Finite-Element Level-Set Curve Particles

Tingting Jiang and Carlo Tomasi[*]
Department of Computer Science, Duke University
Durham, NC 27708
{ruxu, tomasi}@cs.duke.edu

Abstract

Particle filters encode a time-evolving probability density by maintaining a random sample from it. Level sets represent closed curves as zero crossings of functions of two variables. The combination of level sets and particle filters presents many conceptual advantages when tracking uncertain, evolving boundaries over time, but the cost of combining these two ideas seems prima facie *prohibitive. A previous publication showed that a large number of virtual level set particles can be tracked with a logarithmic amount of work for propagation and update. We now make level-set curve particles more efficient by borrowing ideas from the Finite Element Method (FEM). This improves level-set curve particles in both running time (by a constant factor) and accuracy of the results.*

1. Introduction

Tracking an animal in video, a wildfire on the ground, or a drifting cloud in the sky involves following boundaries on the plane that move and deform over time: the silhouette of the animal, the fire front, a vapor iso-density contour in the cloud pattern.

When the region within the boundary splits and recombines, and as it develops and refills holes, it becomes theoretically more elegant and practically simpler to represent the region's boundary as the zero-crossing of a scalar function of the two coordinates on the plane. This is the well known *level set* representation of curves. Although encoding a function of two variables is more expensive than encoding a set of curves, changes in geometry and topology become trivial to handle. Computation is now much more uniform and leads to practical efficiency and simple code.

In many applications, knowledge about the curve's position and shape (the curve's "state" for short) comes from a sparse set of uncertain measurements. Particle filters have been used successfully in many contexts to track and refine the state of evolving objects from sparse measurements. These filters maintain random samples (a set of particles) out of the probability density that represents the current knowledge about the object's state. Particles are propagated as the state changes, and are updated when new measurements become available.

Tempting as the combination of level sets and particle filters may be, it immediately raises the specter of computational complexity: the space of curves is infinite, and a density in such a space needs many samples indeed for a faithful representation. If each particle is a set of curves encoded by a level set, the complexity of the naïve combination of these two representational ideas multiplies the number of particles by the cost of handling a level set function. This product is invariably a large number, and this explains why the merge of these ideas has not been attempted until recently.

The internal structure of individual particles as sets of curves was exploited in [10]. It was shown that it makes mathematical sense to "add" two sets of curves by adding the level set functions that represent them. Using this insight, one can in effect track C particles while reaping the combinatorial benefit of implicitly considering the superposition of all the 2^C combinations of the C sets of curves that are explicitly tracked. This device made level-set curve particles feasible. In that paper, level set functions were mixtures of Gaussian functions. In contrast, we now show that a representation based on the combination of piecewise-linear functions, in the spirit of the Finite Element Method (FEM), leads at the same time to shorter running times (by a constant factor) *and* greater accuracy in the results (Fig.1).

2. Related work

Level sets [14, 3, 13] describe the boundary B of an evolving (not necessarily connected) region on the plane or in space as the zero crossing of a function $\phi(\vec{x}, t)$ of space and time. While many functions ϕ share the same zero-crossing B, computational considerations [15] suggest using the signed distance function of B, which is then maintained in a narrow *band* [1] around the boundary B. The

[*]This research was partially funded under NSF grant IIS-0534897.

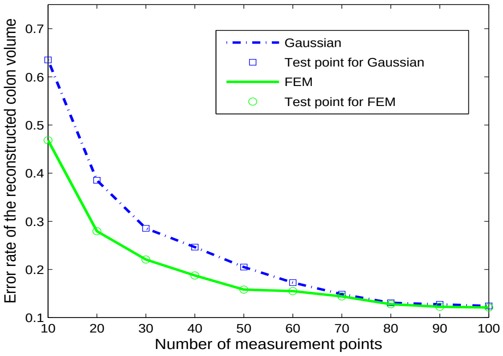

Figure 1. Compare error rates of FEM and Gaussian methods with different number of measurement points for human colon tracking.

motion model is then a partial differential equation (PDE) for ϕ. The main strength of level sets is that they account effortlessly for changes of boundary topology, as exemplified in applications to image segmentation [4], object detection [16], tracking [18], shape modeling [12] and medical image segmentation [17], among others.

Particle filters [6] have been used in this context mainly in the area of active contours [9, 2]. These approaches capture the uncertain position of a boundary at time t by a probability distribution represented by a random sample of boundaries (*particles*). Each boundary is represented explicitly, e.g. with splines [5] and *propagated* forward in time through an assumed, uncertain motion model. Measurements *update* the particles by weighing each particle by its posterior probability given the measurements. *Resampling* draws new particles from the posterior to be ready for a new step of propagation. This cycle is analogous to the estimation loop of a Kalman filter [11], but maintains a multi-modal distribution rather than a Gaussian one.

The recent, full-fledged combination of level sets and particle filtering called "Level-Set Curve Particles" is reviewed in the next section.

3. Level-set curve particles

In [10] a method was proposed to represent and track a dynamic set of closed curves (a "boundary"). This method capitalizes on the observation that many of the curves that populate the boundary distribution being tracked are very similar to each other. A *base curve* which is called "mother" particle accounts for the macroscopic shape of the boundary. The base curve is then deformed by C *base perturbations* which provide an implicit representation of the 2^C deformations obtained by applying any subset of the C base perturbations to the base curve. Through this device, an exponential number of curves can be propagated and their likelihoods can be computed at linear cost in the framework of a particle filter.

More specifically, a particle set $\chi^{(t)} = \{\phi_c^{(t)}\}_{c=0}^C$ represents the probability distribution of the boundary $B^{(t)}$ at time t. Each particle is a signed distance function $\phi_c^{(t)} : \mathbb{R}^2 \rightarrow \mathbb{R}$ whose zero level set denotes an estimate of the boundary $B^{(t)}$. A weight $w_c^{(t)}$ determined by the measurements is associated to each particle. The function $\phi_0^{(t)}$ is the "mother" particle and the other particles $\{\phi_c^{(t)}\}_{c=1}^C$ are explicit "child" particles. Each particle is associated with a weight $w_c^{(t)}$. In fact, there can be N_M such sets to represent multi-modal probability distribution of the boundary. For simplicity, only one particle set is discussed in the following. The result can be extended to N_M particle sets in a straightforward way.

Gaussian mixtures were used in [10] to approximate the difference $\Delta\phi(\vec{x}, t)$ between mother particle and child particles. Assume that $\Delta\phi(\vec{x}, t) = \sum_{c=1}^C a_c(t) G_c(\vec{x}, t)$ where $G_c(\vec{x}, t)$ is initialized as a 2D Gaussian function with randomly generated mean and covariance matrix at $t = 0$ and then propagated by the motion field over time. $a_c(t)$ is a scalar function of t. C is the number of explicit "child" particles. In fact, $a_c(t)$ is restricted to be ± 1 or 0, thereby leading to 2^C combinations of "implicit" child particles. In other words, the representation of $\Delta\phi$ is simplified as a C-bit binary vector with C functions which are initialized as Gaussian functions and changing over time.

The outline of the resulting tracking method is Algorithm 1. At each time step, the input is the particle set $\chi^{(t-1)}$ from last time step and the output is the new particle set $\chi^{(t)}$. $Q_m^{(t)}$ denotes the estimate of a point on boundary $B^{(t)}$ returned by each of M observers.

Algorithm 1 : Tracking algorithm

```
  INPUT: $\chi^{(t-1)}$  $(t > 0)$
  OUTPUT: $\chi^{(t)}$.
  (1) Propagate $\chi^{(t-1)}$ to $\bar{\chi}^{(t)}$;
  (2) Take measurements $\{Q_m^{(t)}\}_{m=1}^M$;
  (3) Update $\bar{\chi}^{(t)}$ to $\hat{\chi}^{(t)}$;
  (4) Resample $\hat{\chi}^{(t)}$ and generate $\chi^{(t)}$.
```

Initialization: The first iteration of Algorithm 1 requires $\chi^{(0)}$ which is initialized as follows. The mother particle of $\chi^{(0)}$ is defined as $\phi_0^{(0)}$. Its explicit child particles are initialized by choosing C different Gaussian functions and combining them with $\phi_0^{(0)}$. Specifically, the initial particle set is denoted as $\chi^{(0)} = \{\phi_0^{(0)}, \{W_c^{(0)}, \Delta\phi_c^{(0)}\}_{c=1}^C, \{w_c^{(0)}\}_{c=0}^C\}$ where $\Delta\phi_c^{(0)}$ represents the difference between mother and child c which is only nonzero inside window $W_c^{(0)}$ and $w_c^{(0)}$ is the weight of particle c. All particles have the same initial weight. If the region of the plan of interest is discretized into a grid of size N, initialization takes $O(CN)$ time.

Propagation: Suppose the motion field at each time step is externally given by a smooth velocity field $\vec{V}(\vec{x})$ on the plane. Both mother and explicit child particles are propagated by the level set equation:

$$\phi_t = -\vec{V} \cdot \nabla \phi \tag{1}$$

which takes $O(CN)$ time. The particle set becomes $\bar{\chi}^{(t)} = \{\bar{\phi}_0^{(t)}, \{\bar{W}_c^{(t)}, \Delta\bar{\phi}_c^{(t)}\}_{c=1}^{C}, \{w_c^{(t-1)}\}_{c=0}^{C}\}$ where $\bar{\phi}_0^{(t)}$ is the propagated mother and the difference between mother and child c is still represented by $\Delta\bar{\phi}_c^{(t)}$ within window $\bar{W}_c^{(t)}$. The weights do not change during propagation. Because of the linearity of the gradient operator ∇, implicit child particles are implicitly propagated for free.

Update: Before update, each observer returns a noisy measurement $Q_m^{(t)}$ which is the position of the closest point on the boundary. The likelihood function used to evaluate the particles is defined as follows:

$$\Lambda^{(t)}(\phi) = \prod_{m=1}^{M} G(\phi(Q_m^{(t)})) \tag{2}$$

where G is a Gaussian function whose standard deviation ζ depends on the noise statistics of the measurements. The mother's weight is updated as $\hat{w}_0^{(t)} = \Lambda^{(t)}(\bar{\phi}_0^{(t)}) \cdot w_0^{(t-1)}$. Since $\bar{\phi}_0^{(t)}$ is always maintained as a signed distance function, $\bar{\phi}_0^{(t)}(Q_m^{(t)})$ can be obtained as a simple lookup. Update of the mother particle takes $O(M)$ time. The likelihoods and weights for explicit child particles can be obtained by modifying those of mother particle based on the differences $\Delta\bar{\phi}_c^{(t)}$ at a cost of $O(CM)$.

A small boundary component is added to the mother particle whenever the measurement information implies that a new boundary component has appeared. This has a worst-time cost $O(MN)$.

Resampling: To resample all explicit and implicit child particles, all their weights must be known and therefore evaluation of the weights of implicit children is necessary. Two cases are discussed in [10]: disjoint windows and intersecting windows. For the latter case, the following lemma is proven in [10]:

Lemma 1 *Let* $K_c^{(t)} = \frac{w_c^{(t)}}{w_0^{(t)}}$ *and* $\hat{K}_c^{(t)} = \frac{\hat{w}_c^{(t)}}{\hat{w}_0^{(t)}}$. *Given* $Z \subseteq \{1, \ldots, C\}$, *suppose that the combined particle* $\phi_Z^{(t)}$ *has weight* $w(\phi_Z^{(t-1)}) = w_0^{(t-1)} \cdot \prod_{c \in Z} K_c^{(t-1)} \prod_{j,k \in Z, j \neq k} S_{jk}^{(t-1)}$ *before propagation. After update, the weight of* $\bar{\phi}_Z^{(t)}$ *is* $w(\bar{\phi}_Z^{(t)}) = \hat{w}_0^{(t)} \cdot \prod_{c \in Z} \hat{K}_c^{(t)} \prod_{j,k \in Z, j \neq k} \hat{S}_{jk}^{(t)}$ *where* $\hat{S}_{jk}^{(t)} = S_{jk}^{(t-1)} \cdot I_{jk}^{(t)}$ *and* $I_{jk}^{(t)} = exp(\sum_{Q_m^{(t)} \in \overline{W}_j^{(t)} \bigcap \overline{W}_k^{(t)}} \frac{-2\Delta\bar{\phi}_j^{(t)}(Q_m^{(t)})\Delta\bar{\phi}_k^{(t)}(Q_m^{(t)})}{\zeta^2})$.

This lemma tells us that besides $\chi^{(t)}$, it is also necessary to maintain an intersection factor set $\{S_{jk}^{(t)}\}$ for every pair of explicit child particles over time. This factor is defined recursively by the lemma itself, as $S_{jk}^{(t)}$ is updated as $\hat{S}_{jk}^{(t)} = S_{jk}^{(t-1)} \cdot I_{jk}^{(t)}$. With $\hat{S}_{jk}^{(t)}$, the weights of implicit particles can be obtained. Then the goal of resampling implicit particles can be achieved by changing the weights of explicit particles. Resampling costs $O(2^C)$ but the constant factor that multiplies 2^C is small. For the case of intersecting windows, the cost of maintaining intersection factors $\{S_{jk}^{(t)}\}$ is $O(MC^2)$.

Complexity: The total complexity for Algorithm 1 is $O(2^C + CN + MN + MC^2)$. A detailed representation of shape variations requires $C = O(N)$, so the asymptotic cost of Algorithm 1 is $O(2^N + MN^2)$.

4. FEM curve particles

Experiments on tracking the boundary of a colon in tomographic imagery from sparse edge measurements show the promise of the above approach. However, the tracking results are "bumpy" compared to the ground truth. To make the tracking results smoother and the tracking approach more efficient, we replace Gaussian perturbations with finite element perturbations.

The finite-element method (FEM) [8] originated from the needs for solving complex elasticity, structural analysis problems in civil engineering and aeronautical engineering. It can be used for finding approximate solution of PDE as well as of integral equations such as the heat transport equation. We are first inspired to use finite element perturbations because of their linearity between the nodal points. If the propagations of the grid points between the nodal points can be implicitly interpolated by the propagations of the nodal points, a significant amount of time can be saved. Second, the approximation error for FEM is inherently better understood than that of mixtures of Gaussian functions.

In this section, we show that FEM is a better way than Gaussian functions to represent the differences $\Delta\phi^{(t)}$ between mother and child particles.

Assume that $\Delta\phi(\vec{x}, t) = \sum_{e=1}^{E} d_e(t) T_e(\vec{x})$ where $T_e(\vec{x})$ denotes the time-independent finite element function and the coefficient $d_e(t)$ is a time-dependent real number. E is the total number of elements. This representation essentially projects $\Delta\phi$ onto E elements linearly and reduces the boundary tracking problem to that of estimating a time-dependent vector $\{d_e(t)\}_{e=1}^{E}$ such that the boundary under tracking can be represented by the zero crossing of $\phi(\vec{x}, t) = \phi_0(\vec{x}, t) + \sum_{e=1}^{E} d_e(t) T_e(\vec{x})$.

There are several choices for 2D finite elements including the bilinear quadrilateral element, isoparametric element, linear triangular element, etc. [8]. For convenience, we choose the linear triangular element. The N grid points is divided into $R \times R$ rectangles each of which has N/R^2 grid points and is cut by its diagonal into two triangles (Fig. 2 (a)). Each triangle is a domain which has three nodes. For example, the triangle $\Delta p_1 p_2 p_3$ is a domain. The element shape function is illustrated in Fig. 2 (b). There are $(R-1)^2$ nodal points on the plane excluding those on the bounding box. Each nodal point has six connected neighbors and is adjacent to six element domains. The combination of six element shape functions which share same nodal point looks like a tent in Fig. 2 (c). These tents are the elements T_e which will be used to represent $\Delta\phi$. There are $E = (R-1)^2$ elements on the plane each of which can perturb the mother particle in a different area (Fig. 3).

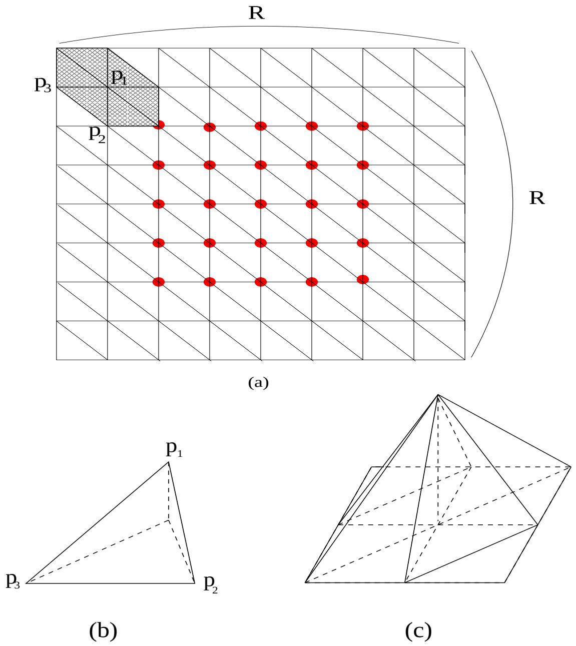

Figure 2. Illustration of the linear triangular element. (a) Triangular domains. (b) The element shape function. (c) The hexagonal “tent” T_e formed by the six element shape functions that share a nodal point.

Now the particle set $\chi^{(t)}$ can be viewed as one mother particle $\phi_0^{(t)}$ plus a set of C E-dimensional vectors each of which corresponds to an explicit child particle. Let a C by E matrix $D^{(t)}$ represent the vectors s.t. for every explicit child particle $\phi_c^{(t)} = \phi_0^{(t)} + \sum_{e=1}^{E} D^{(t)}(c,e) \cdot T_e$. Since the E elements are fixed, the particle set can be written as $\chi^{(t)} = \{\phi_0^{(t)}, D^{(t)}, \{w_c^{(t)}\}_{c=0}^{C}\}$. Compared to [10] , only the representation of the explicit child particles is changed. The way to combine them is same and there are still 2^C implicit child particles.

Based on the new representation, Algorithm 1 can be implemented with the following differences from previous work. First of all, the particle set $\chi^{(0)}$ is initialized by choosing C elements from the above E elements and adding them to $\phi^{(0)}$. In particular, $C \leq (R-3)^2$ because only the nodal points in the middle of the plane have all the neighbors around them. See the red points in Fig. 2 (a). Each element perturbs $\chi^{(0)}$ in a limited area. Fig. 3 shows how the array of tent functions influence the boundary when $R = 8$ and $C = 12$. The blue curve is the mother particle and the red curves are child particles. The red window denotes the position of each tent. After initialization, the particle set is $\chi^{(0)} = \{\phi_0^{(0)}, D^{(0)}, \{w_c^{(0)}\}_{c=0}^{C}\}$ where $D^{(0)}$ is the representation of the C explicit child particles and $w_c^{(0)}$ is the initial weight which is same for all particles.

Second, the propagation of each tent function T_e can be approximated by only propagating function values at its nodal points because the tent function is piecewise linear and the motion field $\vec{V}(\vec{x})$ is smooth. Specifically, use the level set equation Eqn. (1) to calculate the function value change for each nodal point p in the nodal point set $NP(e)$ of T_e. After propagating T_e, construct a matrix U of size E by E such that each row of U represents how the propagated T_e is approximated by its neighboring tent functions and itself, *i.e.*, $T_e - \vec{V} \cdot \nabla T_e \approx \sum_{p \in NP(e)} U(e,p) \cdot T_p$ where both e and p are the indices of the nodal points corresponding to the tent functions. The changes for nodal points on the bounding box is set to zero. If the gradient at a nodal point is not well-defined, take the average of gradients from different directions. Because of the linearity of the ∇ operator, we can propagate each explicit child particle by taking the product of $D^{(t-1)}$ and U as $\bar{D}^{(t)} = D^{(t-1)} \cdot U$. After propagation, the particle set is $\bar{\chi}^{(t)} = \{\bar{\phi}_0^{(t)}, \bar{D}^{(t)}, \{w_c^{(t-1)}\}_{c=0}^{C}\}$.

Third, to update each explicit child particle, compute its signed distance function $\bar{\phi}_c^{(t)}$ and then $\bar{\phi}_c^{(t)}(Q_m^{(t)})$ becomes available by lookup. Evaluate the likelihood of the particle by Eqn. (2) and update its weight.

Last, to maintain the intersection factor set $\{S_{jk}^{(t)}\}$, define $I_{jk}^{(t)} = \prod_{m=1}^{M} exp(-2\Delta\phi_j^{(t)}(Q_m^{(t)})\Delta\phi_k^{(t)}(Q_m^{(t)}))/\zeta^2)$. Note that child particles are allowed to intersect in any combination. Even so, thanks to the quadratic form of the exponents in the Gaussians of the likelihood function, only pairwise intersection factors need to be maintained.

The computational complexity of the algorithm is analyzed as follows. For initialization, the mother particle and tent functions cost $O(N)$, and the matrix D costs $O(C)$. Propagation advances the mother particle by the motion field, at a cost $O(N)$ and E tent functions by $O(E)$ because each tent only takes constant time. Then propagation of explicit child particles requires $O(CE)$ time. For update, the mother particle takes $O(M)$ time and the explicit children take $O(NC + MC)$. The cost of maintaining intersection factors $S^{(t)}$ is $O(MC^2)$. Resampling still costs $O(2^C)$. The generation of new components takes at most $O(MN)$ time.

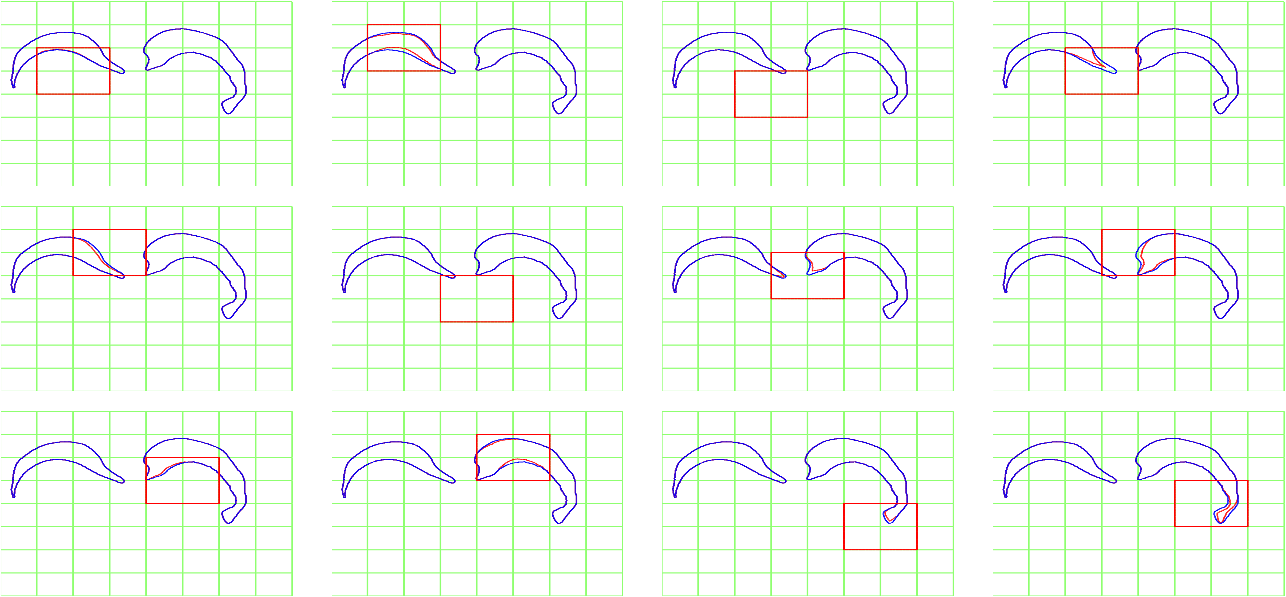

Figure 3. Explicit particle set generated by adding tent functions to a mother particle (blue). Resulting perturbations are in red.

Therefore, the total complexity for the proposed tracking algorithm is $O(2^C + CE + NC + MC^2 + MN)$. Asymptotically, $E = O(C) = O(N)$, so the cost is $O(2^N + MN^2)$ which is same as for the Gaussian approach.

If we had used a standard particle filter with $2^C = O(2^N)$ explicit particles, the cost for propagation and update would be $O(2^N(N+M))$ which is much higher than that for the proposed approach. If we use N_M particle sets each of which has a different mother particle to represent the probability distribution of the boundary, both complexities are scaled by N_M because the algorithm is run N_M times for each of the particle set. If we use the so-called narrow-band version of LSM [1] , assume that the number of grid points in the narrow band is N_B and the number of related elements is E_B. The total cost is $O(2^C + CE_B + CN_B + MC^2 + MN_B)$. Asymptotically, $E_B = O(C) = O(N_B)$, so the cost is $O(2^{N_B} + MN_B^2)$. If we had used a standard particle filter with 2^C explicit particles the cost for propagation and update would be $O(2^{N_B}(N_B + M))$, which is much higher.

5. Experiments

We implemented the proposed FEM-based level-set curve particles on two different data sets: human colon data and sea surface temperature data, and compare the results with those of Gaussian-based level-set curve particles.

5.1. Colon Tracking

The algorithm in Section 4 has been run on the data set used in [10] which is a sequence of 355 slices from a Computerized Axial Tomography (CAT) scan of a human colon. Results were compared to those reported in [10]. The boundaries being tracked are the cross-sections of the colon with the horizontal slices of the CAT volume. Boundaries appear and disappear as from one slice to the next (See Fig. 4 (a)). In the experiments, each boundary was first reconstructed by a standard edge detector and the resulting boundaries were taken as ground truth. The tracker on the other hand can only access the ground truth through 100 measurement points in each slice. Fig. 4 (b) is the 3D reconstruction of the tracking results from the proposed FEM approach and Fig. 4 (c) is the 3D reconstruction with Gaussian perturbations from [10]. Both tests use 10 explicit child particles and 100 measurement points per slice. The grid size is 512×512 and the element array size is 8×8. Although the asymptotic complexities for both approaches are same, practical running times are different. The average running time for the FEM approach is about 11s per slice compared to 20s per slice for the Gaussian approach.

More importantly, the FEM approach yields more accurate results. Accuracy is measured by the error rate defined as the ratio of the symmetric difference between the 3D reconstructions of the tracking results and the ground truth over the volume of the ground truth. Specifically, let $\hat{\mathcal{B}}^{(t)}$ denote the area enclosed by the tracking result boundary and $\mathcal{B}^{(t)}$ denote the area enclosed by the ground truth boundary at time step t respectively. Then the error rate is defined as

$$ErrorRate = \frac{\Sigma_t |(\hat{\mathcal{B}}^{(t)} \backslash \mathcal{B}^{(t)}) \cup (\mathcal{B}^{(t)} \backslash \hat{\mathcal{B}}^{(t)})|}{\Sigma_t |\mathcal{B}^{(t)}|} \qquad (3)$$

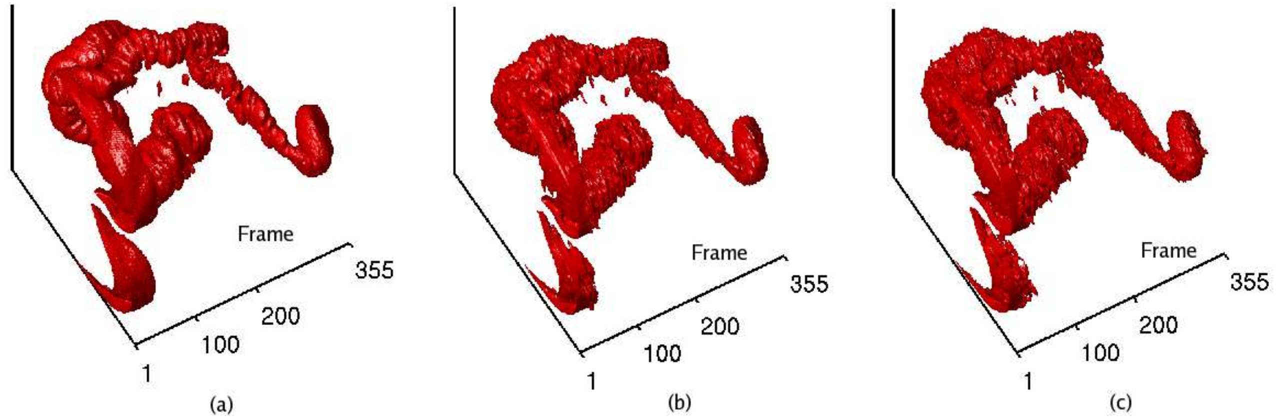

Figure 4. Compare 3D reconstructions of ground truth (a), FEM tracking results (b) and Gaussian tracking results (c).

where $|\cdot|$ means the size of the area or the number of grid points inside the area in discrete case.

Our results are compared to the previous Gaussian approach with the same number of child particles in Fig. 1. With decreasing number of measurement points, the error rate of FEM approach is increasing more slowly than that of the Gaussian approach.

5.2. Sea Surface Temperature Tracking

Sea surface temperature (SST) is the water temperature at the sea surface. SSTs above 26.5 degrees C are favorable for the formation and sustaining of tropical cyclones. In general, the higher the SST, the stronger the storm. So tracking the isotherms of high SSTs is very helpful in forecasting tropical cyclones and hurricanes.

We tracked the SST isotherms in a data set where dense information of the boundary is available, but we simulate the sparseness of observers by withholding the dense information from our tracking algorithm. Specifically, we obtain NOAA_ERSST_V2 data [1]which is an extended reconstruction of historical SST monthly mean values using improved statistical methods from 1854 to present and take it as ground truth. In particular, we define the boundary under tracking is the isotherm at 25 degrees C. One example of the boundary (Jan. 1990) is shown in Fig. 5. The observers are simulated by the points on the map and each observer can report its closest point on the boundary. In real applications, the observers could be a set of ships or buoys with temperature sensors. Since SST changes periodically, for simplicity, a motion model is learned for each calendar month (Jan. to Dec.) from the historical data between 1990 and 1999. For example, to get the motion model for Jan., we compare the data in Jan. and Feb. in every year and find a motion model by the optical flow method [7], and then take the average of the motion models over the ten years (Fig. 5).

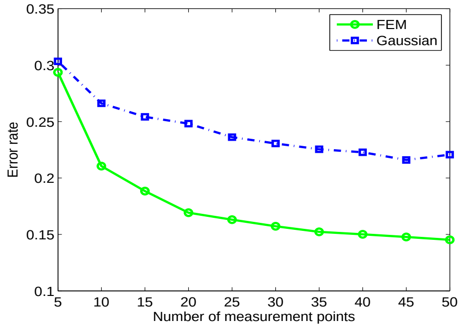

Figure 6. Compare error rates of FEM and Gaussian methods with different number of measurement points for SST tracking.

We started tracking from Jan. 2000 and ended in Dec. 2004 (60 months). The particle set was initialized by perturbations based on the boundary in Dec. 1999. The perturbations are generated by linear combinations of the element functions. The number of explicit child particles is 50. The grid size is 89×180 (2 degree latitude $\times 2$ degree longitude, global grid:$88N - 88S, 0E - 358E$). The element array size is 9×18. We have applied both the FEM method and Gaussian method. The error rate is defined as Eqn. (3). The comparison of the accuracy of these two methods is shown in Fig. 6. The improvement of the accuracy by the FEM approach is obvious when $M > 5$.

6. Summary and future work

We have introduced "Finite-Element Level-Set Curve Particles" that combine piecewise-linear functions to make the merge of level sets and particle filters more efficient. The source of greater efficiency is that FEM nodal points carry all the information about the piecewise-linear functions between them. In addition, the approximation error

[1] provided by NOAA/OAR/ESRL PSD, Boulder, Colorado, USA, from http://www.cdc.noaa.gov/

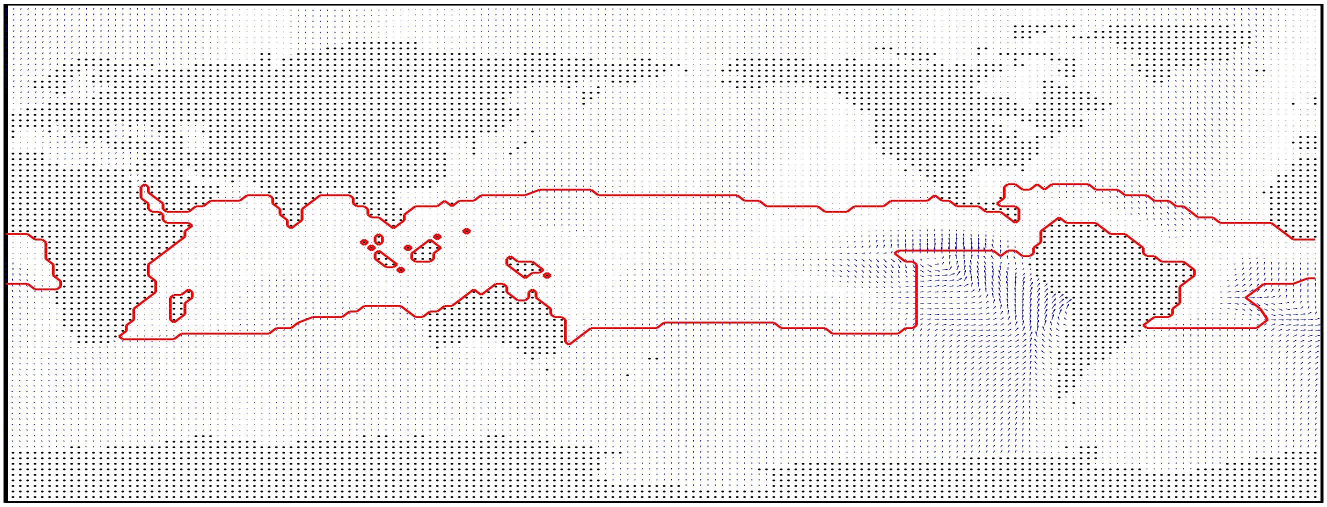

Figure 5. The red curve is the isotherm of SST at 25 degrees C in Jan. 1990 and the blue arrows denote the learned motion model for the isotherm at 25 degrees C in Jan. from the SST data between 1990 and 1999. The black dots denote the land.

for FEM is inherently better understood than for mixtures of Gaussian functions. The new method simplifies the representation of the child particles as well as their propagation. The results show improvements in both running time (by a constant factor) and accuracy compared to Gaussian level-set curve particles.

For future work, we are interested in improving the computation efficiency for maintaining the weights of implicit child particles and resampling which takes exponential cost. We were puzzled by the fact that in the colon experiment the improvement introduced by the FEM method increases with fewer observers, while it increases with more observers in the temperature experiment. Maybe the more accurate motion model available in the latter experiment is a factor in this phenomenon, but we intend to investigate this further for a more definite explanation.

References

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. of Comp. Phys.*, 118(2):269 – 277, May 1995. 1, 5

[2] A. Blake and M. Isard. *Active Contours*. Springer, New York, NY, 1999. 2

[3] V. Caselles, J. M. Morel, G. Sapiro, and A. Tannenbaum, editors. Special issue on partial differential equations and geometrydriven diffusion in image processing and analysis. *IEEE Trans. on Image Proc.*, 7:269–473, 1998. 1

[4] D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 169–176, Oct 2003. 2

[5] C. De Boor. *A Practical Introduction to Splines*. Springer, New York, NY, 2001. 2

[6] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice*. Springer, New York, NY, 2001. 2

[7] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 20, 1981. 6

[8] T. J. Hughes. *The Fintnite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Inc., 2000. 3

[9] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998. 2

[10] T. Jiang and C. Tomasi. Level-set curve particles. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3953 of *LNCS*, pages 633–644. Springer, 2006. 1, 2, 3, 4, 5

[11] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME J. on Basic Eng.*, 82:34–45, 1960. 2

[12] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on PAMI*, 17(2):158–175, 1995. 2

[13] M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors. *Scale Space Theories in Computer Vision*, volume 1682. Springer, Berlin, 1999. 1

[14] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. of Comp. Phys.*, 79:12–49, 1988. 1

[15] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York, NY, 2002. 1

[16] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. PAMI*, 22(3):266–280, March 2000. 2

[17] A. Tsai, A. Yezzi, Wells, W. Jr., C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Trans. on Med. Im.*, 22(2):137–154, February 2003. 2

[18] T. Zhang and D. Freedman. Tracking objects using density matching and shape priors. *ICCV*, 2:1050–1062, Oct 2003. 2