Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects

Yogesh Rathi Namrata Vaswani Allen Tannenbaum Anthony Yezzi
Georgia Institute of Technology
School of Electrical and Computer Engineering
Atlanta, GA, USA 30332
{yogesh.rathi,tannenba}@bme.gatech.edu,{namrata,ayezzi}@ece.gatech.edu

Abstract

Geometric active contours are formulated in a manner which is parametrization independent. As such, they are amenable to representation as the zero level set of the graph of a higher dimensional function. This representation is able to deal with singularities and changes in topology of the contour. It has been used very successfully in static images for segmentation and registration problems where the contour (represented as an implicit curve) is evolved until it minimizes an image based energy functional. But tracking involves estimating the global motion of the object and its local deformations as a function of time. Some attempts have been made to use geometric active contours for tracking, but most of these minimize the energy at each frame and do not utilize the temporal coherency of the motion or the deformation. On the other hand, tracking algorithms using Kalman filters or Particle filters have been proposed for finite dimensional representations of shape. But these are dependent on the chosen parametrization and cannot handle changes in curve topology. In the present work, we formulate a particle filtering algorithm in the geometric active contour framework that can be used for tracking moving and deforming objects.

1 Introduction

The problem of tracking moving and deforming objects has been a topic of substantial research in the field of Computer Vision; see [1, 30] and the references therein. In this paper, we propose a scheme which combines the advantages of particle filtering and geometric active contours realized via level set models for dynamic tracking.

In order to appreciate this methodology, we briefly review some previous related work. First of all, a number of different representations of shape have been proposed in literature together with algorithms for tracking using such representations. In particular, the notion of *shape* has been found to be very useful in this enterprize. For example, the shape of a set of N discrete points (called *landmarks*) in $\mathbf{R}^M$ is defined [12, 8] as the equivalence class of $\mathbf{R}^{MN}$ under the Euclidean similarity group in $\mathbf{R}^M$. The dynamics of the similarity group defines the global motion while the dynamics of the equivalence class defines the deformation. In [34], the authors define a prior dynamical model on the deformation and on the similarity group parameters. A particle filter [9] is then used to track the deformation and the global motion over time.

The possible parameterizations of shape are of course very important. We should note that various finite dimensional parameterizations of continuous curves have been proposed, perhaps most prominently the B-spline representation used for a “snake model” as in [30]. Isard and Blake (see [1] and references therein) apply the B-spline representation for contours of objects and propose the Condensation algorithm [10] which treats the affine group parameters as the state vector, learns a prior dynamical model for them and uses a particle filter [9] to estimate them from the noisy observations. Since this approach only tracks the affine parameters it cannot handle local deformations of the deforming object (see e.g., the fish example in Section 5.1). One possible solution proposed in [37], is to use deformable templates to model prior shapes allowing for many deformation modes of shapes. Some other approaches to tracking are given in [4, 29, 15].

Another approach for representing contours is via the level set technique [20, 27] which is an implicit representation of contours. For segmenting a shape using level sets, an initial guess of the contour is deformed until it minimizes an image-based energy functional. Different energy functionals which utilize different features of the image have been used in literature; see e.g. [13, 17, 3, 14, 16, 2]. Some previous work on tracking using level set methods is given in

[38, 19, 21, 36, 11].

In [19], the authors use dynamic active contours for tracking. The state space is defined by the contour's position and the deformation velocity. Explicit observations of the contour and the contour deformation velocity are incorporated into the PDE using an error injection technique. The prediction step in this approach is obtained using the principle of least action which uses the current image (and hence the predicted contour velocity is correlated with the observation). This can be a problem if there are some outlier observations (such as occlusions) and hence they define a separate occlusion handling method.

The work in this paper extends the ideas presented in [36, 11]. More precisely, in [36], the authors propose a definition for motion and shape deformation for a deforming object. Motion is parameterized by a finite dimensional group action (e.g. Euclidean or Affine) while shape deformation is the total deformation of the object contour (infinite dimensional group) modulo the finite dimensional motion group. This is called *deformotion*. Tracking is then defined as a trajectory on the finite dimensional motion group. This approach relies only on the observed images for tracking and does not use any prior information on the dynamics of the group action or of the deformation. As a result it fails if there is an outlier observation or if there is occlusion. To address this problem, [11] proposes a generic local observer to incorporate prior information about the system dynamics in the "deformotion" framework. They impose a constant velocity prior on the group action and a zero velocity prior on the contour. The observed value of the group action and the contour is obtained by a joint minimization of the energy. This is linearly combined with the value predicted by the system dynamics using an observer.

This approach suffers from two problems. First, as in [36], they must perform a joint minimization over the group action and the contour at each time step which is computationally very intensive. Second, for nonlinear systems such as the one used in [11], there is no systematic way to choose the observer to guarantee stability. The present paper addresses the above limitations. We formalize the incorporation of a prior system model along with an observation model. A particle filter is used to estimate the conditional probability distribution of the group action and the contour at time t, conditioned on all observations up to time t.

Other approaches closely related to our work are given in [30, 23]. Here the authors use a Kalman filter in conjunction with active contours to track nonrigid objects. The Kalman filter was used for predicting possible movements of the object, while the active contours allowed for tracking deformations in the object.

This paper is organized as follows: In the next section we discuss the particle filter and level set method. In Section 3 we describe the state space model and Section 4 discusses the algorithm in detail. Experimental results are given in Section 5. Limitations and future work are discussed in Section 6.

2 Preliminaries

In this section, we review some basic notions from the theory of level set evolutions and particle filtering which we will need in the sequel.

2.1 Particle Filtering

Let $X_t \in \mathbf{R}^n$ be a state vector evolving according to the following difference equation: $X_{t+1} = f_t(X_t) + u_t$ where u_t is i.i.d. random noise with known probability distribution function (pdf), $p_{u,t}$. At discrete times, observations $Y_t \in \mathbf{R}^p$ become available. These measurements are related to the state vector via the observation equation: $Y_t = h_t(X_t) + v_t$ where v_t is measurement noise with known pdf $p_{v,t}$. It is assumed that the initial state distribution denoted by $\pi_0(dx)$, the state transition kernel denoted by $K_t(X_t, X_{t+1}) = p_{u,t}(X_{t+1} - f_t(X_t))$ and the observation likelihood given the state, denoted by $g_t(Y_t|X_t) = p_{v,t}(Y_t - h_t(X_t))$, are known. The particle filter (PF) [9, 7] is a sequential Monte Carlo method which produces at each time t, a cloud of N particles, $\{X_t^{(i)}\}_{i=1}^N$, whose empirical measure closely "follows" $\pi_t(dx_t|Y_{0:t})$, the posterior distribution of the state given past observations (denoted by $\pi_{t|t}(dx)$ in the rest of the paper). The PF was first introduced in [9] as the Bayesian Bootstrap filter and its first application to tracking in computer vision was the Condensation algorithm [10].

The algorithm starts with sampling N times from the initial state distribution $\pi_0(dx)$ to approximate it by $\pi_0^N(dx) = \frac{1}{N}\sum_{i=1}^N \delta_{X_0^{(i)}}(dx)$ and then *implements the Bayes' recursion* at each time step. Now, the distribution of X_{t-1} given observations upto time $t-1$ can be approximated by $\pi_{t-1|t-1}^N(dx) = \frac{1}{N}\sum_{i=1}^N \delta_{X_{t-1}^{(i)}}(dx)$. The **prediction step** samples the new state $\bar{X}_t^{(i)}$ from the distribution $K_{t-1}(X_{t-1}^{(i)}, .)$. The empirical distribution of this new cloud of particles, $\pi_{t|t-1}^N(dx) = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{X}_t^{(i)}}(dx)$ is an approximation to the conditional probability distribution of X_t given observations upto time $t-1$ (*prediction distribution*). In the **update step**, each particle is weighted in proportion to the likelihood of the observation at time t, Y_t, i.e.

$$w_t^{(i)} = \frac{g_t(Y_t|\bar{X}_t^{(i)})}{\sum_{i=1}^N g_t(Y_t|\bar{X}_t^{(i)})}.$$

$\bar{\pi}_{t|t}^N(dx) = \frac{1}{N}\sum_{i=1}^N w_t^{(i)}\delta_{\bar{X}_t^{(i)}}(dx)$ is then an estimate of $\pi_{t|t}$ (*filtering distribution*). We resample N times with replacement from $\bar{\pi}_{t|t}^N(dx)$ to obtain the empirical estimate

$\pi^N_{t|t}(dx) = \frac{1}{N}\sum_{i=1}^N \delta_{X_t^{(i)}}(dx)$. Note that both $\bar{\pi}^N_{t|t}$ and $\pi^N_{t|t}$ approximate $\pi_{t|t}$ but the resampling step is used because it increases the sampling efficiency by eliminating samples with very low weights.

2.2 Curve Evolution Using Level Sets

Geometric active contours evolving according to edge based and/or region based energy flow are very commonly used for image segmentation. In these methods, starting from an initial estimate, the curve deforms under the influence of various forces until it fits the object boundaries. The curve evolution equation is obtained by reducing an energy E_{image} as fast as possible, i.e., by doing a gradient descent on E_{image}. In general, E_{image} may depend on a combination of image based features and external constraints (smoothness, shape etc) [17, 3]. The level set methods of Osher and Sethian [20, 28] offer a natural and numerically robust implementation of such curve evolution equations. Level sets have the advantage of being parameter independent (i.e. they are implicit representation of the curve) and can handle topological changes naturally.

We now briefly go over the level set representation of a given curve evolution equation. Let $C(p,\tau) : S^1 \times [0,\theta) \rightarrow \mathbf{R}^2$ be a family of curves satisfying the following evolution equation:

$$-\nabla_C E_{image} = \frac{\partial C}{\partial \tau} = \beta N \tag{1}$$

where, τ is an artificial time-marching parameter. The basic idea of the level set approach is to embed the contour as the level set of a graph $\Phi : \mathbf{R^2} \longrightarrow \mathbf{R}$ and then evolve the graph so that this level set moves according to the prescribed flow. In this manner, the level set may develop singularities and change topology while Φ itself remains smooth and maintains the form of a graph. Formulating the correct evolution of Φ amounts to solving

$$\nabla\Phi(C,\tau) \cdot \frac{\partial C}{\partial \tau} + \frac{\partial \Phi}{\partial \tau} = 0\,. \tag{2}$$

so that the level set of interest maintains a constant value as the graph, Φ evolves. Choosing the zero level set of Φ to define C and choosing Φ to be negative inside C and positive outside C, allows us to write $N = -\nabla\Phi / \parallel \nabla\Phi \parallel$. So the level set implementation of (1) becomes:

$$\frac{\partial \Phi}{\partial \tau} = \beta \parallel \nabla\Phi \parallel \tag{3}$$

Finally, given an initial curve, one must generate an initial level set function. A well known scheme [28] is to use a signed distance function.

3 The State Space Model

Let A_t denote the 6-dimensional affine parameter vector (see equation 8) and μ_t denote the contour (represented as the zero level set of Φ) at time t. We propose to use the affine parameters and the contour as the state, i.e. $X_t = [A_t, \mu_t]$ and treat the image at time t as the observation, i.e. $Y_t = \text{Image}(t)$. The prediction step for X_t consists of:

1. predicting the local deformations in the shape of the object
2. predicting the affine motion of the object.

The affine motion prediction in (2) is obtained from the state dynamics for A_t, which is given by a first or second order (constant velocity or acceleration) autoregressive (AR) model on the affine parameters (see Section 4.1). Since the curve is infinite-dimensional, it is difficult to have a prediction model for local shape deformation. Hence, we assume that: *prediction for local shape deformation at time t = local shape deformation at time* $t-1$. This can be realized by doing a gradient descent on the image energy E_{image} (any image dependent energy, for e.g. equation (9)) at time $t-1$:

$$\tilde{C}_t = C_{t-1} = f^L_{CE}(\mu_{t-1}, Y_{t-1}). \tag{4}$$

Thus, the prediction for X_t depends on $X_{t-1} = [A_{t-1}, \mu_{t-1}]$ and the observation at $t-1, Y_{t-1}$. In equation (4), $f^L_{CE}(\mu, Y)$ is given by L iterations of gradient descent:

$$\mu^k = \mu^{k-1} - \alpha^k \nabla_\mu E_{image}(\mu^{k-1}, Y),\ k = 1,2,3,..,L$$

$$\text{with} \quad f^L_{CE}(\mu, Y) = \mu^L, \quad \mu^0 = \mu$$

The choice of L depends on the particular PDE used for doing curve evolution. In our experiments, we found that $L = 4$ was a good choice. If μ_{t-1} is evolved until convergence, one would reach a local minimum of the energy E_{image}. But this is not desirable since the local minimum would be independent of all starting contours in its domain of attraction and would only depend on the observation, Y_{t-1}. Thus the state at time t would loose its dependence on the state at time $t-1$ and this may cause loss of track in cases where the observation is bad. But if μ_{t-1} is evolved only a fixed number of times, it will deviate the contour only a little (in a direction which reduces the energy E_{image} as fast as possible) so that particles with state closer to the true state will have smaller energy than other particles and these will get propagated during the resampling step.

Now, the probability of observation $Y_t = \text{Image}(t)$ given state $X_t = [A_t, \mu_t]$ can be defined as

$$p(Y_t|X_t) \stackrel{\triangle}{=} e^{-E_{image}(\mu_t, Y_t)} \tag{5}$$

4 The Algorithm

Based on the description above, the proposed algorithm can be written as follows:

1. **Prediction Step:**
   Perform L steps of curve evolution for each sample as follows: [1]

   $$\tilde{C}_t^{(i)} = C_{t-1}^{(i)} = f_{CE}^L(\mu_{t-1}^{(i)}, Y_{t-1})$$

   Generate samples $\{\tilde{A}_t^{(i)}, \tilde{\mu}_t^{(i)}\}_{i=1}^N$ using:

   $$\begin{aligned} \tilde{A}_t^{(i)} &= f_{AR}(A_{t-1}^{(i)}, u_{t-1}^{(i)}) \\ \tilde{\mu_t}^{(i)} &= \tilde{A}_t^{(i)}(C_{t-1}^{(i)}) \end{aligned}$$

   Thus we have

   $$\pi(A_t, \mu_t | Y_{1:t-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta_{\tilde{A}_t^{(i)}, \tilde{\mu}_t^{(i)}}(A_t, \mu_t)$$

2. **Update Step:**

   (a) Weight each sample by

   $$w_t^{(i)} = \frac{e^{-E_{image}(\tilde{\mu}_t^{(i)}, Y_t)}}{\sum_{j=1}^{N} e^{-E_{image}(\tilde{\mu}_t^{(j)}, Y_t)}}$$

   Thus we have

   $$\pi(A_t, \mu_t | Y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{\tilde{A}_t^{(i)}, \tilde{\mu}_t^{(i)}}(A_t, \mu_t)$$

   (b) Resample from the above distribution to generate N particles $\{A_t^{(i)}, \mu_t^{(i)}\}$ distributed according to $\pi(A_t, \mu_t | Y_{1:t})$, i.e.

   $$\pi(A_t, \mu_t | Y_{1:t}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta_{A_t^{(i)}, \mu_t^{(i)}}(A_t, \mu_t)$$

3. Go back to the prediction step for $t+1$.

Note : Scaling E_{image} by a constant factor will affect the resampling step. This scaling factor will decide how much one trusts the system model versus the observation model.

We discuss the details of the above algorithm in the following subsections[2].

Remark 1 Note that, one could include the curve evolution equation in the update step once the observation at time t is available. However, this will change the state X_t based on the observation Y_t. Thus, the existing convergence results [7] of the particle filtering estimate of the posterior π_t^N to the true posterior π_t as $N \to \infty$ cannot be applied. We are working on studying how this modification might change the convergence results, if at all.

4.1 The AR model

In the above algorithm f_{AR} could be any suitable prediction function which can model the dynamics of motion of the moving object. Rather than conjuring up a model that is merely plausible, one can learn the dynamics of motion from a training set. This can be done using an autoregressive (AR) model. Below, we describe the second-order AR process in which the affine parameters at a given time depend on two previous time-steps:

$$A_{t+1} - \overline{A} = B_1(A_t - \overline{A}) + B_2(A_{t-1} - \overline{A}) + B_0 w_{t+1} \quad (6)$$

where A_t is the 6 dimensional affine parameter vector (8), B_1, B_2, B_0 are 6×6 matrices learned *a priori*, w_{t+1} is a vector of 6 independent random N(0,1) variables and $\overline{A}$ is the steady state mean of the model. We refer the interested reader to [1] for further details on how to learn these parameter matrices and the advantages of using the second-order model (AR-2) versus the first-order model (AR-1).

4.1.1 Learning Affine Motion

Many approaches [35, 18] have been reported in the literature for finding the affine parameters that relate one image to the other. Most of these methods require a set of feature points to be known before one can find the affine parameters that relate them. In [22] the author proposes a method which does not require feature points to be known, instead only the source and target images are required. The affine transformation that relates the curve $C(t)$ and $C(t-1)$ is given by:

$$C(x, y, t) = C(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t-1)$$

where, m_i are the affine parameters. In order to estimate these parameters, the following quadratic error is to be minimized:

$$\begin{aligned} E(\vec{m}) = \sum_{x,y \in \omega} [&C(x,y,t) - \\ &C(m_1 x + m_2 y + m_5, m_3 x + m_4 y + m_6, t-1)]^2 \end{aligned}$$

which is linearized and then minimized to give

$$\vec{m} = \left[\sum_{x,y \in \omega} \vec{d}\, \vec{d}^T\right]^{-1} \left[\sum_{x,y \in \omega} \vec{d}\, k\right] \quad (7)$$

[1] To find the best contour at time $t-1$, find the MAP estimate, i.e., find the particle with the maximum probability and evolve *only* this contour until E_{image} is minimized or until a user defined criteria is satisfied. This is the best estimate of the position and shape of the object at time $t-1$.

[2] Note that the above algorithm differs from the standard particle filter in that the prediction step is a function of the previous state and also the previous observation.

where the scalar k and the vectors $\vec{d}$, $\vec{m}$ are given as[3]:

$$k = C_t + xC_x + yC_y$$

$$\vec{d}^T = (xC_x \quad yC_x \quad xC_y \quad yC_y \quad C_x \quad C_y)$$
$$\vec{m} = (m_1 \quad m_2 \quad m_3 \quad m_4 \quad m_5 \quad m_6)^T \qquad (8)$$

Derivation details are available in [22]. Once the affine parameter vector $\vec{m}$ is known for the training set, the AR model parameter matrices can be learned as given in [1].

4.2 The Model of Chan and Vese

Many methods [5, 38, 24, 31, 14] which incorporate geometric and/or photometric (color, texture, intensity) information have been shown to segment images robustly in presence of noise and clutter. In the prediction step above, f_{CE} could be any edge based or region based (or a combination of both) curve evolution equation. In our numerical experiments we have used the Mumford-Shah functional [17] as modelled by Chan and Vese [3] to obtain the curve evolution equation, which we describe briefly. We seek to minimize the following energy:

$$\begin{aligned} E_{image} = E_{cv}(c_1, c_2, \Phi) &= \int_\Omega (f - c_1)^2 H(\Phi) dx\, dy \\ &+ \int_\Omega (f - c_2)^2 (1 - H(\Phi))\, dx\, dy \\ &+ \nu \int_\Omega |\nabla H(\Phi)| dx\, dy \end{aligned} \qquad (9)$$

where c_1 and c_2 are defined as:

$$c_1 = \frac{\int f(x,y) H(\Phi) dx\, dy}{\int H(\Phi) dx\, dy}, c_2 = \frac{\int f(x,y)(1 - H(\Phi)) dx\, dy}{\int (1 - H(\Phi)) dx\, dy}$$

$H(\Phi)$ is the Heaviside function defined as:

$$H(\Phi) = \begin{cases} 1 & \Phi \geq 0\,, \\ 0 & else \end{cases} \qquad (10)$$

$f(x, y)$ is the image and Φ is the level set function as defined in Section 2.2 before. The Euler-Lagrange equation for this functional can be implemented by the following gradient descent [3, 17]:

$$\frac{\partial \Phi}{\partial t} = \delta_\epsilon(\Phi) \left[\nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) - (f - c_1)^2 + (f - c_2)^2 \right] \qquad (11)$$

where,

$$\delta_\epsilon(s) = \frac{\epsilon}{\pi(\epsilon^2 + s^2)}$$

4.3 Dealing with Multiple Objects

In principle, the Condensation filter [1] could be used for tracking multiple objects. The posterior distribution will be multi-modal with each mode corresponding to one object. However, in practice it is very likely that a peak corresponding to the dominant likelihood value will increasingly dominate over all other peaks when the estimation progresses over time. In other words, a dominant peak is established if some objects obtain larger likelihood values more frequently. So, if the posterior is propagated with fixed number of samples, eventually, all samples will be around the dominant peak. This problem becomes more pronounced in cases where the objects being tracked do not have similar photometric or geometric properties. We deal with this issue as given in [33] by first finding the clusters within the state density to construct a Voronoi tessalation [25] and then resampling within each Voronoi cell separately as follows:

1. Every step, build an importance function which results in equal number of samples being taken in each Voronoi cell

2. Every N steps, rescale the weights in each cell so that the peak weight is 1.

Other solutions proposed by [26, 29, 15] could also be used in tackling this problem of sample impoverishment.

4.4 Coping with Occlusions

Many active contour models [14, 24, 6] which use shape information have been reported in the literature. Prior shape knowledge is necessary when dealing with occlusions. In particular, in [38], the authors incorporate "shape energy" in the curve evolution equation to deal with occlusions. Any such energy term can be used in the proposed model to deal with occlusions. In numerical experiments we have dealt with this issue in a slightly different way by incorporating the shape information in the update step, (see algorithm step 2) instead of the prediction step, i.e. we calculate the weight for each particle using the following:

$$w^i_{t+1} = \lambda_1 \frac{e^{-E^i_{cv}}}{\sum_{j=1}^N e^{-E^j_{cv}}} + \lambda_2 (1 - \frac{E^i_{dissimilarity}}{\sum_{j=1}^N E^j_{dissimilarity}}) \qquad (12)$$

where $\lambda_1 + \lambda_2 = 1$ and $E_{dissimilarity}$ is the dissimilarity measure $d^2(\Phi_{tp}, \Phi_i)$ as given in [6] by,

$$d^2(\Phi_{tp}, \Phi_i) = \int_\Omega (\Phi_{tp} - \Phi_i)^2 \, \frac{h(\Phi_{tp}) + h(\Phi_i)}{2} \, dx\, dy,$$

$$\text{with} \quad h(\Phi) = \frac{H(\Phi)}{\int_\Omega H(\Phi)\, dx\, dy}$$

[3] Note: the subscripts in this equation denote partial derivatives

where Φ_{tp} and Φ_i are the level set functions of a template shape and the current contour shape respectively and $H(\Phi)$ is the Heaviside function as defined before in (10). The dissimilarity measure gives an estimate of how different any two given shapes (in particular, their corresponding level sets) are. So, higher values of $E_{dissimilarity}$ indicate more dissimilarity in shape. Using this strategy, particles which are closer to the template shape are more likely to be chosen than particles with "occluded shapes" (i.e., shapes which include the occlusion).

5 Experiments

In this section we describe some experiments performed to test the proposed tracking algorithm. Results of applying the proposed method on three image sequences are given below. The model of Chan and Vese [3], as described earlier, was used for curve evolution. Level set implementation was done using narrow band evolution [28]. Learning [1] was performed on images without the background clutter, i.e. on the outlines of the object.

5.1 Fish Sequence

In the fish video, the shape of the fish undergoes sudden deformation as the fish turns or gets partially occluded (see Figure 3, Frames 167, 181). This local shape deformation cannot be modelled using an affine motion model. Hence, such motion is difficult to track using the standard Condensation filter [1]. As can been seen in the images, (Figure 3) the proposed method can robustly track nonrigid deformations in the shape of the fish. Note that, *no* shape information either in curve evolution or in the weighting step was used in tracking this sequence, i.e. we did not use the dissimilarity term specified in Section 4.4. For this test sequence, an AR-1 model [1] was used for affine motion prediction.

5.2 Car Sequence

In this sequence, the car is occluded as it passes through the lamp post. It is unclear if the standard Condensation algorithm will be able to track the car all the way, since the shape of the car (including the shadow) undergoes a change which is not affine. Notice that the shadow of the car moves in a non-linear way from the side to the front of the car. On the other hand, trying to track such a sequence using geometric active contours (for example, (11)) without any "shape energy" gives very poor results as shown in Figure 1. However, using the proposed method and a weighting strategy as described in Section 4.4 the car can be successfully tracked (Figure 2). Note that we used equation (11) for the curve evolution which does not contain any shape term. A second-order autoregressive model was used for f_{AR}.

5.3 Couple Sequence

The walking couple sequence demonstrates multiple object tracking. In general, tracking such a sequence by the standard Condensation method [1] can give erroneous results when the couple come very close to each other or touch each other, since the measurements made for the person on the right can be interpreted by the algorithm as coming from the left. One solution has been proposed in [29]. Our method naturally avoids this problem since it uses "region based" energy E_{cv} (9) and weighting as given in Section 4.4 to find the observation probabilities. To track multiple objects, we used the method described in Section 4.3. Since the number of frames in the video is less (about 22) no dynamical motion model was learnt. This video demonstrates the fact that, the proposed algorithm can track robustly (see Figure 4) even when the learnt model is completely absent.

6 Limitations and Future Work

In this paper, we proposed a particle filtering algorithm for geometric active contours which can be used for tracking moving and deforming objects. The proposed method can deal with partial occlusions and can track robustly even in the absence of a learnt model.

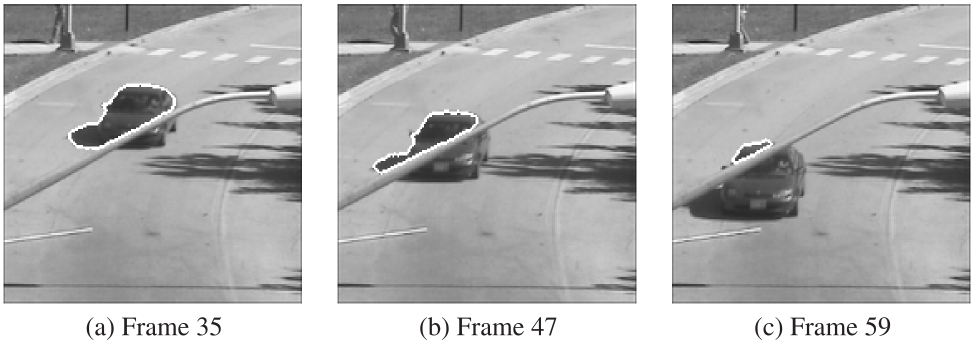
The above framework has several limitations which we intend to overcome in our future work. First, we have to include some kind of shape information when we track objects which undergo major occlusions. This restricts our ability to track highly deformable objects in such situations. Secondly, the algorithm might perform poorly if the object being tracked is *completely occluded* for many frames.

Also, in our current framework the prediction step for the contour is deterministic. We use this model because adding noise to an infinite dimensional representation of the contour is not easy. Nonetheless in [32], the authors have performed PCA on a set of signed distance functions of training shapes to obtain principal directions of variation of the signed distance function for a class of shapes. We can adopt a similar idea and add noise in the principal variation directions. This approach can also provide a shape prior.

References

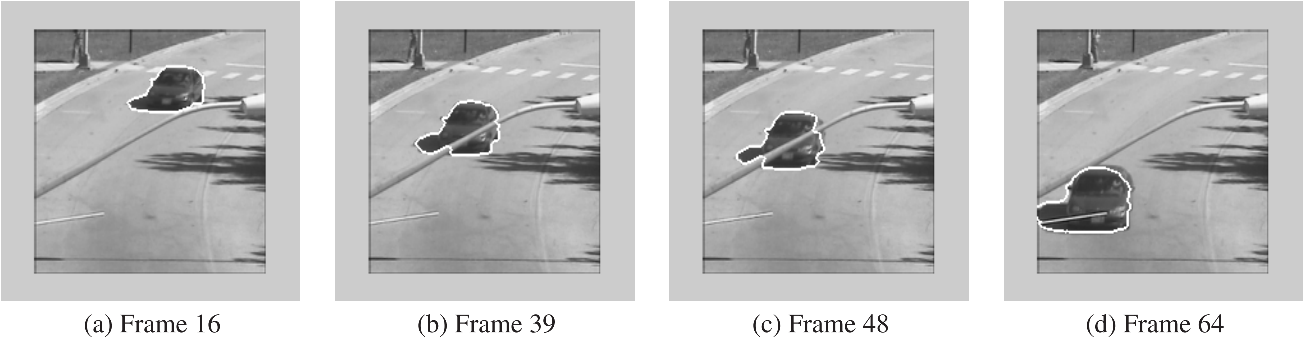
[1] A. Blake and M. Isard, editors. *Active Contours*. Springer, 1998.

[2] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geomteric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.



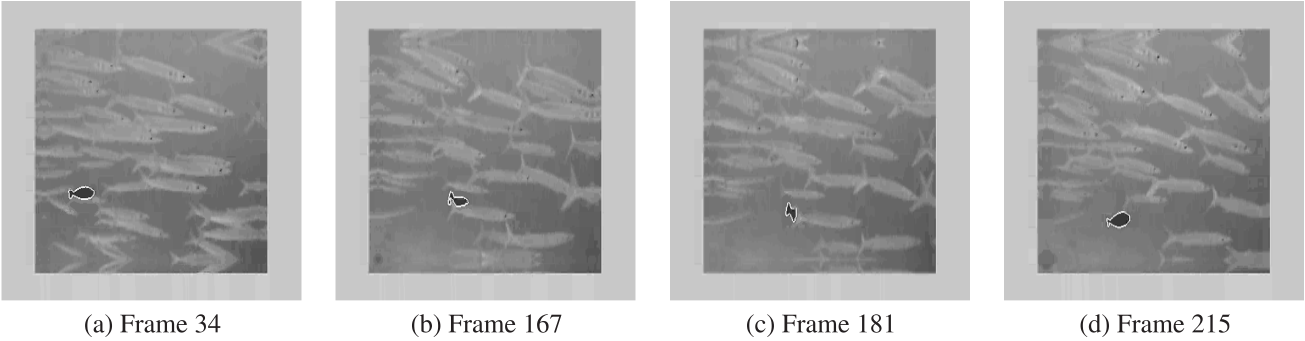
(a) Frame 35 (b) Frame 47 (c) Frame 59

Figure 1. Tracking using equation (11) without particle filter



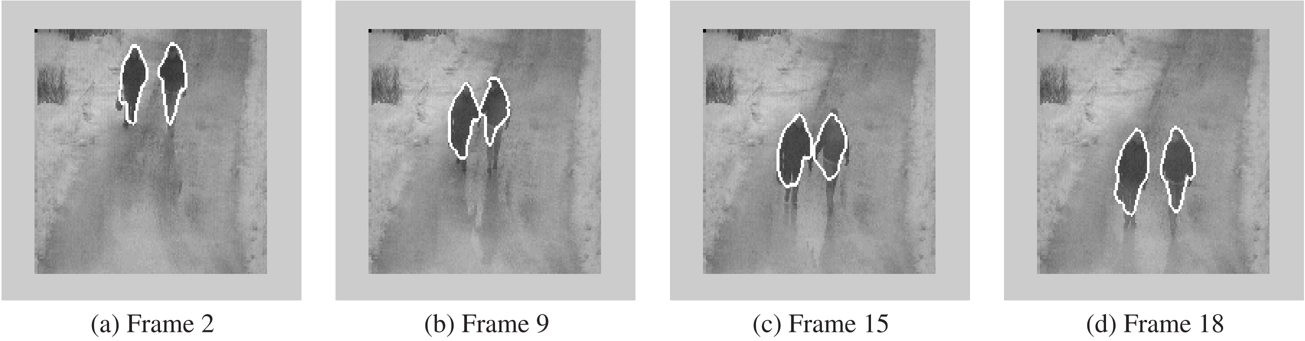
(a) Frame 16 (b) Frame 39 (c) Frame 48 (d) Frame 64

Figure 2. Car Sequence: Number of particles = 50



(a) Frame 34 (b) Frame 167 (c) Frame 181 (d) Frame 215

Figure 3. Fish Sequence: Number of particles = 25



(a) Frame 2 (b) Frame 9 (c) Frame 15 (d) Frame 18

Figure 4. Couple Sequence: Number of particles = 100

[3] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. CVPR*, volume 2, pages 142–149, 2000.

[5] D. Cremers, T. Kohlberger, and C. Schnrr. Nonlinear shape statistics in mumford-shah based segmentation. In *7th ECCV '02*, volume 2351, pages 93–108, 2002.

[6] D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. In *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*. IEEE, 2003.

[7] A. Doucet, N. deFreitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[8] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.

[9] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/nongaussian bayesian state estimation. *IEE Proceedings-F (Radar and Signal Processing)*, pages 140(2):107–113, 1993.

[10] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[11] J. Jackson, A. Yezzi, and S. Soatto. Tracking deformable moving objects under severe occlusions. In *Conf. decision and control*, Dec, 2004.

[12] D. Kendall, D. Barden, T. Carne, and H. Le. *Shape and Shape Theory*. John Wiley and Sons, 1999.

[13] 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.

[14] M. Leventon, W. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. CVPR*, pages 1316–1324. IEEE, 2000.

[15] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39:57–71, 2000.

[16] 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.

[17] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Applied Mathematics*, 42:577–685, 1989.

[18] J. Mundy and A. Z. Eds., editors. *Geometric Invariance in Machine Vision*. MIT Press, 1992.

[19] M. Niethammer and A. Tannenbaum. Dynamic geodesic snakes for visual tracking. In *Proc. CVPR*, volume 1, pages 660–667, 2004.

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

[21] N. Paragois and R. Deriche. Geodesic active contorus and level sets for the detection and tracking of moving objects. *Transactions on Pattern analysis and Machine Intelligence*, 22(3):266–280, 2000.

[22] S. Periaswamy and H. Farid. Elastic registration in the presence of intensity variations. *IEEE Transactions on Medical Imaging*, 2003.

[23] N. Peterfreund. The velocity snake: deformable contour for tracking in spatio-velocity space. *Computer Vision and Image Understanding*, 73(3):346–356, 1999.

[24] M. Rousson and N. Paragois. Shape priors for level set representations. In *Proceedings of European Conference on Computer Vision*, pages 78–92, 2002.

[25] R. Sedgewick. *Algorithms*. Addison-Wesley, 1992.

[26] D. Serby, E. Koller-Meier, and L. V. Gool. Probabilistic object tracking using multiple features. In *International Conference on Pattern Recognition*, 2004.

[27] J. A. Sethian. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed. *J. Differential Geometry*, 31:131–161, 1989.

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

[29] H. Tao, H. Sawhney, and R. Kumar. A sampling algorithm for tracking multiple objects. In *Proceedings of Vision Algorithms, ICCV*, 1999.

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

[31] A. Tsai and A. Yezzi. Model-baesd curve evolution technique for image segmentation. In *Proceedings of Computer Vision and Pattern Recognition*, volume 1.

[32] A. Tsai, T. Yezzi, and W. W. et.al. A shape-based approach to the sementation of medical imagery using level sets. *IEEE Trans. on Medical Imaging*, 22(2):137–153, 2003.

[33] D. Tweed and A. Calway. Tracking many objects using subordinated condensation. In *The British Machine Vision Conference*, pages 283–292, 2002.

[34] N. Vaswani, A. RoyChowdhury, and R. Chellappa. Activity recognition using the dynamics of the configuration of interacting objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, June 2003.

[35] Z. Yand and F. Cohen. Cross-weighted moments and affine invariants for image registration and matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), 1999.

[36] A. Yezzi and S. Soatto. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. *Internaitonal Journal of Computer Vision*, 53(2):153–167, 2003.

[37] A. Yuille, D. Cohen, and P. Halliman. Feature extraction from faces using deformable templates. In *Proc. CVPR*, pages 104–109. IEEE, 1989.

[38] T. Zhang and D. Freedman. Tracking objects using density matching and shape priors. In *Proceedings of the Ninth International Conference on Computer Vision*, pages 1950–1954. IEEE, 2003.