Dynamic Geodesic Snakes for Visual Tracking

Marc Niethammer and Allen Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA
{marcn,tannenba}@ece.gatech.edu

Abstract

Visual tracking using active contours is usually accomplished in a static framework. The active contour tracks the object of interest in a given frame of an image sequence, and then a subsequent prediction step ensures good initial placement for the next frame. This approach is unnatural; the curve evolution gets decoupled from the actual dynamics of the objects to be tracked. True dynamic approaches exist, all being marker particle based, and thus prone to the shortcomings of such particle-based implementations. In particular, topological changes are not handled naturally in this framework. The now "classical" level set approach is tailored for codimension one evolutions. However, dynamic curve evolution is at least of codimension two. We propose a natural, efficient, level set based approach for dynamic curve evolution which removes the artificial separation of segmentation and prediction, while retaining all the desirable properties of level set formulations. This is based on a new energy minimization functional which for the first time puts dynamics into the geodesic active contour framework.

1. Introduction

Typical geometric active contours [2, 8, 10, 12] are static. However, variational formulations many times appear to be dynamic, because the Euler-Lagrange equations are solved by gradient descent, introducing an *artificial* time parameter. To use static active contours for visual tracking one usually uses a two step approach. First, the curve evolves on a static frame until convergence (or for a fixed number of evolution steps). Second, the location of the curve in the next frame is predicted. In the simplest case this prediction is the current location. Better prediction results can be achieved by using optical flow information for example. Here, the curve is not moving intrinsically, but instead is placed in the solution's vicinity by an external observer (the prediction algorithm). The curve is completely unaware of its state. In contrast, the approaches by Terzopoulos and Szeliski [16] or Peterfreund [14] view curve evolution from a dynamical systems perspective. Both methods are marker particle based and are fast, but they might suffer from numerical problems (e.g. in the case of sharp corners (see [15] for details)). In the static case, level set methods are known to handle sharp corners, topological changes, and to be numerically robust. In their standard form, they are restricted to codimension one problems, and thus not suitable for dynamic curve evolution. Extensions of level set methods to higher codimensions exist and a level set formulation for dynamic curve evolution is desirable [11]. We will present a simple level set based dynamic curve evolution framework in this paper. Section 2 reviews the fundamentals of parametrized dynamic curve evolution. Section 3 introduces geometric dynamic curve evolution. The level set formulation is given in Section 4. Sections 5 and 6 deal with error injection into the evolution equations and occlusion detection respectively. Section 7 presents some simulation results.

2. Parametrized Dynamic Curves

We consider the evolution of closed curves of the form $\mathcal{C} : S^1 \times [0, \tau) \mapsto \mathbb{R}^2$ in the plane. Where $\mathcal{C} = \mathcal{C}(p, t)$ and $\mathcal{C}(0, t) = \mathcal{C}(1, t)$ [6], with t being the time, and $p \in [0, 1]$ the curve's parametrization. The classical formulation for dynamic curve evolution as proposed by Terzopoulos and Szeliski [16] is derived by means of minimization of the action integral

$$\mathcal{L} = \int_{t=t_0}^{t_1} L(t, \mathcal{C}, \mathcal{C}_t)\, dt, \tag{1}$$

where the subscripts denote partial derivatives. The Lagrangian $L = T - U$ is the difference between the kinetic and the potential energy. The potential energy of the curve

1063-6919/04 $20.00 © 2004 IEEE

is given by

$$\begin{aligned} U &= \int_0^1 U_{el} + U_{rig} + U_{pf} \; dp \\ &= \int_0^1 \frac{1}{2} w_1 \|\mathcal{C}_p\|^2 + \frac{1}{2} w_2 \|\mathcal{C}_{pp}\|^2 + g(\mathcal{C}) \; dp, \end{aligned}$$

where g is some potential function (with the desired location of the curve forming a potential well); U_{el}, U_{rig}, and U_{pf} are the elasticity, rigidity and potential field contributions, with their (possibly position-dependent) scalar weights w_1, and w_2. A common choice for the potential function is

$$g(\boldsymbol{x}) = \frac{1}{1 + \|G * \nabla I(\boldsymbol{x})\|^r},$$

where $\boldsymbol{x} = [x, y]^T$ are the image coordinates, I is the image, r is a positive integer, and G is a Gaussian of variance σ^2. The kinetic energy is

$$T = \int_0^1 \frac{1}{2} \mu \|\mathcal{C}_t\|^2 \; dp,$$

where μ corresponds to mass per unit length. The Lagrangian used is then

$$L = \int_0^1 \frac{1}{2} \mu \|\mathcal{C}_t\|^2 - \frac{1}{2} w_1 \|\mathcal{C}_p\|^2 - \frac{1}{2} w_2 \|\mathcal{C}_{pp}\|^2 - g(\mathcal{C}) \; dp.$$

Computing the first variation $\delta\mathcal{L}$ of the action integral (1) and setting it to zero yields the Euler-Lagrange equations for the candidate minimizer [17] in force balance form:

$$\mu \mathcal{C}_{tt} = \frac{\partial}{\partial p}(w_1 \mathcal{C}_p) - \frac{\partial^2}{\partial p^2}(w_2 \mathcal{C}_{pp}) - \nabla g.$$

This formulation is not intrinsic with respect to the geometry of the curve, since it is dependent upon its parametrization, p (see Xu *et al.* [18] for a discussion on the relationship between parametric and geometric active contours).

3. Geometric Dynamic Curves

Minimizing equation (1) using the Lagrangian

$$L = \int_0^1 \left(\frac{1}{2} \mu \|\mathcal{C}_t\|^2 - g \right) \|\mathcal{C}_p\| \; dp,$$

instead, results in

$$\begin{aligned} \mu \mathcal{C}_{tt} = -\mu(\mathcal{T} \cdot \mathcal{C}_{ts})\mathcal{C}_t - \mu(\mathcal{C}_t \cdot \mathcal{C}_{ts})\mathcal{T} - \\ - (\frac{1}{2}\mu\|\mathcal{C}_t\|^2 - g)\kappa\mathcal{N} - (\nabla g \cdot \mathcal{N})\mathcal{N}, \quad (2) \end{aligned}$$

which is intrinsic and a natural extension of the geodesic active contour approach [2, 8]. Here $\mathcal{N}$ is the unit inward normal, and $\mathcal{T} = \frac{\partial \mathcal{C}}{\partial s}$ the unit tangent vector to the curve. $\kappa = \mathcal{C}_{ss} \cdot \mathcal{N}$ denotes curvature and s is arclength [3].

Equation (2) describes a curve evolution that is only influenced by inertia terms and information on the curve itself. To increase robustness, region-based terms (see for example [13, 20, 19]) could be included in the formulation.

Normal Geometric Dynamic Curve Evolution

To be able to interpret the behavior of the curve evolution equation (2) it is instructive to derive the corresponding evolution equations for the tangential and normal velocity components of the curve.

We can write

$$\mathcal{C}_t = \alpha(p,t)\mathcal{T} + \beta(p,t)\mathcal{N}, \quad (3)$$

where the parametrization p is independent of time and travels with its particle, and α and β correspond to the tangential and the normal speed functions respectively. By substituting Equation (3) into Equation (2) and using results from [9] we obtain the two coupled partial differential equations:

$$\begin{aligned} \alpha_t &= -(\alpha^2)_s + 2\kappa\alpha\beta, \quad (4) \\ \beta_t &= -(\alpha\beta)_s + \left[(\frac{1}{2}\beta^2 - \frac{3}{2}\alpha^2) + \frac{1}{\mu} g \right] \kappa - \frac{1}{\mu} \nabla g \cdot \mathcal{N}. \end{aligned}$$

Clearly, $-(\alpha^2)_s$ and $-(\alpha\beta)_s$ are the transport terms for the tangential and the normal velocity along the contour. $g\kappa - \nabla g \cdot \mathcal{N}$ is the well known geodesic active contour image influence term. Note, that in contrast to the static geodesic active contour, this term does not directly influence the curve's position, but the curve's normal velocity. It resembles a force. Finally, the terms $2\kappa\alpha\beta$ and $(\frac{1}{2}\beta^2 - \frac{3}{2}\alpha^2)\kappa$ incorporate the dynamic elasticity effects of the curve. If we envision a rotating circle we can interpret the term $(\frac{1}{2}\beta^2 - \frac{3}{2}\alpha^2)\kappa$ as a rubberband (i.e. if we rotate the circle faster it will try to expand, but at the same time it will try to contract due to its then increasing normal velocity; oscillations can occur). If we restrict the movement of the curve to its normal direction (*i.e.* if we set $\alpha = 0$) we obtain

$$\beta_t = \frac{1}{2}\beta^2\kappa + \frac{1}{\mu} g\kappa - \frac{1}{\mu} \nabla g \cdot \mathcal{N}. \quad (5)$$

This is a much simpler evolution equation. In our case it is identical to the full evolution equation (4) if the initial tangential velocity is zero. The image term g only influences the normal velocity evolution β. It does not create any additional tangential velocity.

If there is an initial tangential velocity, and/or if the image influence g contributes to the normal velocity β *and* to the tangential velocity α, the normal evolution equation will not necessarily be equivalent to the full evolution equation (4). We can always parametrize a curve such that the tangential velocity term vanishes. Specifically, if we consider a reparametrization

$$\overline{\mathcal{C}}(q,t) = \mathcal{C}(\phi(q,t),t),$$

where $\phi : \mathbb{R} \times [0,T) \mapsto \mathbb{R}, p = \phi(q,t), \phi_q > 0$ then

$$\frac{\partial \overline{\mathcal{C}}}{\partial t} = \frac{\partial \mathcal{C}}{\partial t} + \frac{\partial \mathcal{C}}{\partial p}\frac{\partial \phi}{\partial t}.$$

The time evolution for $\overline{C}$ can then be decomposed into

$$\overline{\mathcal{C}}_t = \overline{\alpha}\mathcal{T} + \overline{\beta}\mathcal{N} = (\alpha(\phi(q,t),t) + \|\mathcal{C}_p(\phi(q,t),t)\|\phi_t)\mathcal{T} + \overline{\beta}\mathcal{N},$$

where

$$\begin{array}{rcl} \overline{\alpha} & = & \alpha(\phi(q,t),t) + \|\mathcal{C}_p(\phi(q,t),t)\|\phi_t \\ \overline{\beta} & = & \beta(\phi(q,t),t). \end{array}$$

If we choose ϕ as

$$\phi(q,t)_t = -\frac{\alpha(\phi(q,t),t)}{\|\mathcal{C}_p(\phi(q,t),t)\|}$$

we obtain

$$\overline{\mathcal{C}}_t = \overline{\beta}\mathcal{N},$$

which is a curve evolution equation without a tangential component. For all times, t, the curve $\overline{\mathcal{C}}$ will move along its normal direction. However, the tangential velocity is still present in the update equation for $\overline{\beta}$. After some algebraic manipulations, we arrive at

$$\mu(\beta_p \phi_t + \beta_t) = \left(\frac{1}{2}\mu\beta^2 + g\right)\kappa - (\nabla g \cdot \mathcal{N})\mathcal{N}, \qquad (6)$$

which depends on the time derivative of the reparametrization function ϕ, which in turn depends on the tangential component α. The left hand side of Equation (6) represents a transport term along the curve, the speed of which depends on the time derivative of the reparametrization function ϕ.

4. Level Set Formulation

There are different ways to implement the derived curve evolution equations (see for example [11]). We distinguish full and partial level set implementations. In the full case, curves evolve in a space consistent with the dimensionality of the problem. Geometric dynamic curve evolution would thus be performed in $\mathbb{R}^4$ in the simplest case. Normal geometric dynamic curve evolution would be at least a problem in $\mathbb{R}^3$. If n is the dimensionality of the problem the curve will be implicitly described by the zero level set of an n-dimensional vector distance function or the intersection of $n-1$ hypersurfaces [5]. Full level set approaches of this form are computationally expensive, since the evolutions are performed in high dimensional spaces. Furthermore, it is not obvious how to devise a methodology comparable to a narrow band scheme [4] in the case of a representation based on intersecting hypersurfaces.

A partial level set approach uses a level set formulation for the propagation of an implicit description of the curve itself (thus allowing for topological changes), but explicitly propagates the velocity information associated with every point on the contour by means of possibly multiple transport equations. It sacrifices computational efficiency (a narrow band implementation is possible in this case, and the evolution is performed in a low dimensional space) for object separation: tracked objects that collide will be merged.

In what follows we will restrict ourselves to a partial level set implementation of the normal geometric dynamic curve evolution.

Partial Level Set Approach for the Normal Geometric Curve Evolution

The curve $\mathcal{C}$ is represented as the zero level set of the function

$$\Phi(\boldsymbol{x}(t),t) : \mathbb{R}^2 \times \mathbb{R}^+ \mapsto \mathbb{R},$$

where $\boldsymbol{x}(t) = (x(t),y(t))^T$ is a point in the image plane. We assume $\Phi > 0$ outside the curve $\mathcal{C}$ and $\Phi < 0$ inside the curve $\mathcal{C}$. Since the evolution of the curve's shape is independent of the tangential velocity we can write the level set evolution equation for an arbitrary velocity $\boldsymbol{x}_t$ as

$$\Phi_t + \|\nabla\Phi\|\mathcal{N} \cdot \boldsymbol{x}_t = 0, \qquad (7)$$

where

$$\mathcal{N} = -\frac{\nabla\Phi}{\|\nabla\Phi\|}.$$

In our case $\boldsymbol{x}_t = \tilde{\beta}\mathcal{N}$, where

$$\tilde{\beta}(\boldsymbol{x},t) = \beta(p,t) \qquad (8)$$

is the spatial normal velocity at the point $\boldsymbol{x}$. This simplifies Equation (7) to

$$\Phi_t - \tilde{\beta}\|\nabla\Phi\| = 0. \qquad (9)$$

Substituting Equation (8) into Equation (5) and using the relation

$$\kappa = \nabla \cdot \left(\frac{\nabla\Phi}{\|\nabla\Phi\|}\right)$$

yields

$$\tilde{\beta}_t - \tilde{\beta}\nabla\tilde{\beta}\frac{\nabla\Phi}{\|\nabla\Phi\|} = (\frac{1}{2}\tilde{\beta}^2 + \frac{1}{\mu}g)\kappa + \frac{1}{\mu}\nabla g \cdot \frac{\nabla\Phi}{\|\nabla\Phi\|}. \qquad (10)$$

The left hand side of Equation (10) is the material derivative for the normal velocity. If we choose to use extension velocities, Equation (10) simplifies to

$$\tilde{\beta}_t = (\frac{1}{2}\tilde{\beta}^2 + \frac{1}{\mu}g)\kappa + \frac{1}{\mu}\nabla g \cdot \frac{\nabla\Phi}{\|\nabla\Phi\|}.$$

Since the extensions are normal to the contours, normal propagation of the level set function will guarantee a constant velocity value along the propagation direction (up to numerical errors). Specifically $\nabla\tilde{\beta}\perp\nabla\Phi$ in this case and thus

$$\nabla\Phi \cdot \nabla\tilde{\beta} = 0.$$

For an alternative derivation, we can change our Lagrangian, and extend it over a range of level sets. For each time t, and $0 \leq r \leq 1$ let

$$\mathcal{C}^{(r)}(t) := \{(x, y) \in \mathbb{R}^2 : \Phi(x, y, t) = r\}.$$

Using the Lagrangian

$$L = \int_0^1 \int_{\mathcal{C}^{(r)}(t)} \left(\frac{1}{2}\mu\beta^2 - g\right)\, ds\, dr$$

we obtain the action integral

$$\mathcal{L} = \int_t \int_0^1 \int_{\mathcal{C}^{(r)}(t)} \left(\frac{1}{2}\mu\beta^2 - g\right)\, ds\, dr\, dt,$$

which is

$$\begin{aligned}\mathcal{L} &= \int_0^1 \int_0^T \int_{\mathcal{C}^{(r)}(t)} (\frac{1}{2}\mu\beta^2 - g)\, ds\, dt\, dr \\ &= \int_0^T \left(\int_0^1 \int_{\mathcal{C}^{(r)}(t)} (\frac{1}{2}\mu\beta^2 - g)\, d\mathcal{H}^1 \lfloor_{\mathcal{C}^{(r)}(t)}\, dr\right) dt \\ &= \int_0^T \int_\Omega (\frac{1}{2}\mu\tilde{\beta}^2 - g)\|\nabla\Phi\|\, dx\, dy\, dt, \qquad (11)\end{aligned}$$

where $\mathcal{H}^1$ is the one-dimensional Hausdorff measure and we applied the coarea formula (see [1]). This casts the minimization problem into minimization over an interval of level sets in a fixed coordinate frame (x and y are time independent coordinates in the image plane). Using Equation (9) we can express $\tilde{\beta}$ as

$$\tilde{\beta} = \frac{\Phi_t}{\|\nabla\Phi\|}. \qquad (12)$$

Substituting (12) into Equation (11) yields

$$\mathcal{L} = \int_0^T \int_\Omega (\mu\frac{\Phi_t^2}{2\|\nabla\Phi\|} - g\|\nabla\Phi\|)\, dx\, dy\, dt := S[\Phi],$$

which is the new Φ-dependent action integral to be minimized. Then, $\delta S = 0$ if and only if

$$\frac{\partial}{\partial t}\left(\frac{\Phi_t}{\|\nabla\Phi\|}\right) = \nabla \cdot \left(\left(\frac{g}{\mu} + \frac{\Phi_t^2}{\|\nabla\Phi\|^2}\right)\frac{\nabla\Phi}{\|\nabla\Phi\|}\right).$$

The curve evolution is thus governed by the equation system:

$$\begin{aligned}\tilde{\beta}_t &= \nabla \cdot \left(\frac{\nabla\Phi}{\|\nabla\Phi\|}(\frac{g}{\mu} + \frac{1}{2}\tilde{\beta}^2)\right), \qquad (13) \\ \Phi_t &= \tilde{\beta}\|\nabla\Phi\|.\end{aligned}$$

Expanding Equation (13) yields again

$$\tilde{\beta}_t = (\frac{1}{2}\tilde{\beta}^2 + \frac{1}{\mu}g)\kappa + \frac{1}{\mu}\nabla g \cdot \frac{\nabla\Phi}{\|\nabla\Phi\|} + \tilde{\beta}\nabla\tilde{\beta} \cdot \frac{\nabla\Phi}{\|\nabla\Phi\|}.$$

Note, that the equation system (13) constitutes a conservation law for the normal velocity $\tilde{\beta}$. The propagation of the level set function Φ is described (as usual) by an equation of Hamilton-Jacobi type.

5. Error Injection

We know that a system governed by a time-independent Lagrangian (i.e. $L_t \equiv 0$) will preserve energy [17]. This is not necessarily desirable. Envision a curve evolving on a static image with an initial condition of zero normal velocity everywhere and with an initial position of nonminimal potential energy. The curve will oscillate in its potential well indefinitely. A solution to this problem is to dissipate energy (see [16]). This can be accomplished by simply adding a friction term to Equation (13). However, to increase robustness it is desirable to be able to dissipate and to add energy to the system in a directed way. A principled way to do this would be to use an observer to drive the system state of the evolving curve to the object(s) to be tracked. In our case this is not straightforward, since we are dealing with an infinite dimensional nonlinear system. In order for the curve to approximate the dynamic behavior of the tracked objects we use error injection. This will guarantee convergence of the curve to the desired object(s) if the curve is initially in the appropriate basin of attraction.

We need an estimated position and velocity vector for every point on the curve $\mathcal{C}$. Define the line through the point $\boldsymbol{x}(s)$ on the current curve as

$$l(s, p) := \boldsymbol{x}(s) - p\mathcal{N}$$

and the set of points in an interval (a, b) on the line as

$$L(a, b, s) := \{\boldsymbol{x} : \boldsymbol{x} \in l(s, p), p \in (a, b)\}.$$

Define

$$\begin{aligned}f(s) &:= \inf\{p : p < 0, \Phi(\boldsymbol{x}) \leq 0\, \forall\boldsymbol{x} \in L(p, 0, s)\}, \\ t(s) &:= \sup\{p : p > 0, \Phi(\boldsymbol{x}) \geq 0\, \forall\boldsymbol{x} \in L(0, p, s)\}.\end{aligned}$$

Our set of estimated contour point candidates Z is the set of potential edge points in $L(f, t, s)$

$$\begin{aligned}&Z(L(f, t, s)) := \{\boldsymbol{x} : \boldsymbol{x} \in L(f, t, s), \\ &\quad \exists\epsilon > 0 : \|\nabla(G * I(\boldsymbol{x}))\| > \|\nabla(G * I(\boldsymbol{y}))\| \\ &\qquad \forall\boldsymbol{y} \in L(f, t, s) \cap B_\epsilon(\boldsymbol{x}), \boldsymbol{y} \neq \boldsymbol{x}\},\end{aligned}$$

where G is a Gaussian, $B_\epsilon(\boldsymbol{x})$ is the disk around $\boldsymbol{x}$ with radius ϵ, and I is the current image. Given some likelihood function $m(\boldsymbol{z})$[1] the selected contour point is the likelihood maximum

$$\boldsymbol{x}_c(s) = \underset{\boldsymbol{z} \in Z(L(f,t,s))}{\arg\max} \ m(\boldsymbol{z}),$$

at position

$$p_c = d(\boldsymbol{x}, s) = (\boldsymbol{x}(s) - \boldsymbol{x}_c(s))^T \mathcal{N}.$$

It is sufficient to estimate normal velocity, since the curve evolution equation does not take tangential velocity components into account. The estimation then can be performed (assuming we have brightness constancy from image frame to image frame for a moving image point) by means of the optical flow constraint *without* the need for regularization[2]. The optical flow constraint is given as

$$I_t + uI_x + vI_y = 0,$$

where $u = x_t$ and $v = y_t$ are the velocities in the x and the y direction respectively, and I denotes image intensity. We restrict the velocities to the normal direction by setting

$$\begin{pmatrix} u \\ v \end{pmatrix} = \gamma \frac{\nabla I}{\|\nabla I\|}.$$

This yields

$$\gamma = -\frac{I_t}{\|\nabla I\|}$$

and thus the desired velocity estimate

$$\begin{pmatrix} u \\ v \end{pmatrix} = -I_t \frac{\nabla I}{\|\nabla I\|^2}.$$

We define

$$\begin{aligned} \overline{\beta} &:= -\gamma \frac{\nabla I}{\|\nabla I\|} \cdot \frac{\nabla \hat{\Phi}}{\|\nabla \hat{\Phi}\|}, \\ \overline{\Phi} &:= -\|\boldsymbol{x}_c - \boldsymbol{x}\| \operatorname{sign}\left(\hat{\Phi}(\boldsymbol{x}_c)\right). \end{aligned}$$

We propose using the following observer-like dynamical system

$$\begin{aligned} \hat{\Phi}_t &= \left(m(\boldsymbol{x}_c) K_\Phi (\overline{\Phi} - \hat{\Phi}) + \hat{\beta} + \gamma\kappa \right) \|\nabla \hat{\Phi}\| \\ \hat{\beta}_t &= m(\boldsymbol{x}_c) K_\beta (\overline{\beta} - \hat{\beta}) + \left(\frac{1}{2}\hat{\beta}^2 + \frac{g}{\mu} \right) \kappa + \\ &\quad + \frac{1}{\mu} \nabla g \cdot \frac{\nabla \hat{\Phi}}{\|\nabla \hat{\Phi}\|} + \delta \hat{\beta}_{ss}, \end{aligned} \tag{14}$$

to dynamically blend the current curve $\hat{\mathcal{C}}$ into the desired curve $\mathcal{C}$. Here, K_Φ and K_β are the error injection gains for $\hat{\Phi}$ and $\hat{\beta}$ respectively. Any terms related to image features are computed at the current location $\boldsymbol{x}$ of the contour. The error injection gains are weighted by the likelihood $m(\boldsymbol{x}_c)$ of the correspondence points as a measure of prediction quality. The additional terms $\kappa\gamma$ and $\delta\hat{\beta}_{ss}$ with tunable weighting factors γ and δ are introduced to allow for curve and velocity regularization if necessary.

$$\kappa = \nabla \cdot \left(\frac{\nabla \hat{\Phi}}{\|\nabla \hat{\Phi}\|} \right)$$

and

$$\hat{\beta}_{ss} = \mathcal{N}^T \begin{pmatrix} \hat{\beta}_{yy} & -\hat{\beta}_{xy} \\ -\hat{\beta}_{xy} & \hat{\beta}_{xx} \end{pmatrix} \mathcal{N} + \kappa \nabla \hat{\beta} \cdot \mathcal{N}.$$

In case no correspondence point for a point on the zero level set of $\hat{\Phi}$ is found, the evolution equation system (14) is replaced by

$$\begin{aligned} \hat{\Phi}_t &= (\hat{\beta} + \gamma\kappa) \|\nabla \hat{\Phi}\| \\ \hat{\beta}_t &= \delta \hat{\beta}_{ss} \end{aligned} \tag{15}$$

for this point.

6. Occlusion Detection

To assess the curve's prediction capability we implemented the following simple occlusion detection algorithm[3] based on ideas in [7]. The inside and the outside correspondence points are defined as (see Figure (1))

$$\begin{aligned} \boldsymbol{x}_i(s) &= \underset{\boldsymbol{z} \in Z(L(f,p_c,s))}{\arg\max} \ m(\boldsymbol{z}), \\ \boldsymbol{x}_o(s) &= \underset{\boldsymbol{z} \in Z(L(p_c,t,s))}{\arg\max} \ m(\boldsymbol{z}). \end{aligned}$$

The occlusion detection strategy is split into the following six subcases for every point on the contour

(0) There is no correspondence point.

(1) Only the correspondence point is present.

(2) The point is moving outward, the correspondence point is present, but not its outside correspondence point.

(3) The point is moving inward, the correspondence point is present, but not its inside correspondence point.

(4) The point is moving outward, both the correspondence point and its outside correspondence point are present.

[1] This will be problem specific.

[2] Note, that we compute this estimate only on few chosen points in Z.

[3] More sophisticated, and less parametric, occlusion detection algorithms are conceivable; however, this is not the main focus of our work, and the one proposed is sufficient to show that the dynamic geodesic snake can handle occlusions when combined with a suitable occlusion detection algorithm.

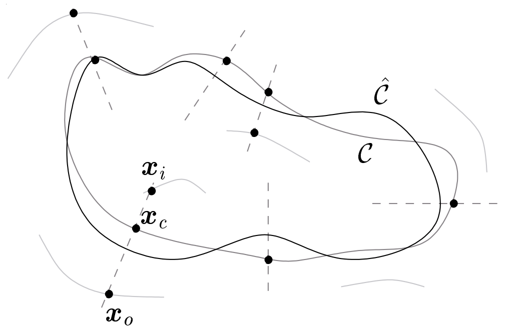


Figure 1. Correspondence point x_c, inside correspondence point x_i, and outside correspondence point x_o of the curve $\hat{\mathcal{C}}$. $\mathcal{C}$ represents the contour of the object to be tracked.

(5) The point is moving inward, both the correspondence point and its inside correspondence point are present.

We define the following Gaussian conditional probabilities

$$\begin{aligned}
Pr(t_{occ}|occ) &= \frac{2}{\sqrt{2\pi}\sigma_t} e^{-\frac{(t_{occ}-\mu_t)^2}{2\sigma_t^2}} \\
Pr(v_a|occ) &= \frac{1}{\sqrt{2\pi}\sigma_v} e^{-\frac{(v_a-\mu_v)^2}{2\sigma_v^2}} \\
Pr(t_{occ}|\bar{o}\bar{c}\bar{c}) &= \frac{2}{\sqrt{2\pi}\sigma_{\bar{t}}} e^{-\frac{(t_{occ}-\mu_{\bar{t}})^2}{2\sigma_{\bar{t}}^2}} \\
Pr(v_a|\bar{o}\bar{c}\bar{c}) &= \frac{1}{\sqrt{2\pi}\sigma_{\bar{v}}} e^{-\frac{(v_a-\mu_{\bar{v}})^2}{2\sigma_{\bar{v}}^2}},
\end{aligned}$$

where t_{occ} is the estimated time to occlusion, v_a is the velocity of the point ahead, overlined symbols denote negated values (i.e. $\bar{o}\bar{c}\bar{c}$ means not occluded), $Pr(t_{occ}|occ)$, $Pr(v_a|occ)$ are the probabilities of t_{occ} and v_a given an occlusion, and $Pr(t_{occ}|\bar{o}\bar{c}\bar{c})$ and $Pr(v_a|\bar{o}\bar{c}\bar{c})$ given there is no occlusion respectively. The corresponding standard deviations are σ_t, σ_v, $\sigma_{\bar{t}}$, and $\sigma_{\bar{v}}$; the means are μ_t, μ_v, $\mu_{\bar{t}}$, $\mu_{\bar{v}}$. To compute the values of t_{occ} and v_a we make use of the currently detected correspondence point $\boldsymbol{x}_c$, and its interior $\boldsymbol{x}_i$ and exterior $\boldsymbol{x}_o$ correspondence points.

The probability for an occlusion is given by Bayes' formula as

$$Pr(occ|v_a, t_{occ}) = \\ \frac{Pr(v_a, t_{occ}|occ)Pr(occ)}{Pr(v_a, t_{occ}|occ)Pr(occ) + Pr(v_a, t_{occ}|\bar{o}\bar{c}\bar{c})Pr(\bar{o}\bar{c}\bar{c})}.$$

We initialize with a uniform probability distribution of $Pr(occ) = 0$. The priors at time step $n+1$ are the smoothed posteriors of time step n. In case (0) $Pr(occ|v_a, t_{occ}) = Pr(occ)$ (i.e. the probability is left unchanged), in all other cases

$$Pr(occ|v_a, t_{occ}) = \frac{Pr^i_{occ}Pr(occ)}{Pr^i_{occ}Pr(occ) + Pr^i_{\bar{o}\bar{c}\bar{c}}Pr(\bar{o}\bar{c}\bar{c})},$$

where

$$\begin{aligned}
Pr^1_{occ} &= Pr(v_a = v_c|occ), \\
Pr^1_{\bar{o}\bar{c}\bar{c}} &= Pr(v_a = v_c|\bar{o}\bar{c}\bar{c}),
\end{aligned}$$

$$\begin{aligned}
Pr^2_{occ} &= \begin{cases} Pr(v_a = v_c|occ) & \text{if } \boldsymbol{x}_c \text{ outside of } \mathcal{C}, \\ 0 & \text{otherwise,} \end{cases} \\
Pr^2_{\bar{o}\bar{c}\bar{c}} &= \begin{cases} Pr(v_a = v_c|\bar{o}\bar{c}\bar{c}) & \text{if } \boldsymbol{x}_c \text{ outside of } \mathcal{C}, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}$$

$$\begin{aligned}
Pr^3_{occ} &= \begin{cases} Pr(v_a = v_c|occ) & \text{if } \boldsymbol{x}_c \text{ inside of } \mathcal{C}, \\ 0 & \text{otherwise,} \end{cases} \\
Pr^3_{\bar{o}\bar{c}\bar{c}} &= \begin{cases} Pr(v_a = v_c|\bar{o}\bar{c}\bar{c}) & \text{if } \boldsymbol{x}_c \text{ inside of } \mathcal{C}, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}$$

$$\begin{aligned}
Pr^4_{occ} &= Pr(v_a = v_o|occ)Pr(t_{occ} = t^o_{occ}|occ), \\
Pr^4_{\bar{o}\bar{c}\bar{c}} &= Pr(v_a = v_o|\bar{o}\bar{c}\bar{c})Pr(t_{occ} = t^o_{occ}|\bar{o}\bar{c}\bar{c}),
\end{aligned}$$

$$\begin{aligned}
Pr^5_{occ} &= Pr(v_a = v_i|occ)Pr(t_{occ} = t^i_{occ}|occ), \\
Pr^5_{\bar{o}\bar{c}\bar{c}} &= Pr(v_a = v_i|\bar{o}\bar{c}\bar{c})Pr(t_{occ} = t^i_{occ}|\bar{o}\bar{c}\bar{c}),
\end{aligned}$$

and

$$\begin{aligned}
v_c &= \bar{\beta}(\boldsymbol{x}_c) & v_o &= \bar{\beta}(\boldsymbol{x}_o) \\
v_i &= \bar{\beta}(\boldsymbol{x}_i) & v &= \bar{\beta}(\boldsymbol{x}) \\
t^i_{occ} &= \frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|}{|v - v_i|} & t^o_{occ} &= \frac{\|\boldsymbol{x} - \boldsymbol{x}_o\|}{|v - v_o|}.
\end{aligned}$$

To estimate the current rigid body motion, the equation system

$$\begin{aligned}
(u_r\ ,\ v_r)^T \int_{\mathcal{C}} n^1 \mathcal{N}\, ds &= -\int_{\mathcal{C}} n^1 \beta\, ds \\
(u_r\ ,\ v_r)^T \int_{\mathcal{C}} n^2 \mathcal{N}\, ds &= -\int_{\mathcal{C}} n^2 \beta\, ds,
\end{aligned}$$

is solved, where $\mathcal{N} = (n^1, n^2)^T$. We set $\mu_{\bar{v}} = -(u_r\ ,\ v_r)^T \mathcal{N}$ and $\mu_v = 0$.

The evolution equation is changed to

$$\begin{aligned}
\hat{\Phi}_t &= \Big(Pr(\bar{o}\bar{c}\bar{c})\Big(m(\boldsymbol{x}_c)K_\Phi(\bar{\Phi} - \hat{\Phi})\Big) + \hat{\beta} + \gamma\kappa\Big)\|\nabla\hat{\Phi}\| \\
\hat{\beta}_t &= Pr(\bar{o}\bar{c}\bar{c})\left(m(\boldsymbol{x}_c)K_\beta(\bar{\beta} - \hat{\beta}) + \left(\frac{1}{2}\hat{\beta}^2 + \frac{g}{\mu}\right)\kappa\right) + \\
&\quad Pr(\bar{o}\bar{c}\bar{c})\frac{1}{\mu}\nabla g\kappa + \delta\hat{\beta}_{ss}.
\end{aligned}$$

This is an interpolation between the systems (14) and (15) based on the occlusion probability.

7. Simulation Results

The algorithm is tested on two real video sequences. Figure (2) shows three frames of a fish sequence and Figure (3) shows three frames of a car sequence respectively. In both cases occlusions occur. For the fish sequence no occlusion detection is performed. Define[4]

$$q(x) := \frac{1}{1+e^{-(p_1+x)}}, \qquad r := q(0) + \frac{e^{-p_1}}{q(0)^2} p_2,$$

$$w(\boldsymbol{x}) := \begin{cases} q(d(\boldsymbol{x})) & \text{if } d(\boldsymbol{x}) \leq 0 \\ q(0) + \frac{e^{-p_1}}{q(0)^2} d(\boldsymbol{x}) & \text{if } 0 < d(\boldsymbol{x}) \leq p_2 \\ r - \frac{r}{p_3 - p_2}(d(\boldsymbol{x}) - p_2) & \text{if } p_2 < d(\boldsymbol{x}) \leq p_3 \\ 0 & \text{otherwise.} \end{cases}$$

The used likelihood function for the fish sequence is

$$m(\boldsymbol{z}) = e^{-\left(\frac{(g(\boldsymbol{z})-\mu_g)^2}{2\sigma_g^2} + \frac{(I(\boldsymbol{z})-\mu_I)^2}{2\sigma_I^2}\right)} w(\boldsymbol{z}).$$

It depends on the image intensity I, the potential function g and the distance d to the contour. For the car sequence we define

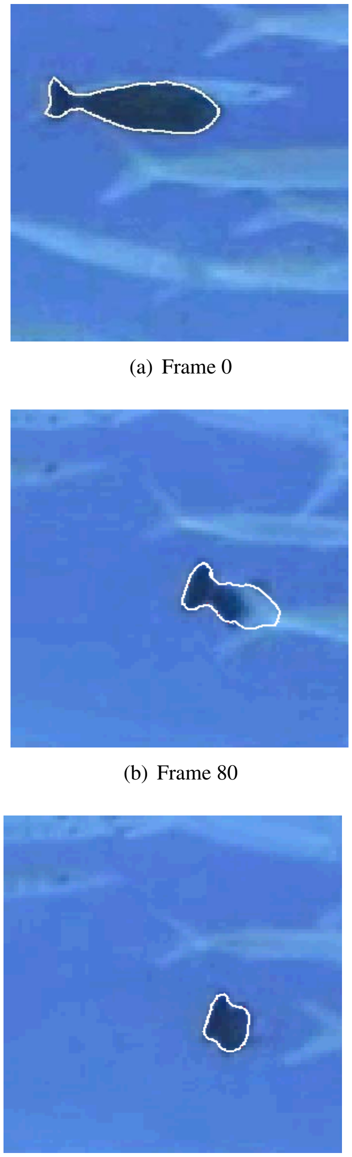
$$\begin{aligned} a(\boldsymbol{x}) &:= \arccos\left(\frac{\nabla(G * I)}{\|\nabla(G * I)\|} \cdot \mathcal{N}\right) \\ a_n(\boldsymbol{x}) &:= \min\left(|a(\boldsymbol{x})|, \pi - |a(\boldsymbol{x})|\right). \end{aligned}$$

This is a measure of angle difference between edge orientation at correspondence points and the normal of the curve. Ideally both should be aligned. The likelihood for a contour point candidate $\boldsymbol{z} \in Z$ is then computed as

$$m(\boldsymbol{z}) = e^{-\left(\frac{(|d(\boldsymbol{z})|-\mu_d)^2}{2\sigma_d^2} + \frac{(g(\boldsymbol{z})-\mu_g)^2}{2\sigma_g^2} + \frac{(a_n(\boldsymbol{z})-\mu_a)^2}{2\sigma_a^2}\right)},$$

and the occlusion detection of Section (6) is performed.

In both cases occlusions are handled. For the fish sequence the occlusion is dealt with implicitly. The occluding fish moves over the tracked fish quickly, so that the inertia effects keep the fish at a reasonable location. For the car example the occlusion (the lamp post) is treated explicitly by means of the proposed occlusion detection algorithm. In both cases the likelihood functions do not incorporate any type of prior movement information. Doing so would increase robustness, but limit flexibility. Finally, since this active contour model is edge-based, the snake captures the sharp edge of the shadow in the car sequence. Presumably this could be handled by including more global area-based terms or shape information in the model.



(a) Frame 0

(b) Frame 80

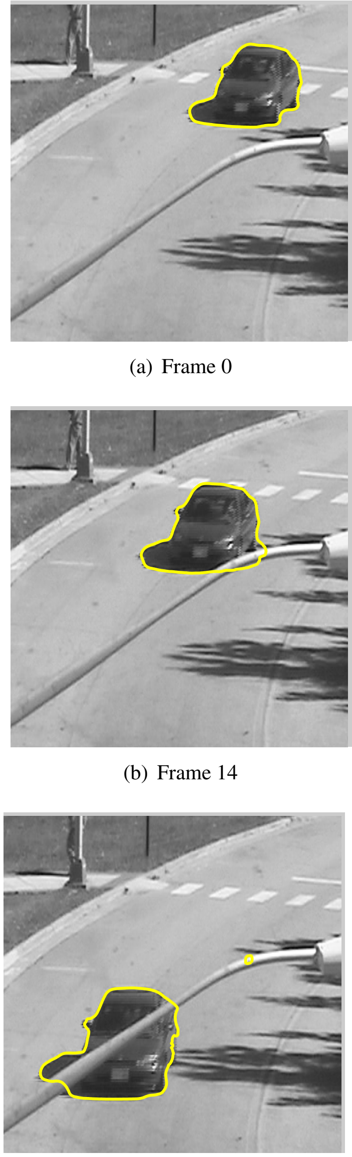
(c) Frame 90

Figure 2. Three frames of a fish sequence.

8. Summary and Conclusions

In this paper we proposed a new approach for visual tracking based on dynamic geodesic snakes. This methodology incorporates state information (here, normal velocity, but any other kind of state information can be treated in a similar way) with every particle on a contour described by means of a level set function. It facilitates a natural framework to combine velocity and position estimation within a level set framework. It has the potential to deal with partial occlusions.

[4] This is simply a monotonic function that increases like a sigmoid up to $x = p_1$, linearly increases for $x \in (p_1, p_2]$, linearly decreases to zero for $x \in (p_2, p_3]$ and is zero everywhere else.



(a) Frame 0

(b) Frame 14

(c) Frame 55

Figure 3. Three frames of a car sequence.

References

[1] F. Bethuel and J.-M. Ghidaglia. *Geometry in Partial Differential Equations*, chapter 1, pages 1–17. World Scientific Publishing Co., 1994.

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

[3] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[4] J. Gomes and O. Faugeras. Shape representation as the intersection of $n - k$ hypersurfaces. Technical Report 4011, INRIA, 2000.

[5] J. Gomes, O. Faugeras, and M. Kerckhove. Using the vector distance functions to evolve manifolds of arbitrary codimension. In *Scale-Space and Morphology in Computer Vision*, volume 2106 of *Lecture Notes in Computer Science*, pages 1–13, 2001.

[6] M. Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26:285–314, 1987.

[7] S. Haker, G. Sapiro, and A. Tannenbaum. Knowledge-based segmentation of SAR data with learned priors. *IEEE Transactions on Image Processing*, 9(2):299–301, 2000.

[8] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: From phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134(3):275–301, 1996.

[9] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. On the evolution of curves via a function of curvature. i. the classical case. *Journal of Mathematical Analysis and Applications*, 163(2):438–458, 1992.

[10] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[11] M. Niethammer and A. Tannenbaum. Dynamic level sets for visual tracking. In *Proceedings of the Conference on Decision and Control*. IEEE, 2003.

[12] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003.

[13] N. Paragios and R. Deriche. Geodesic active regions: A new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13:249–268, 2002.

[14] N. Peterfreund. Robust tracking of position and velocity with Kalman snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):564–569, 1999.

[15] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[16] D. Terzopoulos and R. Szeliski. *Active Vision*, chapter Tracking with Kalman Snakes, pages 3–20. MIT Press, 1992.

[17] J. L. Troutman. *Variational Calculus and Optimal Control*. Springer Verlag, 2nd edition, 1996.

[18] C. Xu, A. Yezzi, and J. L. Prince. On the relationship between parametric and geometric active contours. In *Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 483–489, 2000.

[19] A. Yezzi, A. Tsai, and A. Willsky. Medical image segmentation via coupled curve evolution equations with global constraints. In *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 12–19, 2000.

[20] A. Yezzi, A. Tsai, and A. Willsky. A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13:195–216, 2002.