A Refined Particle Filter Method for Contour Tracking

Xin Sun, Hongxun Yao, Shengping Zhang
{xsun, yhx, spzhang}@vilab.hit.edu.cn
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

ABSTRACT

Traditional particle filter which uses simple geometric shapes for representation cannot track objects with complex shape accurately. In this paper, we propose a refined particle filter method for contour tracking based on a binary level set model. In contrast with other previous work, the computational efficiency is greatly improved due to the simple form of the level set function. In addition, we perform curve evolution in the update step to make good use of the observation at current time. Finally, we consider some appearance information as well as the energy function to measure the weight for particles, which can identify the target more accurately. Experiment results on several challenging video sequences have verified the proposed algorithm is efficient and effective in many complicated scenes.

Keywords: Tracking, particle filter, active contour, level set, dynamic scenes

1. INTRODUCTION

Visual tracking is a challenging research topic in the field of computer vision and has been widely used in many applications such as surveillance, human-computer interfaces, vision-based control, and so on. In previous literature, a huge number of tracking algorithms have been proposed. Stochastic methods, in contrast with deterministic methods, fuse some stochastic factors and are well-adapted since their searching process can cover a much larger region and have a higher probability of reaching the global optimum of the cost function. Particle filter methods, which use a set of weighted particles to simulate the probability distribution of target, have become more and more prevalent in object tracking, especially in solving nonlinear and non-Gaussian problems. Nevertheless, traditional particle filter usually use rectangle or oval to represent the tracking results while objects, in practice, may have complex shapes, for example, hands, head, and shoulders that cannot be well described by simple geometric shapes.

Some attempts in literature have been made to use silhouette or contour, segmenting technique for dynamic tracking [1, 2, 3, 4]. A prior dynamical model on the deformation and on the similarity group parameters is defined in [1], within a particle filter framework which is used to track the deformation and the global motion over time. Chen et al. propose a contour tracker in [3] and the contour is parameterized as an ellipse. Given the observation likelihood and the state transition probabilities, the current contour state is estimated where HMM and the Viterbi algorithm is involved. Isard and Blake apply the B-spline representation for contours of objects and propose the Condensation algorithm in [4]. The authors formulate a propagation rule of shapes as an equivalent to Bayes’ rule for inferring a posterior state density from data for time-varying cases given a learned prior. Snake model [5] and its improved model are also prevalent and taken into account for tracking [6, 7, 8, 9]. In [6], as an example, an algorithm combined Kalman filter and active contours to track nonrigid objects is proposed, where the prominent B-spline method is again used for representation. Leymarie et al. in [8] try to segment and track deformable objects like amebas and proved the convergence of Snake’s motion. Since these approaches represent the contours using explicit representation and only track the affine parameters, they cannot handle local deformations of the deforming object.

In contrast with parametric active contour model, level set technique [10, 11, 12, 13] is an implicit representation of contours and able to deal with changes in topology. Explicit representation defines the boundary of the silhouette by a set of control points while implicit representation defines the silhouette by means of a function defined on a grid. In level set technique, the contour, which is formulated in a parameterization independent manner, is represented as the zero level set of the graph of a higher dimensional function and deformed until it minimizes an image-based energy functional. Some closely previous work on tracking using level set methods is given in [14, 15, 16, 17, 18]. In [14], the authors put dynamics into the geodesic active contour framework for tracking. In [15], the authors describe a unified approach for the detection and tracking of moving objects by the propagation of curves. An original scheme is proposed that can be...
viewed as a geodesic active motion detection and tracking model which basically attracts the given curves to the bottom of a potential well corresponding to the boundaries of the moving objects. Yilmaz et al. model the object shape and its changes by means of a level set based shape model in [16], where the grid points of the level set hold the means and the standard deviations of the distances of points from the object boundary and the object occlusions are resolved during the course of tracking. In [17], the authors propose a nonlinear model for tracking a slowly deforming and moving contour despite significant occlusions. The contour is represented implicitly as the (infinite-dimensional) locus of zeros of a given function, which evolves in time under the action of a group. In [18], the authors add Mumford-Shah model into the particle filter framework. The level set function need to be re-initialized to the signed distance function of the curve after each iteration, which takes very expensive calculation. The curve evolution is included in the prediction step which is a function of the previous state and observation. Actually, the current observation can be fully used once it became available at current time. The only factor they considered is the energy function of the curves, which is not enough to determine the target. Some other approaches to tracking are given in [19, 20]. In [19], the authors use deformable templates to model prior shapes allowing for many deformation modes of shapes. Mansouri uses the optical flow constraint for contour tracking in [20].

In this paper, we propose a refined particle filter approach based on a binary level set model for contour tracking, where a two-valued level set function is used to replace the signed distance function of traditional level set models. It avoids the re-initialized process of the level set function in each iteration as well as the cumbersome numerical realization, so the computational efficiency is greatly improved while maintaining the capability of processing topology changes. In addition, we perform curve evolution in the update step to make good use of the observation at current time. Finally, we consider some appearance information as well as the energy function to measure the weight for particles, which can identify the target more accurately.

The rest of this paper is organized as follows: We briefly go over the particle filter framework in Section 2. In Section 3, the proposed active contour tracking algorithm is described in detail. Experiment results on different video sequences are shown in Section 4, and Section 5 is devoted to conclusion.

2. PARTICLE FILTER FRAMEWORK

Particle filter algorithm, more detail in [21], is an estimate process based on sequential Monte Carlo methods whose substance is using the samples sequentially to implement stochastic simulation and accomplish online learning within a Bayesian framework. The purpose is to estimate the unknown state \( x_k \) at time \( k \) from a sequential observations \( z_{1:k} = \{ z_1, \ldots, z_k \} \) perturbed by noises. Two basic equations, state equation and observation equation, are referred in the whole filter algorithm:

\[
x_{k+1} = f_k(x_k, u_k) \quad (1)
\]

\[
z_k = h_k(x_k, v_k) \quad (2)
\]

where \( x_k, z_k \) are the system state and observation, \( u_k \) and \( v_k \) are the system noise and observation noise, \( f_k \) and \( h_k \) represent the state transition and observation models.

The two equations correspond to probability distributions \( p(x_k | x_{k-1}) \) and \( p(z_k | x_k) \) respectively. The key idea of particle filter is to estimate the posterior probability distribution \( p(x_k | z_{1:k}) \) by a set of weighted samples \( \{ x'_i, w'_i \}_{i=1}^{N} \), which are sampled from a proposal distribution \( q(\cdot) : x'_k \sim q(x_k | x'_{k-1}, z_{1:k}) \), (i = 1, ⋯, N). The weight of each particle is set to

\[
w'_i \propto \frac{p(z_k | x'_i)}{q(x_k | x'_{k-1}, z_{1:k})} \quad (3)
\]

Then, the posterior probability distribution can be approximated as
where $\delta(\cdot)$ is the Dirac function.

### 3. The Proposed Active Contour Tracking Method

In the refined particle filter algorithm, we regard the particle represented by a rectangle (in prediction step) or active contour (in update step) as the state $x$, and treat the image at time $k$, $z_k$, as the observation. In the prediction step, we generate a new set of particles based on the old set according to the probability distribution $p(x_k | x_{k-1})$. In practice, we can assign the position and size of the new particle by the way of constant velocity model [22]. Then we include the curve evolution equation in the update step, where we use not only the energy function of the binary level set model but also the appearance information to assign weight for particles. The update step is described in detail as follow.

#### 3.1 The level set framework

Level set methods, first proposed by Osher and Sethian in [10, 11], offer a very effective implementation of curve evolution. The basic idea of the level set approach is to embed the contour $C$ as the zero level set of the graph of a higher dimensional function $\phi(x, y, k)$, that is

$$C_k = \{(x, y) | \phi(x, y, k) = 0\}$$

where $k$ is an artificial time-marching parameter, 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 $\phi$ itself remains smooth and maintains the form of a graph.

Generality, the curve evolution equation can be defined as

$$\frac{\partial \phi}{\partial k} = V \mathbf{N}$$

where $V$ represents the speed of curve evolution while $\mathbf{N}$ represent the inward unit normal vector. Then we can get the following equation:

$$\frac{\partial \phi}{\partial k} + \nabla \phi \cdot \frac{\partial C}{\partial k} = 0$$

Based on the definition of the level set function $\phi(x, y, k)$ described above, the vector $\mathbf{N}$ can be written as $\mathbf{N} = -\nabla \phi / \|\nabla \phi\|$. Then we can get the level set implementation corresponding to the curve evolution equation (6):

$$\frac{\partial \phi}{\partial k} = V \|\nabla \phi\|$$

Given an initial curve, one must generate an initial level set function. Further more, Level-set function also needs to be re-initialize continually during its update process which usually takes a lot of computing time.

#### 3.2 The binary level set active contour model

In response to the low efficiency of the traditional level set models, binary level set active contour models use two-valued level set function $\phi$ to replace the traditional signed distance function:

$$\phi(x, y, k) = \begin{cases} 
1, & \text{if } (x, y) \text{ inside } C_k \\
-1, & \text{if } (x, y) \text{ outside } C_k
\end{cases}$$

Using this simple form can avoid the re-initialized process of the level set function in each iteration as well as the cumbersome numerical realization.
A piecewise constant-valued function \( u \) is used to approximate the image intensity distribution \( I \). The image is divided into two regions \( \Omega_1 \) and \( \Omega_2 \), in region \( \Omega_1 \), \( \phi = 1 \) and \( u = c_1 \) while in region \( \Omega_2 \), \( \phi = -1 \) and \( u = c_2 \). So the piecewise constant-valued function can be defined as

\[
u = \frac{c_1}{2}(\phi + 1) + \frac{c_2}{2}(\phi - 1)
\]  

where \( c_1 \) and \( c_2 \) are positive constants.

Then the energy function of the binary level set active contour models can be defined as follow:

\[
E_{image} = E_\phi(c_1,c_2,\phi) = \frac{1}{2} \iint_{\Omega_1} |\mu(c_1,c_2,\phi) - I|^2 \, dx \, dy + \mu \iint_{\Omega_1} |\nabla \phi| \, dx \, dy + \frac{1}{\tau} \iint_{\Omega_1} W(\phi) \, dx \, dy
\]

where \( \mu \) and \( \tau \) are the proportional coefficients. The first item is used to measure the similarity of the two-valued function \( u \) with the image \( I \), and make the function \( u \) more close to the image intensity distribution \( I \). The second item is used to measure the length of the curve \( C \), playing the role of smoothing region boundaries. The last item is for constraint of \( \phi^2 = 1 \), and \( W \) can be defined as \( (\phi^2 + 1)^2 \).

We minimize the energy function, then get \( c_1 \), \( c_2 \) as follow:

\[
c_1 = \frac{\iint_{\Omega_1} I(1+\phi) \, dx \, dy}{\iint_{\Omega_1}(1+\phi) \, dx \, dy}, \quad c_2 = \frac{\iint_{\Omega_1} I(1-\phi) \, dx \, dy}{\iint_{\Omega_1}(1-\phi) \, dx \, dy}
\]

We can see that, \( c_1 \) and \( c_2 \) are the average intensity of image \( I \) in region \( \Omega_1 \) and \( \Omega_2 \). The Euler-Lagrange equation for this functional can be implemented by the following gradient descent:

\[
\frac{\partial \phi}{\partial t} = -[u(c_1,c_2,\phi) - I] \frac{\partial u}{\partial \phi} + \mu \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \frac{1}{\tau} W'(\phi)
\]

### 3.3 Curve evolution for each sample

For each particle \( p_i \) get from the prediction step, we evolve the curve depend on the observation at time \( k \), \( z_k \), which can be realized by doing a gradient descent on the image energy \( E_{image} \):

\[
C_k = \text{evo}(s_k, z_k) = s_k^{(M)}
\]

where \( s_k \) denote the contour at time \( k \), and go through \( M \) iterations in the direction which reduces the energy \( E_{image} \) as fast as possible:

\[
s^{(\omega)} = s^{(\omega-1)} - \eta^{(\omega)} \nabla E_{image}(s^{(\omega-1)}, z), \quad \omega = 1,2,\cdots, M \quad \text{and} \quad s^{(0)} = p'
\]

Note that, the contour \( s \) is represented as the zero level set of \( \phi \), and the evolution is carried out in form of equation (13). So particles with state closer to the true state will have smaller energy than other particles after evolution.

### 3.4 The appearance models

Only considering the energy function of the curve are not reliable. In the proposed method, we also introduce some appearance models to identify the target accurately. The color-cues likelihood model is constructed due to its availability and simplicity. A similarity measure between the color histograms of the particle region \( p = \{ p^{(\omega)} \}_{\omega=1,\cdots,m} \) and the template \( q = \{ q^{(\omega)} \}_{\omega=1,\cdots,m} \) had been defined as
\[ \rho[p,q] = \sum_{n=1}^{m} \sqrt{p^{(n)} q^{(n)}} \]  

(16)

The larger \( \rho \) is, the more similar the distributions are. For two identical normalized histograms we obtain \( \rho = 1 \), indicating a perfect match.

Due to the mobile camera, the proposed method uses motion compensation first, from which we can get the corresponding parts in two successive frames. It is obvious that the differencing method based on the two parts is valid. The sum of pixel values in particle region corresponding in differences image, \( \text{sum} \), can show the amount of moving elements.

### 3.5 Compute weight for each sample

Finally, the weight of the \( i \)th particle at time \( k \) is specified by a Gaussian:

\[ w_i^k = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( -\frac{d_i^2}{2\sigma^2} \right) \]  

(17)

where \( d_i \) is the distance between the \( i \)th particle and the true state, and can be defined as follow:

\[ d_i^2 = \alpha \times E_{\text{image}}(C_i^k, z_k) - \beta \times \rho[p_i,q] - \gamma \times \text{sum}_i \]  

(18)

where \( \alpha, \beta, \gamma \) are the proportional coefficients. We choose them according to the order of magnitude of each item and set them at 0.015, 0.6, 0.4 \times 10^{-5} experimentally. Then the weights are normalized.

### 3.6 The proposed algorithm

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

**Algorithm 1: The Refined Particle Filter**

1. **Prediction Step:**
   - Generate samples \( \{\hat{p}_{i,k}\}_{i=1}^{N} \) according to \( p(x_k | x_{k-1}) \)
   - Thus we have
   \[ p(p_k | x_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(p_k - \hat{p}_i) \]

2. **Update Step:**
   - (a) Perform \( M \) iterations of curve evolution for each sample depend on the observation \( z_k \)
   \[ \hat{C}_i^k = \text{evo}(\hat{S}_i^k, z_k) = \hat{S}_i^{(M)} \]
   - (b) Compute the appearance similarity for each sample
   \[ \rho[\hat{p}_i, q] = \sum_{n=1}^{m} \sqrt{\hat{p}_i^{(n)} q^{(n)}} \text{ and } \text{sum}_i \]
   - (c) Weight each sample by
   \[ w_i^k = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( -\frac{d_i^2}{2\sigma^2} \right) \]
   \[ d_i^2 = \alpha \times E_{\text{image}}(C_i^k, z_k) - \beta \times \rho[\hat{p}_i, q] - \gamma \times \text{sum}_i \]
   - Thus we have
   \[ p(C_i | z_{k,i}) \approx \sum_{i=1}^{N} w_i^k \delta(C_i - \hat{C}_i) \]
   - (d) Resample from the above distribution to generate particles \( \{C_i^i\}_{i=1}^{N} \) distributed according to \( p(C_i | z_{k,i}) \), then we have
   \[ p(C_i | z_{k,i}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(C_i - C_i^i) \]

3. Go back to the prediction step for time \( k+1 \).
4. EXPERIMENT RESULTS

In this part, we test the proposed method on several challenging video sequences taken from moving cameras outdoors. We use HSV color space, and set $\sigma=0.15$.

First, we compare the proposed method with some other algorithms on a ship sequence to show the improvement of computational efficiency. The sequence consists of 350 frames and describes a ship, with similar color distribution to the water, navigating on the river with moving waves behind and illumination changes. The four algorithms tested are: (a) traditional particle filter; (b) the proposed method excluding appearance information process; (c) the proposed method; (d) particle filter with tradition Mumford-Shah method, more detail in [18]. We use 30 particles and $M=20$. Table 1 shows the average CPU time of these algorithms (obtained by Vc 6.0, CPU Pentium (R) Dual E2160 1.8GHz 1.8GHz), and Fig.1 shows the specific CPU time of each frame (from the 10th frame to the 350th frame, sampled every 10 frames). We can see that the computational efficiency of our method is improved a lot while the tracking performance is good as shown in Fig.2.

Table 1. Average CPU time per frame of the four algorithms on ship sequence.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
<td>0.1127s</td>
<td>0.6332s</td>
<td>0.6755s</td>
<td>1.0828s</td>
</tr>
</tbody>
</table>

Figure 1. The specific CPU time per frame of the four algorithms on ship sequence

![Figure 1](image_url)

Figure 2. Tracking results of the proposed algorithm on ship sequence for frame 12, 45, 100, 170, 290.

The algorithm (d), which uses only energy function to weight particles and evolves the curve in prediction step, can get the similar good results as the proposed method on the single-color targets, such as the ship in fig.2. And here, we test them on two colorful targets to show the advantages of the proposed method. The first sequence shows a man in colorful clothes walking on the balcony. We use 20 particles and $M=50$. As we can see in Fig.3, the accuracy of the results using algorithm (d) are not satisfactory while our proposed algorithm can improve the tracking quality dramatically.
A refined particle filter algorithm for contour tracking based on a binary level set model has been proposed in this paper. This method greatly improves the computational efficiency and performs curve evolution in update step to make good use of the observation at current time. Some appearance factors are considered as well as the energy function to identify the target more accurately. Experiment results have verified that the proposed method is efficient and robust in many complex scenes.

6. ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (60775024), National Basic Research Program of China (2009CB320906).
REFERENCES