Snakes in Movement

Author(s): V. Caselles and B. Coll

Source: *SIAM Journal on Numerical Analysis*, Vol. 33, No. 6 (Dec., 1996), pp. 2445-2456

Published by: Society for Industrial and Applied Mathematics

Stable URL: https://www.jstor.org/stable/2158479

Accessed: 08-10-2021 00:35 UTC

SIAM J. NUMER. ANAL.
Vol. 33, No. 6, pp. 2445–2456, December 1996

SNAKES IN MOVEMENT*

V. CASELLES† AND B. COLL†

Abstract. In this paper, we propose a geometric partial differential equation (PDE) for tracking one or several moving objects from a sequence of images, which is based on a geometric model for active contours. The active contour approach permits us to simultaneously handle both aspects: finding the boundaries and tracking them. We also describe a numerical scheme to solve the geometric equation and we present some numerical experiments.

Key words. image analysis, motion tracking, active contours, optical flow, geometric equation, viscosity solutions

AMS subject classifications. 68U10, 65C20

1. Introduction.

One of the aims in the study of the motion from a sequence of images is to follow several moving objects and to determine the successive positions of its boundaries. Both aspects, finding the boundaries and tracking them, can be handled simultaneously with an active contour approach. This is, indeed, the strategy followed by Leymarie and Levine [16] when trying to follow the motion and deformations of a population of living cells (see also [3]). In this paper we propose a different model for tracking several moving objects in the plane based on a geometric model for active contours (also called "snakes") introduced in [4] and independently in [17]. In [4], Caselles, Catte, Coll, and Dibos propose a partial differential equation (PDE) model for active contours based on the equation

$$\frac{\partial u}{\partial t} = g(x)|\nabla u| \left(\operatorname{div}\left(\frac{\nabla u}{| \nabla u |} \right) + \nu \right) \qquad u(0,x) = u_0(x), \quad x \in R^2, \tag{1.1}$$

where $g(x) = \frac{1}{1+(\nabla G_\sigma * I)^2}$, ν is a positive real constant, $G_\sigma * I$ is the convolution of the image I, where we are looking for the contour of an object O, with the Gaussian $G_\sigma = c\sigma^{1/2}\exp^{(-\frac{|x|^2}{4\sigma})}$ and u_0 is the initial data which is taken as a smoothed version of the function $1 - \chi_C$, where χ_C is the characteristic function of a set C containing the object O. In this model, the object that one is looking for is contained in the zero level set of $u_0(x)$, i.e., $\{x : u_0(x) = 0\}$, and one is interested in following the boundary of the zero level set of u:

$$\partial c(t) = \partial\{x : u(t,x) = 0\}.$$

Following [4], let us recall the geometric interpretation of this model. Recall that if $u(t,x)$ is a smooth function, k any real number, and

$$\partial c(t) = \{x : u(t,x) = k\}$$

is a smooth curve in R^2, then the curvature at a point x on the curve $\partial c(t)$ is given by $\operatorname{curv}(x) = \operatorname{div}(\frac{\nabla u}{|\nabla u|})$.

Let us interpret geometrically the model (1.1).

(1) The term $|\nabla u|\operatorname{div}(\frac{\nabla u}{|\nabla u|})$ ensures that the grey level set at a point in ∂c increases proportionally to the algebraic curvature of ∂c at this point. The factor $|\nabla u|$ ensures that the function u hardly changes its grey level except on a neighborhood of ∂c. This term is responsible for the regularizing effect of the model. The constant ν is a correction term chosen so that $|\nabla u|(\operatorname{div}(\frac{\nabla u}{|\nabla u|}) + \nu)$ remains always positive. Hence, the grey level at a point at ∂c will increase (going from black to grey) and one expects that ∂c reproduces the boundary of the object O.

*Received by the editors September 30, 1994; accepted for publication (in revised form) February 14, 1995. This research was partially supported by EC project Mathematical Methods for Image Processing, reference ERBCHRXCT930095.

†Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears, 07071 Palma (Balears), Spain (dmivca0@ps.uib.es, dmitcv0@ps.uib.es).

The constant ν may be interpreted as a force pushing ∂c towards O when the curvature of ∂c becomes null or negative.

(2) The term $g(x)$ controls the speed at which ∂c moves. When ∂c is near the boundary of the object O, $|\nabla G_\sigma * I|$ is large and ∂c stops. The image I is convolved to eliminate the effect of noise on the motion of ∂c. This coefficient slows down the growth of the function u near the boundary of the object O and it stops exactly on the boundary of it, if this boundary is a regular curve.

As was proved in [4], model (1.1) is a well-posed problem from the mathematical point of view; i.e., one has existence and uniqueness of solutions, is consistent with the purpose of finding contours, at least, in the ideal case where no noise is present in the image, and it permits one to design a stable numerical algorithm with a minimum set of adjustment parameters [4], [21]. More information about these issues and the comparison with the classical model for active contours of Kass, Witkin, and Terzopoulos [15], [6] based on the minimization of an energy functional can be seen in [4], [17].

Then, based on model (1.1), we propose a new model to follow, in principle, several moving objects in a sequence of images. Let us briefly explain the principles of the method. At the initial stage, we use the geometric model (1.1) described above to detect on the first image the contours we want to follow. The next step consists of computing a good initial condition for the next image and then again using model (1.1) proceeding iteratively in this way. We shall be able to give a good initialization at each stage if we are able to find a sufficiently good estimation for the velocity field of the motion, at least, on the neighborhood of the moving objects.

Before going into the details, let us explain the plan of the paper. In §2 we describe our model and some existence results. In §3 we review several strategies used to compute the velocity vector field and to describe the one we shall follow below. Finally, in §4, we discuss the numerical algorithm and present the experimental results obtained with our PDE model of §2.

2. The model. We propose a new model to follow the successive positions of the boundaries of moving objects in a sequence of images $I(t,x)$ based on (1.1). To write the model as a PDE, we suppose that the sequence $I(t,x)$ is parameterized by a continuous time t. The idea is to add a term to (1.1) which gives the displacement of the objects from one image to the next. Hence, the resulting model is given by

$$\begin{aligned} &\frac{\partial u}{\partial t} = g(t,x) \mid \nabla u \mid \operatorname{div}\left(\frac{\nabla u}{\mid \nabla u \mid} + \nu\right) + (1 - g(t,x))v.\nabla u, \quad t > 0, x \in R^2, \\ &u(0,x) = u_0(x), \quad x \in R^2, \end{aligned} \tag{2.1}$$

where $v = (v_1, v_2)$ is the velocity vector field (more precisely, the apparent velocity (see §3 below)) and $g(t,x)$ is a smooth function such that $g(t,x) = 1$ if $|G_\sigma * \nabla I(t,x)| < k - \epsilon$ and $g(t,x) = 0$ if $|G_\sigma * \nabla I(t,x)| > k + \epsilon$, where k is a given threshold and $\epsilon > 0$ is supposed to be small. As in the snakes model (1.1), the initial condition $u_0(x)$ is taken as a smoothed version of the function $1 - \chi_C$, where χ_C is the characteristic function of a set C containing the objects O_i we want to follow. As in (1.1), our active contour will be the boundary of the level set of the solution u of (2.1). The first term of the right-hand side above has the same interpretation as the right-hand side of model (1.1) already explained in the introduction. The second term of the right-hand side of (2.1) tries to keep the estimated contour or snake near to the boundary of the moving object. Indeed, that effect would be produced by the term $v.\nabla u$. The term $(1 - g(t,x))$ multiplying it takes the value zero outside of a neighborhood of the boundaries of the moving objects and values near to one at the boundaries of those objects. On one hand, this is of no harm since we are only interested in following that boundary. On the other hand, this permits us to estimate the vector field v only in a neighborhood of those boundaries which is, in fact, the only thing we are able to do. The concrete computational details will be explained in §4.

As in the geometric model (1.1), model (2.1) permits us to treat the case of several moving objects. Moreover, as in the static case, topological changes are possible which give us the possibility of analyzing occlusions, at least, in some cases.

Model (2.1) is well posed in the mathematical sense; i.e., we have existence and uniqueness of solutions in the viscosity sense. Before stating that result, let us recall the notion of viscosity solution for problem (2.1).

First we rewrite (2.1) in the form

$$\begin{aligned}&\frac{\partial u}{\partial t} - g(t,x)a_{ij}(\nabla u)\partial_{ij}u - \nu g(t,x)|\nabla u|\\ &+ (1-g(t,x))v(t,x).\nabla u = 0, \qquad (t,x)\in[0,\infty)\times R^2,\\ &u(0,x) = u_0(x), \quad x\in R^2,\end{aligned} \tag{2.2}$$

where $a_{ij}(p) = \delta_{ij} - \frac{p_i p_j}{|p|^2}$ if $p \neq 0$, $g(t,x) \in C([0,+\infty[\times R^2)$, $g(t,x) \geq 0$, $v(t,x) \in C(([0,+\infty[\times R^2), R^2)$. We use the usual notations $\partial_i u = \frac{\partial u}{\partial x_i}$, $\partial_{ij}u = \frac{\partial^2 u}{\partial x_i \partial x_j}$, and the classical Einstein summation convention in (2.2) and in all of what follows.

This equation should be solved in $D = [0,1]^2$ with Neumann boundary conditions. But to simplify the presentation and as is usual in the literature, we extend the images by reflection to R^2 and we look for solutions of (2.2) satisfying $u(x+2h) = u(x)$ for all $x \in R^2$ and all $h \in Z^2$. The initial condition $u_0(x)$ and the data $g(t,x)$ are taken extended to R^2 with the same periodicity as u.

Let us recall the definition of viscosity solution [7]. Let $u \in C([0,T]\times R^2)$ for some $T \in]0,\infty[$. We say that u is a viscosity subsolution of (2.2) if for any function $\phi \in C^2(R\times R^2)$ and any local maximum $(t_0,x_0) \in]0,T]\times R^2$ of $u-\phi$ we have the following: if $\nabla\phi(t_0,x_0) \neq 0$, then

$$\begin{aligned}&\frac{\partial\phi}{\partial t}(t_0,x_0) - g(t_0,x_0)a_{ij}(\nabla\phi(t_0,x_0))\partial_{ij}\phi(t_0,x_0) - \nu g(t_0,x_0)|\nabla\phi(t_0,x_0)|\\ &-(1-g(t_0,x_0))v(t_0,x_0).\nabla\phi(t_0,x_0) \leq 0;\end{aligned}$$

if $\nabla\phi(t_0,x_0) = 0$, then

$$\frac{\partial\phi}{\partial t}(t_0,x_0) - g(t_0,x_0)\limsup_{p\to 0}a_{ij}(p)\partial_{ij}\phi(t_0,x_0) \leq 0$$

and $u(0,x) \leq u_0(x)$ for all $x \in R^2$.

In the same way we define the notion of viscosity supersolution changing "local maximum" by "local minimum", "≤ 0" by "≥ 0", and "lim sup" by "lim inf" in the expressions above. A viscosity solution is a function which is both a viscosity subsolution and a viscosity supersolution.

THEOREM 2.1. *Suppose that* $g \in C([0,+\infty[\times R^2)$, $v \in C([0,+\infty[\times R^2, R^2)$, $g \geq 0$. *Suppose that for each* $T > 0$, *there exists a constant* C_T *such that*

$$\|(g^{1/2})_x(t,.)\|_\infty \leq C_T, \qquad \|v_x(t,.)\|_\infty \leq C_T \tag{2.3}$$

for all $t \in [0,T]$. *Let* $u_0, v_0 \in C(R^2)\cap W^{1,\infty}(R^2)$. *Then the following holds:*

(1) *Equations* (2.1), (2.2) *admit a unique viscosity solution* $u \in C([0,+\infty[\times R^2) \cap L^\infty(0,T;W^{1,\infty}(R^2))$ *for all* $T < \infty$. *Moreover, it satisfies*

$$\inf_{R^2}u_0 \leq u(t,x) \leq \sup_{R^2}u_0.$$

(2) *Let $v \in C([0, +\infty[\times R^2)$ be the viscosity solution of* (2.1) *with initial datum v_0. Then for all $T \in [0, \infty[$ we have*

$$\sup_{0\leq t\leq T} \|u(t, .) - v(t, .)\|_{L^\infty(R^2)} \leq \|u_0 - v_0\|_{L^\infty(R^2)}$$

($W^{1,\infty}(R^2)$ denotes the space of bounded Lipschitz functions of R^2).

The assumptions (2.3) are technical. They imply some smoothness of the coefficients in (2.2) required to prove existence and uniqueness of viscosity solutions for (2.2) using the method in [2], [4]. In particular, we require continuity in t and Lipschitz continuity in x for the vector field v. This implies a well-defined trajectory for the flow defined by $x_t = v(t, x)$ going through every point $x_0 \in R^2$. Observe that this does not cover the case of singular flows like occlusions. In spite of this, at the numerical level, we shall make some experiments showing that the model behaves correctly in the case of occlusions.

The proof follows the same steps as the proof of the corresponding existence and uniqueness result for model (1.1) (see [4, Thm. 3.1]) and we shall omit the details. As explained above, the active contour we are interested in will be the boundary of the zero level set of the unique solution of (2.1) given by the previous theorem.

3. An estimation of the velocity vector field. Our purpose in this section is to give an estimate of the vector field v, at least, on the boundaries of the moving objects. Many articles are devoted to the computation of the optical flow as a vector field close to v. Different approaches have been proposed to estimate the optical flow and they can be characterized as gradient-based models or tracking of special features.

In the first one, let us assume that the 3-D surface illumination does not vary from time t to time $t + \Delta t$. Therefore, if $I(x, y, t)$ is the grey level function at point (x, y) at time t, then $\frac{dI}{dt} = 0$; i.e.,

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

and if we take $v_1 = \frac{dx}{dt}$, $v_2 = \frac{dy}{dt}$, we have a single linear equation in the two unknowns v_1, v_2, the components of the optical flow field $I_x v_1 + I_y v_2 + I_t = 0$.

But this equation does not give us sufficient information to completely determine the two components v_1 and v_2. As a consequence, in order to compute the flow velocity $v = (v_1, v_2)$, we need to introduce additional constraints. Horn and Schunck [13] proposed an additional constraint by minimizing the square of the magnitude of the gradient of the optical flow velocity under the hypothesis that neighboring points on the objects have similar velocities. Based on this idea, Nagel [20] proposed the oriented smoothness constraint. In this case the unknown displacement vector v between the two consecutive images $I_1(x)$ and $I_2(x)$ is estimated by the following minimization problem:

$$\min \int\int dxdy\{(I_2(x) - I_1(x - v\Delta t))^2 + \alpha^2 \mathrm{trace}((\nabla v)^T W(\nabla v))\}.$$

The first term minimizes the difference between the grey value in images I_1 and I_2 while the second one is the formulation of the oriented smoothness constraint and the orientation is given by a weight matrix W. This matrix is a function of the first and second derivatives of the image I_1.

Enkelman [9] developed a modified formulation of the one above using multigrid methods to solve the Euler–Lagrange equations resulting for the minimization problem.

Other approaches proposed to determine the displacement vector field are based on the motion tracking model. The idea of this method is to do a tracking of special features from one frame to the next. The investigations are concerned with the tracking of points, lines, and contour segments (see Deriche and Faugeras [8]) or to group the features into consistent entities (see Meyer and Bouthemy [19]).

We are going to follow a different approach to estimate the velocity vector field v based on a recent work by Guichard [10], [11]. As argued in [10], it is generally acknowledged that the real velocity of objects observed in a movie is not always recoverable. In the following we shall assume that observed objects are Lambertian. Even with this assumption we can only hope to recover an apparent velocity. Now there are several ways to define the apparent velocity. Following Guichard [10], we choose to define the apparent velocity so that it gives the correct estimate of the velocity vector field when the objects are in translation motion.

Let us first introduce some useful notation. Let $I(x, y, t)$ be a movie; i.e., $I(x, y, t) \in C([0, \infty) \times R^2)$. We write $\nabla I = (I_x, I_y, 0)$ to denote the spatial gradient of the movie $I(x, y, t)$ to distinguish it from $DI = (I_x, I_y, I_t)$. We associate with DI two other vectors $DI^\perp$, $DI^\pm$ defined by

$$DI^\perp = (-I_y, I_x, 0), \qquad DI^\pm = (I_x I_t, I_y I_t, -(I_x^2 + I_y^2)).$$

If ∇I is not equal to zero, then $(DI, DI^\perp, DI^\pm)$ is an orthogonal basis of R^3. Observe that they are not normalized. If we define

$$T = \frac{DI}{|DI|}, \quad T^\perp = \frac{DI^\perp}{|DI^\perp|}, \quad T^\pm = \frac{DI^\pm}{|DI^\pm|},$$

then $(T, T^\perp, T^\pm)$ is an orthonormal basis of R^3. Finally we set

$$\gamma_1 = \langle D^2 I(DI^\perp), DI^\perp \rangle, \quad \gamma_2 = \langle D^2 I(DI^\perp), DI^\pm \rangle.$$

We observe that $\gamma_1 = |\nabla I|^3 \mathrm{curv}(I)$ where $\mathrm{curv}(I)(\mathbf{x}, t)$ is the spatial curvature of the level set of I, $[I \geq I(\mathbf{x}, t)]$ at $(\mathbf{x}, t)$.

Let us recall the definition of optical flow or apparent velocity given in [10]. The optical flow $v(x, y, t)$ is a function from R^3 to R^2 representing the velocity of the point (x, y) at time t. In order to simplify the computations we shall add a third component to the flow, which will be always equal to one, $V(x, y, t) = (v(x, y, t), 1)$. We denote by W the set of all "possible" velocity vectors

$$W = \{V = (v, 1) : v \in R^2\}.$$

In the classical definition of the flow one simply states that the grey level value of a point does not change during its motion [12], [13], [22]. This assumption implies that the derivative of I in the space–time direction V is zero; that is,

$$\langle V, DI \rangle = 0$$

and, therefore,

$$v.\nabla I = -I_t. \tag{3.1}$$

This yields at each point the component of the optical flow on the spatial gradient direction. Thus, when ∇I is not equal to zero, V is a vector of the form

$$V_\mu = \mu T^\perp - \frac{|DI|}{|\nabla I|} T^\pm$$

for some $\mu \in R$. We define at each point where $\nabla I \neq 0$ the set Σ of vectors of W which are orthogonal to DI:

$$\Sigma = \left\{ V_\mu = \mu T^\perp - \frac{|DI|}{|\nabla I|} T^\pm; \mu \in R \right\} = \left\{ V_\mu = \frac{1}{|\nabla I|^2}(\mu DI^\perp - DI^\pm); \mu \in R \right\}.$$

We see that the Lambertian assumption leads to (3.1) and does not give complete information on v. We still have one free parameter left—μ. This parameter corresponds to the component of the velocity vector on the spatial direction orthogonal to ∇I, which is, by definition, $T^{\perp}$.

Following [10], we define the velocity vector.

DEFINITION 3.1. *The velocity vector (or apparent velocity, or optical flow) is the vector V defined when $\nabla I \neq 0$ by*

$$V = \frac{1}{|\nabla I|^2}\left(DI^{\perp}\frac{\gamma_2}{\gamma_1} - DI^{\pm}\right). \tag{3.2}$$

Then, if we set $v_1 = \langle V, \vec{n}\rangle$, the component of V in the direction of the spatial gradient ∇I, where $\vec{n} = \frac{\nabla I}{|\nabla I|}$, and $v_2 = \langle V, T^{\perp}\rangle$, the component of V in the direction orthogonal to the gradient, we have

$$v_1 = -\frac{I_t}{|\nabla I|}, \quad v_2 = \frac{\gamma_2}{\gamma_1|\nabla I|}.$$

As noticed by Guichard [10, Prop. 2], if the movie corresponds to a translation motion, the vector V is equal to the real velocity. In view of our application below we shall give a slightly different formulation of this proposition.

LEMMA 3.2. *Let $I(\mathbf{x}, t)$ be a movie, $\mathbf{x} \in R^2$ (the image plane), $t \in [0, \tau]$. Assume that $I \in C^2(R^2 \times [0, \tau])$. Let $O(t) \subseteq R^2$ be a moving object in the image plane and assume that its boundary $\partial O(t)$ is a smooth simple Jordan curve. Let $\chi(\mathbf{x_0}, t)$, $t \in [t_0, \tau]$, denote the trajectory followed by the point $\mathbf{x_0} \in \partial O(t_0)$. Suppose that $\nabla I(\chi(\mathbf{x_0}, t), t) \neq 0$, $\gamma_1(\chi(\mathbf{x_0}, t), t) \neq 0$, $t \in [t_0, \tau]$. Assume also that*

$$I(\chi(\mathbf{x_0}, t), t) = I(\mathbf{x_0}, t_0), \tag{3.3}$$

$$\vec{n}(\chi(\mathbf{x_0}, t), t) = \vec{n}(\mathbf{x_0}, t_0) \tag{3.4}$$

for all $t \in [t_0, \tau]$. Then $(\chi_t, 1) = V$.

Proof. Differentiating both equations (3.3) and (3.4) with respect to t, we have

$$I_t + \nabla I.\chi_t = 0, \tag{3.5}$$

$$\vec{n}_t + \nabla_{\mathbf{x}}\vec{n}.\chi_t = 0 \tag{3.6}$$

on the trajectory $(\chi(\mathbf{x_0}, t), t)$. Let us write $\chi_t = w_1\vec{n} + w_2T^{\perp}$. From (3.5) we obtain

$$w_1 = -\frac{I_t}{|\nabla I|} = v_1. \tag{3.7}$$

To analyze (3.6), we write $Q_p = I - \frac{p\otimes p}{|p|^2}$, $p \in R^2$, and observe that

$$\vec{n}_t = \frac{1}{|\nabla I|}Q_{\nabla I}(\nabla I_t),$$

$$D_{\mathbf{x}}\vec{n} = \frac{1}{|\nabla I|}Q_{\nabla I}D^2_{\mathbf{x}}I.$$

Substituting in (3.6), we have

$$Q_{\nabla I}(\nabla I_t) + Q_{\nabla I}D^2_{\mathbf{x}}I(w_1\vec{n} + w_2T^{\perp}) = 0. \tag{3.8}$$

Since $Q_{\nabla I}(T^{\perp}) = T^{\perp}$ and taking the scalar product with $T^{\perp}$ in (3.8), we have

$$\langle \nabla I_t, T^{\perp}\rangle + w_1\langle D^2_{\mathbf{x}} I(\vec{n}), T^{\perp}\rangle + w_2\langle D^2_{\mathbf{x}} I(T^{\perp}), T^{\perp}\rangle = 0.$$

After some computations, we get

$$w_2\langle D^2_{\mathbf{x}} I(T^{\perp}), T^{\perp}\rangle = \frac{\gamma_2}{|\nabla I|^3}.$$

Now, observe that $\langle D^2_{\mathbf{x}} I(T^{\perp}), T^{\perp}\rangle = \frac{\gamma_1}{|\nabla I|^2}$. Hence,

$$w_2 = \frac{\gamma_2}{\gamma_1 |\nabla I|} = v_2. \tag{3.9}$$

Then, from (3.7) and (3.9), we obtain $(\chi_t, 1) = V$.

According to our previous discussion, to justify our approach to compute v, let us assume a rigid translation motion in such a way that (besides the Lambertian assumption (3.3)) we do not have variation on the shape of the moving objects. Under this assumption the normal vector at each point of the moving contour should remain constant; hence, (3.4) holds. In practice, assuming the continuity principle, i.e., there are no sudden changes between two consecutive frames, we try to put in correspondence the points in the contours of two consecutive images with the same normal (and least distance). We do this by minimizing the difference between normals belonging to three consecutive images. Indeed, given three consecutive images $I(\mathbf{x}, t - \Delta t)$, $I(\mathbf{x}, t)$, $I(\mathbf{x}, t + \Delta t)$ and a moving object $O(t)$, to find $\chi(\mathbf{x_0}, t)$ for some $\mathbf{x_0} \in \partial O(t)$, we minimize

$$|\vec{n}(\mathbf{x_2}, t + \Delta t) - \vec{n}(\mathbf{x_0}, t)| + |\vec{n}(\mathbf{x_1}, t - \Delta t) - \vec{n}(\mathbf{x_0}, t)| \tag{3.10}$$

on the set of points $\mathbf{x_1}, \mathbf{x_2} \in B(\mathbf{x_0}, \delta)$, $\delta > 0$, satisfying the restriction

$$|I(\mathbf{x_2}, t + \Delta t) - I(\mathbf{x_0}, t)| + |I(\mathbf{x_1}, t - \Delta t) - I(\mathbf{x_0}, t)| \leq \epsilon, \quad \epsilon > 0 \tag{3.11}$$

which is a reasonable way of introducing the Lambertian assumption. Let us conclude with some remarks.

Remarks.

(i) A rough estimate on δ can be obtained by taking the difference of two consecutive images.

(ii) In practice one can restrict the minimization to points $\mathbf{x_1}, \mathbf{x_2}$, where $|\nabla I|$ is sufficiently large. Even more, in practice, to save computations, one can forget about the verification of (3.11).

(iii) It would be also possible to minimize the first term in (3.10),

$$|\vec{n}(\mathbf{x_2}, t + \Delta t) - \vec{n}(\mathbf{x_0}, t)|,$$

but this strategy is not so robust as the minimization of both terms in (3.10).

(iv) Finally, let us observe that rougher estimates of V could be used (i.e., obtained along the lines of [14]) and expect that the snake of the model will correct this estimate.

4. Numerical scheme and experimental results.

4.1. Numerical scheme. To solve (2.1) numerically we split the equation in two:

$$\frac{\partial u}{\partial t} = g(t, x) \mid \nabla u \mid \left(\operatorname{div}\left(\frac{\nabla u}{\mid \nabla u \mid}\right) + \nu\right), \tag{4.1}$$

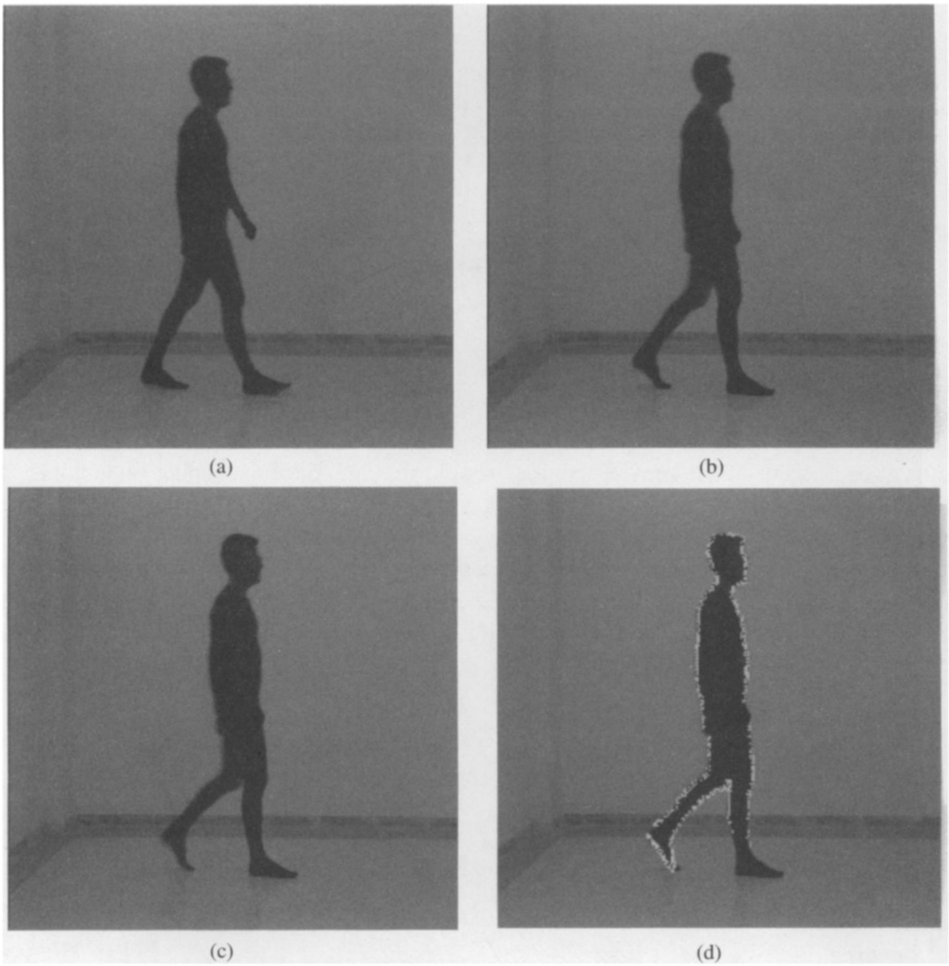


FIG. 1. *This is a sequence of three images given by the left top, right top, and left bottom. The fourth image given by the right bottom shows the superposition of the estimated contour and the real contour.*

$$\frac{\partial u}{\partial t} = (1 - g(t, x))v.\nabla u. \tag{4.2}$$

We have used the following iterative algorithm to estimate the motion of different objects in a sequence of images:

1. Initialize the first image and find the different contours by (4.1).

2. For each contour C_i compute the vector field at each point $x \in C_i$ by the method described in §3.

3. Apply (4.2) to give an initial estimate of the contour in the next image.

4. To correct possible errors on the estimate contour found in step 3, do some dilatation of it and take it as initial condition to start with point 1 again. This dilatation amounts to change v by $v + \epsilon \frac{\nabla I}{|\nabla I|}$ on (2.1), for some $\epsilon > 0$.

The numerical scheme for (4.1) is the Alvarez implicit scheme [2] used by [4], [5]. By our assumptions on $g(t, x)$ and step 1, we are only interested in computing v in a neighborhood of the snake C_i. The actual discretization of (4.2) can be avoided since the only thing we need is to displace the contour C_i to a new contour which can be obtained by knowing v on C_i.

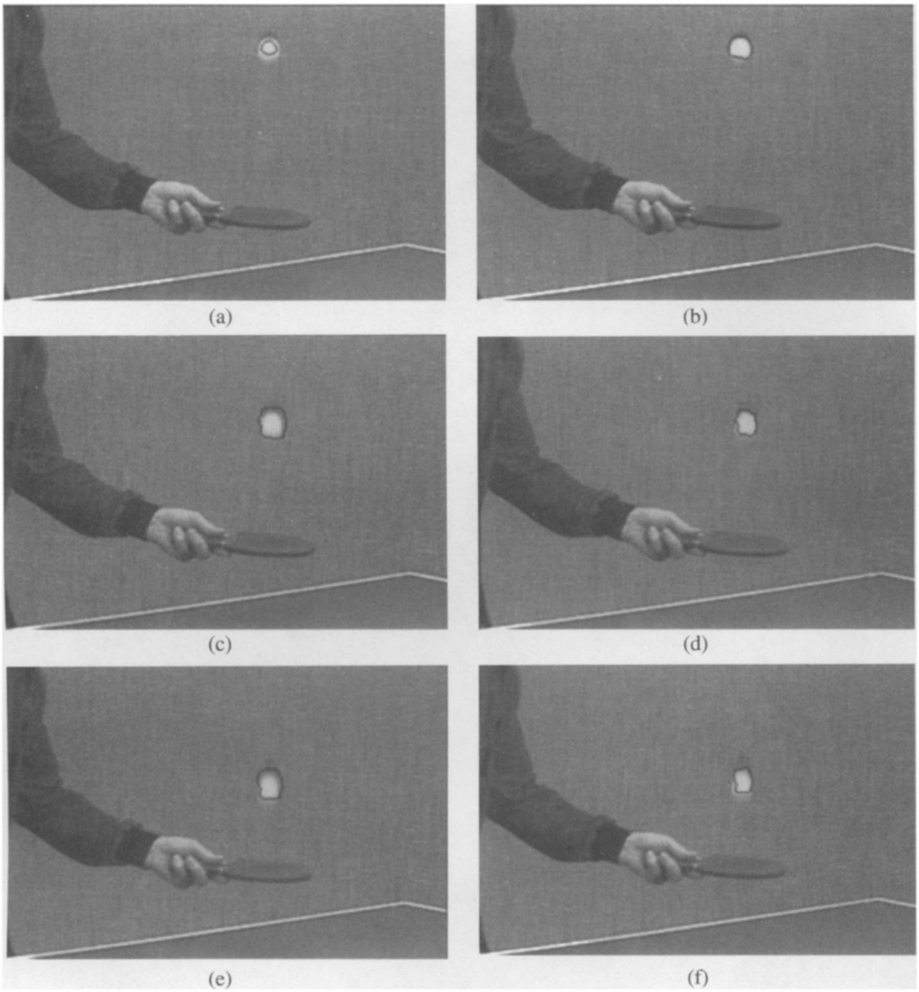


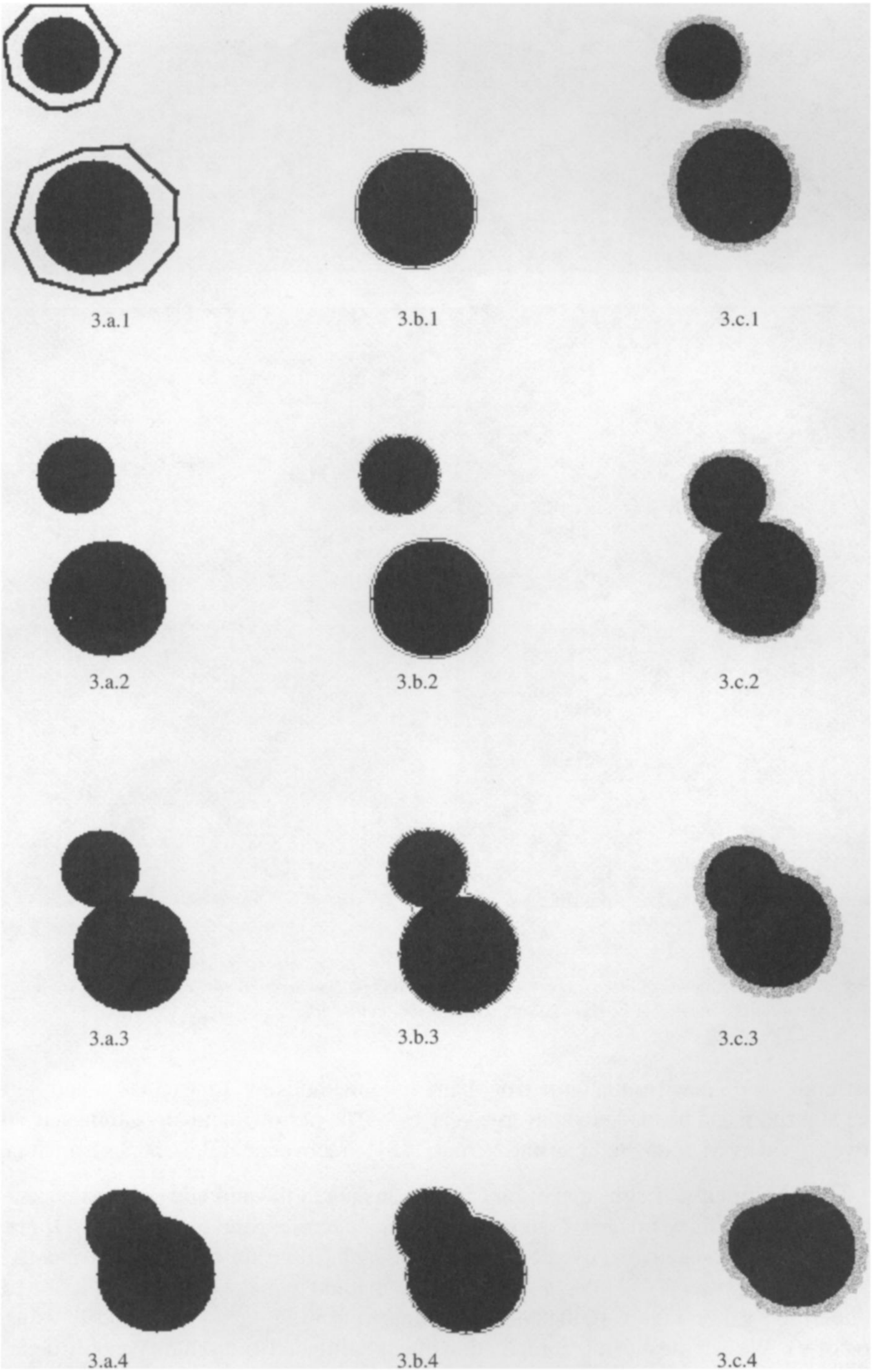
FIG. 2. *This is a sequence of three images. From an initial condition given by the left top image, we follow the motion of the contour of the ball. The right bottom image is the final result.*

Furthermore, using new results on narrow-band tube methods of Adalsteinsson and Sethian [1], the algorithm can be made to converge very fast. The narrow-band algorithms have been effectively used by Malladi, Sethian, and Vemuri [18] to recover complex shapes from images.

4.2. Experiments. Let us make some comments about the numerical experiments.

In Figure 1 the three images, Figures 1a, 1b, and 1c, correspond to $I(\mathbf{x}, t - \Delta t)$, $I(\mathbf{x}, t)$, and $I(\mathbf{x}, t + \Delta t)$, respectively. We computed the associated vector field or, equivalently, the correspondence $\chi(\mathbf{x_0}, t_0) \rightarrow \chi(\mathbf{x_0}, t_0 + \Delta t)$ by the method explained in §3. In Figure 1d we have superimposed on Figure 1c the contour estimated from Figures 1a, 1b, and 1c using the method of §3. We can see that our computation of v is sufficiently accurate. We only depicted, in white, those points for which the vector field was not zero.

In Figure 2 we have a sequence of different images and we want to follow the contour of the ball. The left top image gives the initial condition. The right top image, Figure 2b, shows the contour found by the snakes method. The other two images on the left column, Figures 2c and 2e, show the estimated contour using the strategy described in §3 applied to the contour


3.a.1 3.b.1 3.c.1

3.a.2 3.b.2 3.c.2

3.a.3 3.b.3 3.c.3

3.a.4 3.b.4 3.c.4

FIG. 3. *This is a sequence of eight images following the motion of two objects with occlusion. On the right column we can see the sequence we want to follow, the middle column shows the detection of the contours, and the left column shows the estimated contour from an image to the next.*

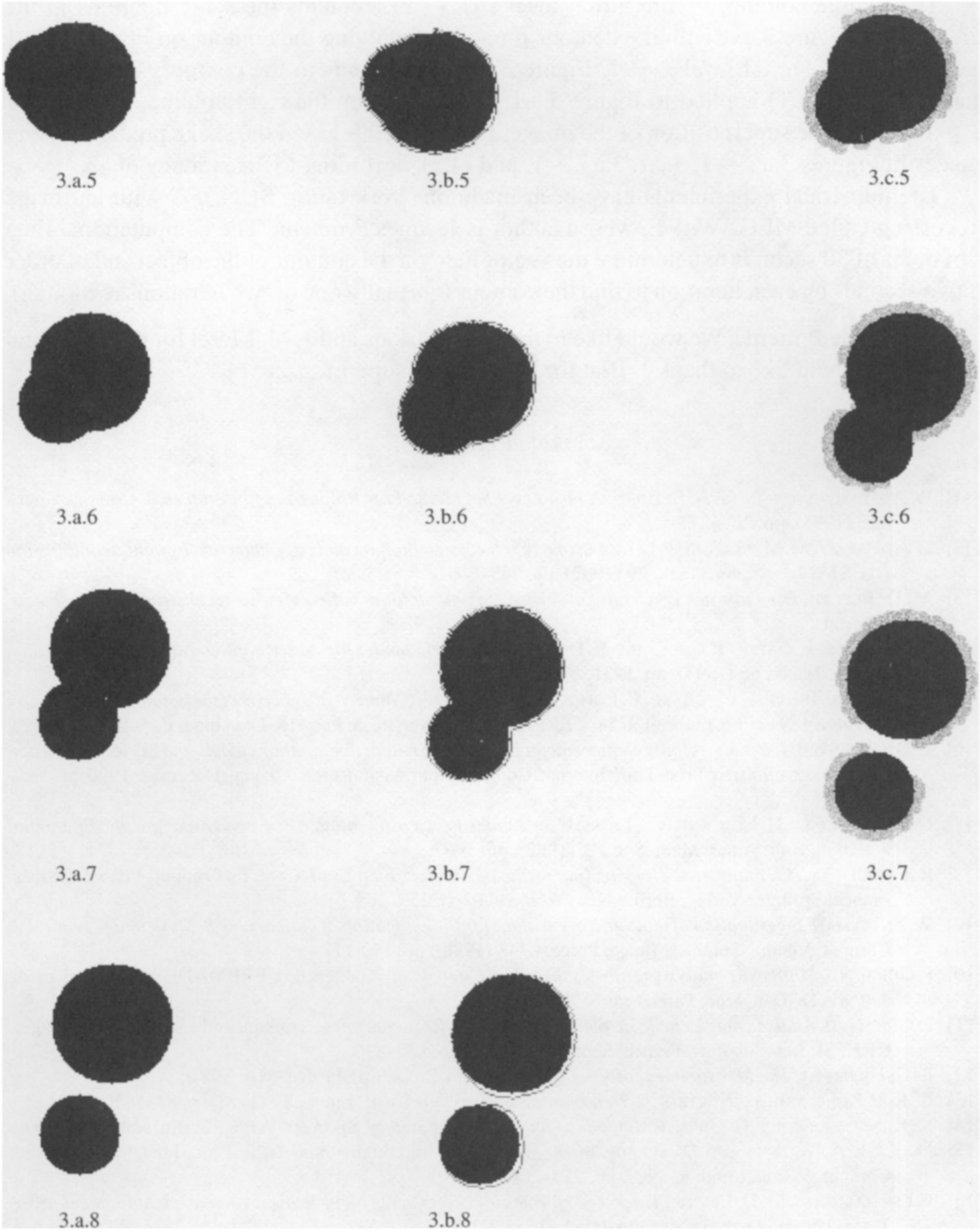


FIG. 3. *Continued.*

on the previous image in the right column, Figures 2b and 2d, respectively. The images on the right column show the contour found by evolving the contour according to (4.1) in the previous image of the left column.

In Figure 3 we present an experience to follow the motion of two contours with occlusion. The original sequence consists of nine images, Figures 3.a.i ($i = 0, \dots, 8$). The first one ($i = 0$) will be used only at the level of computation and has been omitted. At the left column, we show the images, Figures 3.a.i ($i = 1, \dots, 8$), of the original sequence.

The middle column, Figure 3.b at level i ($i \geq 2$), contains the same figure as in the left column, Figure 3.a.i, with the contour found by envolving the contour on Figure 3.c.i-1 according to (4.1). At level $i = 1$, Figure 3.b.1 corresponds to the contours found by the snakes method (4.1) applied to Figure 3.a.1. The image on the right column, Figure 3.c.i ($i \geq 1$), shows the superposition of the image, Figure 3.a.$i + 1$, and the snake predicted using images in Figures 3.a.$i - 1$, 3.a.i, 3.a.$i + 1$, and (4.2), according to the strategy of §3.

The numerical experiments have been made on Workstation SUN IPC with an image processing called MEGAWAVE, whose author is Jacques Froment. The computational time is of order of 30 seconds to determine the vector field on the contour of the object and of order of two seconds by each iteration to find the contour (normally one or two iterations is enough).

Acknowledgment. We would like to thank P. L. Lions and J. M. Morel for their help and advice. We would like to thank J. Blat for his constant support.

REFERENCES

[1] D. Adalsteinsson and J. A. Sethian, *A Fast Level Set Method for Propagating Interfaces*, J. Comput. Phys., 118 (1995), pp. 269–277.

[2] L. Alvarez, J. M. Morel, and P. L. Lions, *Image selective smoothing and edge detection by nonlinear diffusion* (II), SIAM J. Numer. Anal., 29 (1992), pp. 845–866.

[3] M. O. Berger, *How to track efficiently piecewise curved contours with a view to reconstruction 3D objects*, preprint.

[4] V. Caselles, F. Catte, T. Coll, and F. Dibos, *A geometric model for active contours in image processing*, Numer. Math., 66 (1993), pp. 1–31.

[5] T. Cohignac, F. Eve, F. Guichard, C. Lopez, and J. M. Morel, *Numerical analysis of the fundamental equation of image processing*, preprint 9254, CEREMADE, Université de Paris IX-Dauphine, Paris, France, 1992.

[6] L. D. Cohen and I. Cohen, *A finite element method applied to new active contour models and 3D reconstruction from cross sections*, in Proc. Third Internat. Conf. Comput. Vision, Osaka, Japan, December 1990, pp. 587–591.

[7] M. G. Crandall, H. Ishii, and P. L. Lions, *User's guide to viscosity solution of second order partial differential equation*, Bull. Amer. Math. Soc., 27(1992), pp. 1–67.

[8] R. Deriche and O. Faugeras, *Tracking line segments*, in Proc. First Euro. Conf. on Comput. Vision, Antibes, France, Springer-Verlag, Berlin, New York, 1990, pp. 259–268.

[9] W. Enkelman, *Investigation of multigrid algorithms for the estimation of optical flow fields in image sequence*, Comput. Vision, Graphics, Image Process., 43 (1988), pp. 150-177.

[10] F. Guichard, *Multiscale analysis of movies theory and algorithms*, Cahiers du CEREMADE n 9339, Université de Paris IX-Dauphine, Paris, France, 1993.

[11] F. Guichard, *Analyse multiechelle de films*, Quatorzième Colloque pur le Traitement du Signal et des Images, GRETSI, Juan-les-Pins, France, September, 1993, pp. 887–890.

[12] E. C. Hildreth, *The Measurement of Visual Motion*, MIT Press, Cambridge, MA, 1984.

[13] B. K. P. Korn and B. G. Schunck, *Determining optical flow*, Artif. Intell., 17 (1981), pp. 185–203.

[14] K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Springer-Verlag, Berlin, New York, 1990.

[15] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, in Proc. First Internat. Conf. Comput. Vision, London, 1987, pp. 259–268.

[16] F. Leymarie and M. D. Levine, *Tracking deformable objects in the plane using an active contour model*, IEEE Trans. Pattern Analysis Machine Intell., 16 (1993), pp. 617–634.

[17] R. Malladi, J. A. Sethian, and B. C. Vemuri, *Shape modelling with front propagation: A level set approach*, IEEE Trans. Pattern Analysis Machine Intell., 17 (1995), pp. 158–175.

[18] R. Malladi, J. A. Sethian, and B.C. Vemuri, *A fast level set based algorithm for topology independent shape modeling*, J. Math. Imaging Vision, A. Rosenfeld and Y. Kong, eds., to appear.

[19] F. Meyer and P. Bouthemy, *Region-based tracking in an image sequence*, Institut de Recherche en Informatique et Systemes Aleatoires (IRISA), Rennes, France, Publication interne n 664, 1992.

[20] H. Nagel, *On the estimation of optical flow: Relations between different approaches and some new results*, Artif. Intell., 33 (1987), pp. 299–324.

[21] S. Osher and J. A. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Physics, 79 (1988), pp. 12–49.

[22] A. Verri and T. Poggio, *Against quantificative optical flow*, in Proc. First Internat. Conf. on Comput. Vision, London, 1987, IEEE Press, Piscataway, NJ, 1987, pp. 171–180.