Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003

Dynamic level sets 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

In this paper we describe two methods for tracking planar curves which are allowed to change topology. In contrast to previous approaches a level set formulation is used that allows for the propagation of state information (here a velocity vector) with every point on a curve. The curve dynamics are derived by minimizing an action integral (based on Hamilton's principle). Incorporating velocity information for every point on a curve lifts the originally two dimensional problem to four dimensions, and thus to a codimension three problem. Since basic level set approaches implicitly describe codimension one hypersurfaces, we introduce two methods suitable for codimension three problems within a level set framework. The partial level set approach, which propagates velocity information along with the curve by solving two additional transport equations, and the full level set approach, which is formulated by means of a vector distance function evolution equation. The full level set approach allows for complete topological flexibility (including intersecting curves in the image plane). However, it is computationally expensive. The partial level set approach compromises the topological flexibility for computational efficiency. In particular, the full level set approach has the potential for tracking objects throughout occlusions, when combined with a suitable collision detection algorithm.

1 Introduction

Visual tracking is a key task in controlled active vision. Fast, reliable segmentation algorithms are particularly important. While the segmentation of objects in still images is in itself quite challenging, adding temporal information to the segmentation problem introduces new difficulties. Various approaches for visual tracking exist: particle filtering [1], deformable templates [2], and active contours (e.g. snakes) as introduced by Kass *et al.* [3], to name but a few.

We distinguish two different methods for active contour based visual tracking: the static and the dynamic approaches. In the static approach the contour is oblivious of its own state; no velocity information is propagated. Tracking can then be achieved by solving subsequent static problems [3] (assuming that the movements of the object to be tracked are relatively slow and the contour does not leave the object's capture range from one frame to the next), or by incorporating temporal information (e.g. optical flow) into an energy functional to be minimized [4, 5]. The dynamic approach is based on a dynamical systems perspective, where points on the contour possess an inherent kinetic energy. They are typically associated with a mass, a velocity vector, are related to their neighboring points through elasticity and rigidity constraints, and move based on an underlying potential field. Terzopoulos and Szeliski [6] derive the equations of motion based on a parametric active contour model, and give a formulation for a Kalman filter based implementation. Peterfreund [7] augments this approach by optical flow measurements to steer the curve evolution, and proposes a way to handle occlusions. Note that using the (extended) Kalman filter (as in [6, 7]) is computationally demanding for a large number of marker particles, due to the need to solve a differential matrix Riccatti equation of dimension $4n \times 4n$ (where n is the number of particles), to determine the state covariance matrix at every time instant.

A huge advantage of static approaches is that they can easily be implemented in a level set framework, thus naturally allowing for topological changes and providing a numerically stable, Eulerian solution method, where the curve is represented as the zero level set of a two-dimensional manifold in $\mathbb{R}^3$ [8] (typically evolving a signed distance function). The dynamic approaches [6, 7] are marker particle based (Lagrangian), due to the restriction of the basic level set methodology to the representation of codimension one objects, whereas the problem in this paper is of codimension three.

Recent work on level sets extends the approach to capture objects of arbitrary codimension. Ambrosio and Soner [9] propose to surround the evolving surface of codimension k in $\mathbb{R}^n$ by a family of hypersurfaces (all of them being level sets). Lorigo *et al.* [10] utilize this method to segment blood vessels in magnetic resonance

0-7803-7924-1/03/$17.00 ©2003 IEEE

angiography images. This is a "natural" application for the theory developed by Ambrosio and Soner, since objects are represented by "hyper-tubes". However, extracting the actual position of the codimension k surface is nontrivial, since it is not explicitly represented.

Instead, Bertalmio *et al.* [11] perform region tracking on a two-dimensional manifold in $\mathbb{R}^3$ by intersecting two hypersurfaces represented as level sets of two level set functions. Similarly Osher *et al.* [12] model planar wavefronts of geometric optics, as objects in three-dimensional space (codimension two), and are thus able to deal with self intersections of the wavefronts. The main complication for these approaches is the initialization of the level set functions. Classically, these are signed distance functions, where the zero level set describes the hypersurfaces. However, an initialization based on signed distance functions is not unique (different sets of hypersurfaces can have the same intersection). Especially at a distance from the intersection it is not clear how the level sets should be extended.

To cope with this problem Gomes *et al.* [13] evolve vector distance functions to implicitly move manifolds of arbitrary dimension. This amounts to the intersection of n hypersurfaces in an n-dimensional space. The description is thus redundant, but does not suffer from initialization problems, since the description of a manifold in terms of a vector distance function is unique.

Note, that evolutions based on vector distance functions or on the intersection of multiple level sets are computationally expensive. Typically, the evolution has to be performed over the complete computational domain to be able to deal naturally with topological changes. Specifically, the use of narrow banding techniques (where the equations are only solved within a small band around the manifold to be evolved; see for example [8]) is generally not possible.

We will thus introduce two different tracking methods in this paper: the partial, and the full level set approach. The full level set approach is based on a vector distance function evolution. It possesses full topological flexibility, and can deal with curve intersections in the image plane, but is computationally expensive. Note, that handling self intersecting objects is especially important for tracking moving objects which temporarily occlude each other. The partial level set approach gives up the topological flexibility for increased computational efficiency (it cannot represent curve intersections); it allows for a narrow banding implementation.

The paper is organized as follows. Section 2 describes the basics of dynamic parametric curve evolution, and the dynamic geometric curve evolution model. A controls perspective is given, and the equations of motion are derived. Section 3 describes the two level set approaches, and Section 4 how to deal with occlusions. Conclusions are given in Section 5.

2 Dynamic curve evolution

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 $\mathcal{C}(0, t) = \mathcal{C}(1, t)$ [14, 15], with t being the time, and $p \in [0, 1]$ the curve's parameterization. The classical formulation for dynamic curve evolution as proposed by Terzopoulos and Szeliski [6] is derived by means of minimization of the *action integral*

$$\mathcal{L} = \int_{t=t_0}^{t_1} \int_{p=0}^{1} L(t, p, \mathcal{C}, \mathcal{C}_p, \mathcal{C}_{pp}, \mathcal{C}_t)\ dp\ dt, \qquad (1)$$

where the subscripts denote partial derivatives with respect to the time t and the parameterization p. The Lagrangian $L = T - U$ is the difference between the kinetic and the potential energy. The potential energy of the curve is given by

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

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

$$g(\boldsymbol{x}) = \frac{1}{1 + \|G_\sigma * \nabla I(\boldsymbol{x})\|^r}, \qquad (2)$$

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

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

where μ corresponds to mass per unit length. 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 [16] in force balance form[1]:

$$\mu \mathcal{C}_{tt} = \frac{\partial}{\partial p}(w_1 \mathcal{C}_p) - \frac{\partial^2}{\partial p^2}(w_2 \mathcal{C}_{pp}) - \nabla g. \qquad (3)$$

Note, that the right hand side of equation (3) corresponds to a force (compare with Newton's second law $ma = F$, where m is the mass, a the acceleration, and F the force), where ∇g is the force exerted by the image on the curve. The capture range of the potential force ∇g will depend on the variance of the Gaussian

[1]Note, that to dampen the motion of the curve, a Rayleigh dissipation functional can be introduced [6].

in equation (2). If a greater capture range is desired, ∇g can be replaced by a more general force, the gradient vector flow [17]. This formulation is not intrinsic with respect to the geometry of the curve, since it depends on the parameterization p (see Xu *et al.* [18] for a discussion on the relationship between parametric and geometric active contours).

Minimizing equation (1) using the Lagrangian

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

yields

$$\mu\mathcal{C}_{tt} = -\mu(\mathcal{C}_t \cdot \mathcal{C}_{ts})\mathcal{T} - \mu(\mathcal{T} \cdot \mathcal{C}_{ts})\mathcal{C}_t - \frac{1}{2}\mu\|\mathcal{C}_t\|^2\kappa\mathcal{N} + \\ + g\kappa\mathcal{N} - (\nabla g \cdot \mathcal{N})\mathcal{N}, \quad (4)$$

which is intrinsic. 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 [19]. The second line of equation (4) corresponds to the force exerted by the image potential g on the curve $\mathcal{C}$ (compare this to the evolution equation for geodesic active contours as given in [20, 14]). From a controls perspective this can be interpreted as a control law based on g and its spatial gradient ∇g, designed to move the curve close to the potential well (where g and ∇g are small).

The curve tries to smooth out ($g\kappa\mathcal{N}$), while moving in the direction of the potential well ($(\nabla g \cdot \mathcal{N})\mathcal{N}$). The force required for dynamical smoothing of the curve is proportional to the square of the velocity at each point ($\|\mathcal{C}_t\|^2\kappa\mathcal{N}$).

Note that the control law will not guarantee perfect tracking, since the potential forces associated with g will have to outweigh the dynamical forces. As a remedy PI control (optionally with an anti-windup scheme) of the form

$$\mu\mathcal{C}_{tt} = -\mu(\mathcal{C}_t \cdot \mathcal{C}_{ts})\mathcal{T} - \mu(\mathcal{T} \cdot \mathcal{C}_{ts})\mathcal{C}_t - \frac{1}{2}\mu\|\mathcal{C}_t\|^2\kappa\mathcal{N} + \\ + \left(\alpha_p g + \alpha_i \int g\ dt\right)\kappa\mathcal{N} - \\ - \left((\mathrm{diag}(\beta_p)\nabla g + \mathrm{diag}(\beta_i)\int \nabla g\ dt) \cdot \mathcal{N}\right)\mathcal{N} \quad (5)$$

can be used. Where α_p, α_i, β_p, β_i are controller parameters, and $\mathrm{diag}(\boldsymbol{\beta})$ denotes the diagonal matrix with $\mathrm{diag}(\boldsymbol{\beta})_{kk} = \beta_k$. The PI control law adds three new states (for the integrators) to the evolution equation. This is an evolution in $\mathbb{R}^7$ and a problem of codimension six. For the sake of simplicity we will continue to use equation (4) in what follows.

Equations (3,4,5) describe a curve evolution that is only influenced by inertia terms, and information on the curve itself. To increase robustness the potential energy U can include region-based terms (see for example [4, 21, 22]). This would change the evolution equations (3,4,5), but poses no problems to our proposed level set approaches.

The state-space form of equation (4) is

$$\boldsymbol{x}_t(s,t) = \left(x_3(s,t), \quad x_4(s,t), \quad f_1(\boldsymbol{x}), \quad f_2(\boldsymbol{x})\right)^T, \quad (6)$$

where $\boldsymbol{x}^T = [x_1, x_2, x_3, x_4]^T$, $x_1 = x(s,t)$, $x_2 = y(s,t)$, $x_3 = x_t(s,t)$, $x_4 = y_t(s,t)$, and f_i are scalar functions in $\boldsymbol{x}$ and its derivatives. The evolution describes the movement of a curve in $\mathbb{R}^4$, where the geometrical shape can be recovered by the simple projection

$$\Pi(\boldsymbol{x}) = \left(x_1(s,t), \quad x_2(s,t)\right)^T.$$

3 Level set approaches

To implement the curve evolution equation (6) of Section 2 we propose two different approaches. The full level set approach propagates the curve in four-dimensional space, where the curve is implicitly described by the intersection of three hypersurfaces or the zero level set of a vector distance function. The second methodology, the partial level set approach, uses a level set formulation for the propagation of an implicit description of the curve in the image plane itself (thus allowing for topological changes), but explicitly propagates the velocity information associated with every point on the contour by means of two transport equations.

3.1 Partial level set approach

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

$$\Phi(\boldsymbol{x}(t), t) : \mathbb{R}^2 \times \mathbb{R}^+ \mapsto \mathbb{R}, \quad (7)$$

where $\boldsymbol{x}(t) = [x(t), y(t)]$ is a point in the image plane. Typically, Φ is initialized as the signed distance function to the curve $\mathcal{C}$. Taking the time derivative of equation (7) for the zero level set results in

$$\Phi_t + \nabla\Phi \cdot \boldsymbol{x}_t = 0. \quad (8)$$

This is a two dimensional transport equation, with $\boldsymbol{x}_t$ being the speed of the points on the contour. If the speed $\boldsymbol{x}_t$ for each point on the contour is known, the solution of equation (8) guarantees, that the movement of the zero level set of Φ corresponds to the desired movement of the curve[2]. Note, that generally curves are only moved in their normal directions (i.e., $\nabla\Phi \cdot \boldsymbol{x}_t$ gets

[2]For the evolution of the level set function Φ the velocity vector $\boldsymbol{x}_t$ has to be defined on the complete domain. Since equation (6) only gives the velocities on the curve itself, extension velocities have to be constructed [8]. This is true for all the evolution equations in this paper. We assume that extension velocities have been computed if required.

replaced by $\|\nabla\Phi\|\mathcal{N}\cdot\boldsymbol{x}_t$), because the tangential component of the evolution law only changes the parameterization of the curve, but not its geometrical shape (see [20] for a proof). The position of our curve will thus be described for all times by

$$\Phi_t + \|\nabla\Phi\|\mathcal{N}\cdot\boldsymbol{x}_t = 0.$$

To keep track of the velocities of points on the contour, the two functions $u : \mathbb{R}^2 \times \mathbb{R}^+$ and $v : \mathbb{R}^2 \times \mathbb{R}^+$, representing the x and the y components of the velocity vector respectively ($\boldsymbol{x}_t = [u, v]^T$), are propagated along with the zero level set of Φ by solving two additional transport equations. Specifically the following algorithm is proposed for numerical implementations[3]

1) Compute the current velocities at every point of the contour based on equation (6).
2) Update the velocity fields u and v using the results from step 1.
3) Propagate Φ_t, u, and v by one time step using the velocities from step 1. For u and v this amounts to solving

$$\begin{aligned} u_t + \boldsymbol{x}_t \cdot \nabla u &= 0 \\ v_t + \boldsymbol{x}_t \cdot \nabla v &= 0 \end{aligned}$$

respectively. Note, that it is important to propagate u and v in the direction $\boldsymbol{x}_t$ opposed to the normal direction with respect to the curve in this case.

It is immediately clear, that this approach can be used to propagate any kind of information along with the contour (see [25] on how to propagate material quantities). By distributing marker particles on the contour one could use this method for region tracking (compare Bertalmio and Sapiro [11] where region tracking is performed by intersecting two hypersurfaces – a full level set approach). The advantage in comparison to full level set approaches is the reduction in computational complexity, since it allows for a narrow-band implementation. Specifically, the computational complexity is only about three times the complexity of a normal narrow band level set implementation for curve propagation in the plane. However, the approach cannot handle intersections of contours, and so contours that would lie on top of each other will necessarily be merged.

3.2 Full level set approach

Unlike the partial level set approach of subsection 3.1, a full level set approach evolves a completely implicit representation of the curve based on equation (6). This allows for full topological flexibility: of specific interest in this paper is hereby the property that two curves will only be merged at a point if their positions *and* velocities at this point are identical, i.e. the methodology allows for curves to slide over each other if this is what their dynamical description requires them to do. The projection on the image plane will then show curves intersecting each other.

Moving a curve in $\mathbb{R}^4$ is a codimension three problem (the difference in dimension between the embedding space and the object to be represented is three). Typically the level set formulation is used to represent codimension one objects (a curve in $\mathbb{R}^2$, a surface in $\mathbb{R}^3$, etc.), due to the fact that isocontours of a function defined over a d dimensional space will generally be objects of dimension $d-1$.

A straightforward approach to the evolution of a codimension k object is thus to evolve the intersection of isocontours of k level set functions in a way consistent with the desired movement of the codimension k object (see [26, 23, 11] for codimension two, [12] for codimension three, and [27] for arbitrary codimensions).

Given the three level set functions $\alpha : \mathbb{R}^4 \mapsto \mathbb{R}$, $\beta : \mathbb{R}^4 \mapsto \mathbb{R}$, $\gamma : \mathbb{R}^4 \mapsto \mathbb{R}$ this amounts to the evolution of

$$\begin{aligned} \alpha_t + \boldsymbol{x}_t \cdot \nabla\alpha &= 0 \\ \beta_t + \boldsymbol{x}_t \cdot \nabla\beta &= 0 \\ \gamma_t + \boldsymbol{x}_t \cdot \nabla\gamma &= 0, \end{aligned}$$

where $\boldsymbol{x}_t$ is the velocity vector as given by equation (6). The intersection of the zero level sets of α, β, and γ represents the current state of the curve. Two major questions arise from this formulation:

1) How are the level set functions initialized? This has to be performed globally (over the whole computational domain), if well behaved handling of topological changes is required (narrow-banding approaches cannot be used *per se* for this problem). Note that the representation of an object as the intersection of multiple hypersurfaces is not unique.
2) How can the speed functions in the complete computational domain be determined for the three level set functions α, β, γ? Note that the speed function is only known at the intersection of the hypersurfaces. Based on these velocities, extension velocities have to be constructed (on the zero level sets, and in the interior of the domain).

To deal with these problems a novel approach based on vector distance functions is introduced in [28, 13]. Given a manifold $\mathcal{M}$ (in our case the closed curve in $\mathbb{R}^4$)

$$\delta(\boldsymbol{x}) := \operatorname{dist}(\boldsymbol{x}, \mathcal{M}),$$

is defined as the distance from point $\boldsymbol{x} \in \mathbb{R}^4$ to the manifold $\mathcal{M}$. The vector distance function $\boldsymbol{u}(\boldsymbol{x})$ is then

[3] Note, that this implementation needs to be tied into a whole numerical scheme. This is beyond the scope of this paper. See [8, 23, 24] for details.

given as the derivative of the squared distance function

$$\eta(\boldsymbol{x}) = \frac{1}{2}\delta^2(\boldsymbol{x}).$$

Thus

$$\boldsymbol{u}(\boldsymbol{x}) = \nabla\eta(\boldsymbol{x}) = \delta(\boldsymbol{x})\nabla\delta(\boldsymbol{x}).$$

The vector distance function is an implicit representation of the manifold $\mathcal{M}$ with

$$\mathcal{M} = \boldsymbol{u}^{-1}(\boldsymbol{0}).$$

This amounts to the intersection of the n hypersurfaces

$$\boldsymbol{u}_i = 0, i = 1, \ldots, n.$$

The description is redundant, but unique.

For the case of curve evolution in $\mathbb{R}^4$ (as specified by equation (6)) the evolution equation for the one dimensional manifold $\mathcal{M}(p,t)$, parameterized by p, becomes

$$\mathcal{M}_t(p,t) = \Pi^N_{\mathcal{M}(p,t)}(\mathcal{D}(\mathcal{M}(p,t),t)) = \boldsymbol{V}(\mathcal{M}(p,t),t),$$

where $\mathcal{D}(\boldsymbol{x},t)$ is a vector field defined on $\mathbb{R}^4 \times \mathbb{R}^+$. At any time $\mathcal{D}(\boldsymbol{x},t) = \boldsymbol{x}_t$ of equation (6) on the curve. $\Pi^N_{\mathcal{M}(p,t)}$ is a projection operator, that projects the velocity $\boldsymbol{x}_t$ defined on the curve, into the curve's normal space. Note that in this case the tangential component is of no importance to the evolution equation, since the curve evolution is performed in $\mathbb{R}^4$.

To evolve the manifold $\mathcal{M}$, a speed function has to be constructed in the subspace of $\mathbb{R}^4$, that will contain the evolution of the curve (i.e. the domain will be limited based on the image dimensions, and the expected velocities). This speed function should

1) maintain the vector distance function throughout the evolution,
2) move the manifold $\mathcal{M}$ as desired.

It can be shown (see [13]) that the characteristic equation for the vector distance function $\boldsymbol{u}(\boldsymbol{x})$ is

$$(D\boldsymbol{u})^T\boldsymbol{u} = \boldsymbol{u}, \tag{9}$$

where $D\boldsymbol{u}$ denotes the Jacobian of $\boldsymbol{u}$. Taking the time derivative of equation (9) and using the fact that $(D\boldsymbol{u})^T = D\boldsymbol{u}$ yields

$$D\boldsymbol{b}\boldsymbol{u} = (\boldsymbol{I} - D\boldsymbol{u})\boldsymbol{b},$$

where $\boldsymbol{b}$ is the desired velocity for the vector distance function evolution with initial condition

$$\boldsymbol{b}(\mathcal{M},t) = -\boldsymbol{V}(\mathcal{M},t).$$

The overall evolution is then given by

$$\boldsymbol{u}_t(\boldsymbol{x},t) = \boldsymbol{b}(\boldsymbol{x},t),$$

which completes our full level set tracking algorithm.

4 Occlusion handling

The full level set approach is capable of describing intersecting curves in the image plane. While the partial level set approach of Section 3.1 by construction merges objects once they start touching each other in the image plane, the full level set approach does not, unless the velocities match at the point of contact. This is a powerful property to deal with objects occlusions, and the main advantage of the full level set approach over the partial level set approach.

Assume we are given a function $O : \mathbb{R}^2 \times \mathbb{R}^+ \mapsto \mathbb{R}$, defined on $\mathcal{C}$ for any time t, with $O(\boldsymbol{x},t) = 0$ if a point on a curve is occluded, and $O(\boldsymbol{x},t) = 1$ if it is not (we can allow intermediate values to express occlusion probabilities).

Changing equation (4) to

$$\mu\mathcal{C}_{tt} = O(\mathcal{C},t)\cdot(-\mu(\mathcal{C}_t\cdot\mathcal{C}_{ts})\mathcal{T} - \mu(\mathcal{T}\cdot\mathcal{C}_{ts})\mathcal{C}_t - \\ - \frac{1}{2}\mu\|\mathcal{C}_t\|^2\kappa\mathcal{N} + g\kappa\mathcal{N} - (\nabla g\cdot\mathcal{N})\mathcal{N}), \tag{10}$$

will then propagate possibly occluded points solely based on their associated velocity and unaffected by smoothness constraints or underlying image information, as long as they stay occluded. Peterfreund [7] compares image intensities to predicted image intensities to detect occlusions. Yue *et al.* [29] use a contour prediction scheme, and Niyogi [30] performs a spatiotemporal junction analysis to detect motion boundaries. For the full level set approach the geometric intersections of curves in the image plane could be used for occlusion detection, if all the moving objects in an image sequence are being tracked. Details will be discussed in a future paper.

5 Conclusion

In this paper we discussed level set approaches for codimension three curve propagation in the context of visual tracking. We described two methods, the partial and the full level set approaches, to facilitate the propagation of velocity information with every point on a curve. In contrast to previous work, both methods are Eulerian and allow for topological changes. The restriction of the partial level set approach compared to the full level set approach is its inability to represent intersecting curves. Its advantage is its relatively low computational cost. Note that in comparison with particle filtering (condensation) the proposed algorithms do not require learning steps before tracking can be accomplished. Furthermore, curves are allowed to deform arbitrarily, which would lead to very large sampling spaces, and thus to a decrease in performance, for the particle filtering method. The dynamic approach allows (due to inertia effects of the propagating curves) for noise rejection, and short temporal occlusions. The curves will simply slide over small or short-time disturbances. This is the main advantage in comparison with static methods, where the curves are propagated from frame to frame (maybe even using velocity information, such as optical flow), followed by a refinement procedure (curve evolution on a static image). In particular the handling of curve intersections of the full level set approach can be of great value in dealing with object occlusions, when combined with a suitable collision detection algorithm. Upon collision detection, the affected points can simply be propagated forward in time, based on their current velocity information (unaffected by the underlying image or smoothness constraints). Note, that we can show for the special action integral considered in this paper that the curve's movement can be restricted to its normal direction in the absence of initial tangential velocities. Then the normal velocity obeys a hyperbolic conservation law. We will explore this in the full version of the paper [31].

References

[1] A. Blake and M. Isard, *Active Contours.* Springer Verlag, 1998.

[2] A. Yuille and P. Hallian, *Active Vision.* MIT Press, 1992, ch. Deformable Templates, pp. 21–38.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, pp. 321–331, 1988.

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

[5] G. Unal, A. Yezzi, and H. Krim, "Fast incorporation of optical flow into active polygons," preprint, submitted to *IEEE Transactions on Image Processing*, 2003.

[6] D. Terzopoulos and R. Szeliski, *Active Vision.* MIT Press, 1992, ch. Tracking with Kalman Snakes, pp. 3–20.

[7] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

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

[9] L. Ambrosio and M. Soner, "Level set approach to mean curvature flow in arbitrary codimension," *Journal of Differential Geometry*, vol. 43, pp. 693–737, 1996.

[10] L. M. Lorigo, O. Faugeras, W. E. L. Grimson, R. Keriven, R. Kikinis, and C.-F. Westin, "Co-dimension 2 geodesic active contours for MRA segmentation," in *Proceedings of the International Conference on Information Processing in Medical Imaging*, 1999, pp. 126–139.

[11] M. Bertalmio, G. Sapiro, and G. Randall, "Region tracking on level-sets methods," *IEEE Transactions on Medical Imaging*, vol. 18, no. 5, pp. 448–451, 1999.

[12] S. Osher, L. Cheng, M. Kang, H. Shim, and Y. Tsai, "Geometric optics in a phase space based level set and Eulerian framework," *Journal of Computational Physics*, vol. 179, no. 2, pp. 622–648, 2002.

[13] 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*, ser. Lecture Notes in Computer Science, vol. 2106, 2001, pp. 1–13.

[14] A. Tannenbaum, "Three snippets of curve evolution theory in computer vision," *Mathematical and Computer Modelling Journal*, vol. 24, pp. 103–119, 1996.

[15] B. Kimia, A. Tannenbaum, and S. Zucker, "Shapes, shocks, and deformations, i: the components of shape and the reaction-diffusion space," *International Journal of Computer Vision*, vol. 15, pp. 189–224, 1995.

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

[17] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[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*, vol. 1, 2000, pp. 483–489.

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

[20] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis.* Cambridge University Press, 2001.

[21] 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*, vol. 13, pp. 195–216, 2002.

[22] ——, "Medical image segmentation via coupled curve evolution equations with global constraints," in *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 2000, pp. 12–19.

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

[24] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics, 2002.

[25] D. Adalsteinsson and J. A. Sethian, "Transport and diffusion of material quantities on propagating interfaces via level set methods," *Journal of Computational Physics*, vol. 185, no. 1, pp. 271–288, 2003.

[26] P. Burchard, L. Cheng, B. Merriman, and S. Osher, "Motion of curves in three spatial dimensions using a level set approach," *Journal of Computational Physics*, vol. 170, pp. 720–741, 2001.

[27] J. Gomes and O. Faugeras, "Shape representation as the intersection of $n-k$ hypersurfaces," INRIA, Tech. Rep. 4011, 2000. [Online]. Available: http://www.inria.fr/RRRT/RR-4011.html

[28] ——, "Representing and evolving smooth manifolds of arbitrary dimension embedded in R^n as the intersection of n hypersurfaces: The vector distance functions," INRIA, Tech. Rep. 4012, 2000. [Online]. Available: http://www.inria.fr/RRRT/RR-4012.html

[29] Y. Fu, A. T. Erdem, and A. M. Tekalp, "Tracking visible boundary of objects using occlusion adaptive motion snake," *IEEE Transactions on Image Processing*, vol. 9, no. 12, pp. 2051–2060, 2000.

[30] S. A. Niyogi, "Spatiotemporal junction analysis for motion boundary detection," in *Proceedings of the International Conference on Image Processing*, vol. 3. IEEE, 1995, pp. 468–471.

[31] M. Niethammer, S. Angenent, and A. Tannenbaum, "Dynamic tracking and conformal snakes," in preparation.