximum displacement of 5 pixels in each direction. The matching criterion was the zero-mean normalized sum of squared differences (ZNSSD).

6.3 Implementation

The active contour evolution was implemented using a smoothing B-splines contour representation [1].

6.4 Results

6.4.1 Translation on a still background

A face was translated by two pixels in each direction on a still, noisy background. Since the vector field on the background is composed of zero vectors, the proposed method cannot be applied directly. Therefore the zero vectors were first replaced with random vectors. Figure 3 shows the two images of the sequence. Figure 4 shows the optical flow computed by block matching. Figure 5 shows the segmentation result with λ equal to 0.01.

6.4.2 Translation on a translating background

The sequence is the same as the sequence of section 6.4.1 except that the background was also translated by two pixels in the horizontal direction. Figure 6 shows the two images of the sequence. Figure 7 shows the optical flow computed by block matching. Figure 8 shows the segmentation result with λ equal to 0.1.

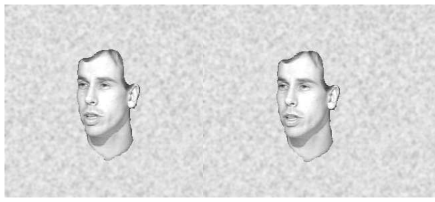


Figure 3: The two images of the sequence with a still background

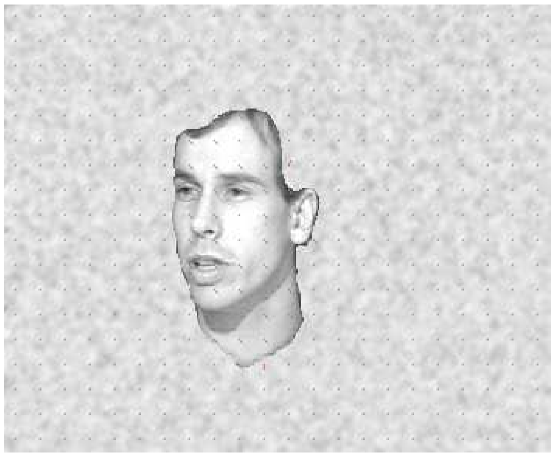


Figure 4: Optical flow (still background)

6.4.3 Sponge translation on a still background

Figure 9 shows the two images of the sequence. Figure 10 shows the optical flow computed by block matching. Figure 11 shows the segmentation result with λ equal to 0.05.

Initial contour Intermediate contour Final contour

Figure 5: From initial contour to final contour (still background)

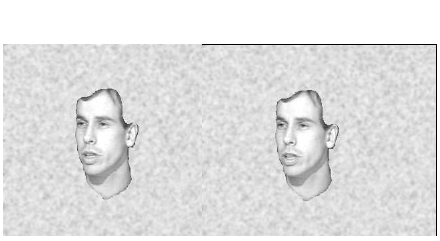


Figure 6: The two images of the sequence with a translating background

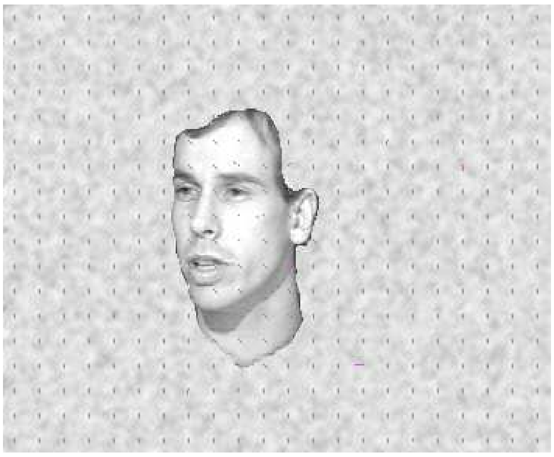
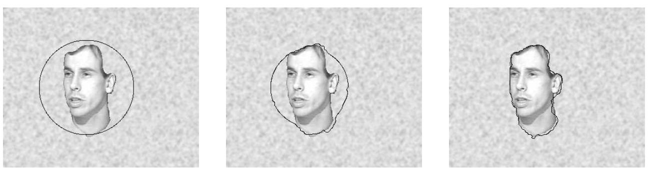


Figure 7: Optical flow (translating background)

6.5 Discussion

Both the dominant direction computation and our energy definition involve a normalized vector field. This has two disadvantages: *(i)* A vector with a small norm having a random direction within a coherent region (a “noise” vector) has as much influence on the dominant direction computation as a vector with a large norm having a coherent direction; *(ii)* It is not possible to make the distinction between a translating object and the background if the background is translating in the same direction as the object (even if the translations have very different amplitudes).
Another limitation of the proposed method is that it depends



Initial contour Intermediate contour Final contour

Figure 8: From initial contour to final contour (translating background)

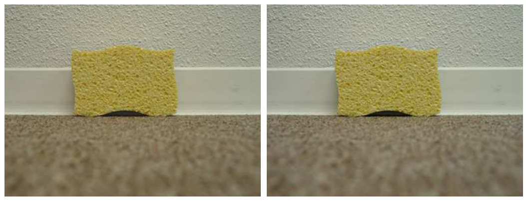


Figure 9: The two images of the sequence

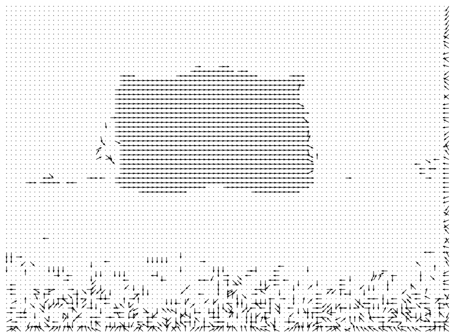


Figure 10: Optical flow

on an optical flow estimation. Therefore an homogeneous object (the optical flow inside the object is equal to zero) cannot be segmented.
Simultaneous segmentation of several translating objects is possible even if their translations have different directions. However there must be one active contour per direction (therefore it is equivalent to segmenting each object one after the other) because the common dominant direction of two objects with different translations do not make sense.

Acknowledgements

The optical flow estimations were provided by Muriel Gastaud and Thomas André, and the smoothing B-spline implementation was done by Frédéric Precioso, Laboratoire I3S.

Initial contour Intermediate contour Final contour

Figure 11: From initial contour to final contour

Appendix

The purpose is to prove proposition 2.
We use (18) and (35), (36), (37) to compute

$$a' = -\int_{\partial\Omega(\tau)} w_1^2 (v \cdot N) ds \tag{49}$$

$$b' = -\int_{\partial\Omega(\tau)} w_2^2 (v \cdot N) ds \tag{50}$$

$$c' = -\int_{\partial\Omega(\tau)} w_1 w_2 (v \cdot N) ds \tag{51}$$

Then (38) is equal to (a similar calculation can be done using (40))

$$w'_{\mathrm{dom}} = \begin{bmatrix} -\frac{1}{2}\left[(\lambda_{\max}-a)^2+c^2\right]^{-\frac{3}{2}} \times \\ 2\left[(\lambda_{\max}-a)(\lambda_{\max}-a)'+cc'\right]c \\ +c'\left[(\lambda_{\max}-a)^2+c^2\right]^{-\frac{1}{2}} \\ \\ -\frac{1}{2}\left[(\lambda_{\max}-a)^2+c^2\right]^{-\frac{3}{2}} \times \\ 2\left[(\lambda_{\max}-a)(\lambda_{\max}-a)'+cc'\right](\lambda_{\max}-a) \\ +(\lambda_{\max}-a)'\left[(\lambda_{\max}-a)^2+c^2\right]^{-\frac{1}{2}} \end{bmatrix} \tag{52}$$

Using (39) we have

$$\lambda_{\max} - a = \frac{b-a}{2} + \sqrt{(\frac{b-a}{2})^2 + c^2} \tag{53}$$

Differentiating (53) using (49), (50), and (51) we obtain

$$\begin{aligned}(\lambda_{\max}-a)' &= \int_{\partial\Omega(\tau)} (\frac{w_1^2 - w_2^2}{2})(v\cdot N)ds \\ &+ \frac{1}{\sqrt{(\frac{b-a}{2})^2+c^2}}\left[\frac{b-a}{2}\frac{b'-a'}{2}+cc'\right]\end{aligned} \tag{54}$$

which can be rewritten as

$$\begin{aligned}(\lambda_{\max}-a)' &= \int_{\partial\Omega(\tau)} \Bigg(\left(\frac{w_1^2-w_2^2}{2}\right) \\ &\left(1+\frac{\frac{b-a}{2}}{\sqrt{(\frac{b-a}{2})^2+c^2}}\right) \\ &-\frac{c}{\sqrt{(\frac{b-a}{2})^2+c^2}} w_1 w_2 \Bigg)(v\cdot N)ds\end{aligned} \tag{55}$$

Combining (51), (52), and (55) we can conclude that there exists $\alpha(m, \partial\Omega(\tau))$ and $\beta(m, \partial\Omega(\tau))$ such that

$$w'_{\mathrm{dom}} = \begin{bmatrix} \int_{\partial\Omega(\tau)} \alpha(v\cdot N)ds \\ \int_{\partial\Omega(\tau)} \beta(v\cdot N)ds \end{bmatrix} \tag{56}$$

References

[1] P. Brigger, J. Hoeg, and M. Unser. B-Spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496, September 2000.

[2] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Nümerische Mathematik*, 66:1–31, 1993.

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

[4] D. Cremers and C. Schnörr. Motion competition: Variational integration of motion segmentation and shape regularization. In *Pattern Recognition*, 2002.

[5] M.C. Delfour and J.-P. Zolésio. *Shapes and geometries*. Advances in Design and Control. Siam, 2001.

[6] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] 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*, 53(1):45–70, 2003.

[8] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.

[9] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[10] N. Paragios and R. Deriche. Geodesic active regions for motion estimation and tracking. In *International Conference on Computer Vision*, 1999.

[11] A. Ravishankar Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical models and image processing*, 53(2):157–185, 1991.

[12] C. Schnörr. Computation of discontinuous optical flow by domain decomposition and shape optimization. *International Journal of Computer Vision*, 8(2):153–165, 1992.

[13] J. Sokolowski and J.-P. Zolésio. *Introduction to shape optimization. Shape sensitivity analysis.*, volume 16 of *Springer Ser. Comput. Math.* Springer-Verlag, Berlin, 1992.

[14] Y. Weiss and E. H. Adelson. Perceptually organized em: A framework for motion segmentation that combines information about form and motion. *MIT Media Laboratory, Perceptual Computing Section, Technical Report 315*, 1994.