Contour Tracking

Akil Narayan and Chia-Ying Lee

May 8, 2007

The focus of our problem is the following: suppose we have a contour C which evolves as a function of t. Further suppose that we are given (noisy) image observations Y, which are function of C, but do not give us direct information about C. How can we reconstruct the evolution of this contour?

1 Introduction

Because the nature of this problem is so closely connected with nonlinear filtering, we shall adopt that notation here. Let $\theta(t)$ represent the contour. We wish to know $\theta(t)$, but we are only given $\theta(0)$ (or an approximation of it). We suppose that we are given observations $Y(t)$ in the form

$$Y(t) = h(\theta(t)) + \xi(t) \tag{1}$$

where $h(\cdot)$ is some function transforming the contour θ into the image data $Y(t)$. It should be noted that we have no knowledge about the nature of h. We do not know in what sense h is 'smooth' with respect to θ, and we must even allow h to be multi-valued. However, in our current problem of simply tracking a contour, we shall assume that $h(x) = x$ so that $Y = \theta + \xi(t)$. $\xi(t)$ is some random process which introduces noise into the observations. We shall soon return to how we can cope with ξ in order to reconstruct θ, but first let us present some assumptions we make on θ.

2 Evolution of θ

We have in mind that θ be the edge of some object that we track in the image. With this in mind, we shall suppose that the object moves in a relatively smooth fashion with respect to time, but allow for it to experience random deformations due to, among other things, the transformation from the 3D world to the 2D contour/image.

Let $\theta(t, x) : [0, T] \times R \to \{0, 1\}$ with $x \in R$ where R is some rectangular domain in $\mathbb{R}^2$. We propose the following model for the evolution of θ:

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= -(\boldsymbol{v}(x,t) + \boldsymbol{w}(t)) \cdot \nabla \theta && (2) \\ \theta(0, x) &= \theta_0(x) && (3) \end{aligned}$$

Technically speaking, θ is the indicator function of the object to track, but throughout this paper we use θ loosely to refer to the contour that is the boundary of the object. We have introduced random velocity terms $\boldsymbol{v} = (v^1, v^2)$ and $\boldsymbol{w} = (w^1, w^2)$ which represent the motion of the actual contour θ. This can be interpreted as a rigid motion with velocity $\boldsymbol{w}$, with a random location-dependent deformation given by $\boldsymbol{v}$. If so desired, we can also impose a fixed, deterministic velocity $\boldsymbol{u}(x, t)$ that dictates a typical motion behaviour (e.g. when tracking cars travelling on a road, the cars follow roughly the trajectory of the road), but we do not consider this case in this project.

To be more explicit, we assume that $\boldsymbol{v}$ and $\boldsymbol{w}$ have the following colored noise forms:

$$\boldsymbol{v}^i(\omega, x, t) = \sum_{n=1}^{N} \zeta_n^i(\omega)\phi_n(x,t) \tag{4}$$

$$\boldsymbol{w}^i(\omega, x, t) = \sum_{m=1}^{M} \chi_m^i(\omega)\psi_m(t) \tag{5}$$

where ζ_n^i and χ_m^i are independent, $\mathcal{N}(0,1)$ random variables, and ϕ_n and ψ_m are orthonormal bases for $[0,T] \times R$ and $[0,T]$, respectively. With this form, taking $N, M \to \infty$ leads $\boldsymbol{v}$ and $\boldsymbol{w}$ to be white noises on their respective spaces. We use this model as a general rule for evolving contours. The lower the order of the noises (N and M), the smoother the evolution of the actual contour θ.

In practice we use the Legendre polynomials for ψ_m, and we write $\phi_n(x,t) = \phi_r^x(x)\phi_s^t(t)$ where r and s are functions of n, and ϕ^t are the Legendre polynomials, and the ϕ^x are a tensor-product basis for R using one-dimensional Legendre polynomials. We usually take $M \sim 15$.

We briefly remark that since $\boldsymbol{v}$ and $\boldsymbol{w}$ are Lipschitz in x for every ω, the solution to (2) is well-defined classically speaking for every ω, and the solution can be computed via the method of characteristics. Let $\boldsymbol{x}(t)$ denote a characteristic of equation (2). Then, for any fixed ω, the evolution equation for the characteristics and for the values of θ are

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{v} + \boldsymbol{w} \tag{6}$$

$$\dot{\theta}(\boldsymbol{x}(t), t) = 0 \tag{7}$$

where $\dot{\boldsymbol{x}}$ represents the full time derivative of $\boldsymbol{x}$.

This model gives us a good way of generating flowing, deforming contours. We shall assume that this model is the way in which the object in an image moves. However, if we are only given observations Y of the location, then we do not actually have access to θ and this leads us to consider defining our system as a Hidden Markov Model.

3 The Hidden Markov Model

Before trying to mold our problem into a Hidden Markov Model, we must first construct a mathematical model by which observations Y are defined in terms of the actual parameters θ.

There are many ways to model noisy observations. The baseline, as mentioned earlier, is to have the relationship $Y = h(\theta) + \xi$, where ξ represents some kind of noise. We first present an evolution model for the 'noisy' contour Y. Let $Y(t,x) : [0,T] \times R \to \{0,1\}$ with $x \in R$ where R is some rectangular domain in $\mathbb{R}^2$, and let Y be random. We suppose that Y evolves according to the following advective SPDE:

$$\frac{\partial Y}{\partial t} = -(\boldsymbol{v}(x,t) + \boldsymbol{w}(t)) \cdot \nabla\theta - f(x,t) \tag{8}$$

$$Y(0,x) = Y_0(x) \tag{9}$$

The term $f(x,t)$ is a random process that takes into account the randomness introduced by noise or blurring of the object. We intend $f(x,t)$ to represent randomness in the representation of the contour as an image, and not any sort of error that is cumulative in time. Thus we wish to impose some sort of condition in the spirit of $\mathbb{E}\int_0^t f(\cdot,s)\mathrm{d}s = 0$. However, it is not clear whether this is an appropriate restriction for f, nor is it clear what form f should take. For this reason, while modeling the observation with (8) seems attractive, we have chosen to abandon this model at present due to its lack of mathematical maturity. However, we remark for completeness that the characteristic equations for (8) are very similar to those for (2) assuming smoothness of f (which is not necessarily the case).

As an alternative to the SPDE observation model, we can model noisy observations by first parameterizing the contour $\theta(s) : [0,1] \to \mathbb{R}^2$, and considering the space of all such parameterized contours. By equipping this space with a metric d, say the symmetric Hausdorff distance metric, we can impose a measure on the space of contours. In particular, we can create a Gaussian zero-mean noise ξ by selecting its density to be

$$p(\xi) \sim k e^{-\alpha d(\xi,\mathbf{0})^2} \tag{10}$$

with appropriate constants k, α. Consequently, the observation may be defined by $Y(s) = \theta(s) + \epsilon\xi(s)$, and the likelihood density is $p(Y|\theta) = p(Y - \theta)$.

How can one simulate ξ? In our algorithm, we simulated ξ as an $\mathbb{R}^2$-valued coloured noise on the parameter s:

$$\xi(\omega, s) = \epsilon \sum_{k=1}^{K} \xi_k(\omega)\psi_k(s), \qquad \xi_k = (\xi_k^{(1)}, \xi_k^{(2)}) \tag{11}$$

where $\xi_k^{(i)} \sim \mathcal{N}(0,1)$ are independent and ψ_k is an orthonormal basis for $[0,1]$. We have included a parameter ϵ to control the strength of the noise (or, equivalently, the variance of the $\xi_k^{(i)}$. Again, taking $K \to \infty$ leads ξ to be white noise. It should be checked how closely the density of the numerically generated noise resembles the density in (10).

We have now defined the ways in which the contour and the observations evolve. All that remains is for us to determine some concrete way in relating the state θ to the observation Y. With the model we have assumed in (2), we have the following relation between θ and Y:

$$\begin{aligned} Y(s) &= \theta(s) + \xi(s) && (12) \\ \xi(\omega, s) &= \epsilon \sum_{k=1}^{K} \xi_k(\omega)\psi_k(s), \qquad \xi_k = (\xi_k^{(1)}, \xi_k^{(2)}) && (13) \end{aligned}$$

Recall again that we interpret θ and Y both as $\mathbb{R}^2$-valued contours, and as $\{0,1\}$-valued indicator functions. In the above, we take the former interpretation, and let s denote the normalized arc length of the contour.

We shall use equation (12) in order to determine the conditional probabilities of the observations given the states. Of course, $p(Y|\theta)$ now simply boils down to the distribution of the process ξ.

4 Particle Filter

In order to solve for θ, we shall use a particle filter in the following manner: we shall discretize the time axis by a sequence of times $\{t_n\}_{n=1}^N$. In the algorithm we set $t_{n+1} - t_n$ to be constant, independent of n, but there is no reason to do this other than for simplicity. Let $\theta_n^{(i)} \doteq \theta_{t_n}^{(i)}$ for $i = 1 \dots P$ be a collection of P samples of θ from some known discrete probability density $\pi(\theta_{n-1}|Y_{n-1})$. We assume that we have at our disposal the distributions $\pi(\theta_n|\theta_{n-1})$ and $\pi(Y_n|\theta_n)$. Our algorithm procedes as follows:

- Given $\pi_{n-1}(\theta|Y_{n-1})$, P samples $\theta_{n-1}^{(i)}$ from π_{n-1}, and corresponding velocity predictions $\boldsymbol{w}_{n-1}^{(i)}$ and $\boldsymbol{v}_{n-1}^{(i)}$.

1. Evolve $\theta_{n-1}^{(i)} \to \theta_n^{(i)}$ via some probability law $\pi(\theta_n|\theta_{n-1})$ which depends on $\boldsymbol{v}_{n-1}$ and $\boldsymbol{w}_{n-1}$.

2. Given the observation, Y_n, compute the probabilities $p_i = \pi(Y_n|\theta_n^{(i)})$

$$w_i \leftarrow w_i \frac{p_i \pi(\theta_n^{(i)}|\theta_{n-1}^{(i)})}{\pi(\theta_n^{(i)}|\theta_{n-1}^{(i)}, Y_n)} \approx w_i p_i \tag{14}$$

3. Normalize the importance weights as $w_i \leftarrow \frac{w_i}{\sum_j w_j}$.

4. Endow the distribution $\pi(\theta_n|Y_{n-1})$ with the discrete state space $\theta^{(i)}$ and assign probability w_i to these states. Resample the $\theta^{(i)}$ from this distribution and go back to step 1.

We remark that it is not necessarily most efficient to resample at each step. Instead, one can compute a measure of the effective number of particles from the weights, and then resample based on some decision threshold.

5 Algorithm

It is unfortunately the case that we cannot exactly compute all the quantities needed in order to use the particle filter to estimate a distribution for θ_n. We must thus the following modelling assumptions:

- We do not know $\pi(\theta_n|\theta_{n-1})$. This is tantamount to not knowing the values of $\boldsymbol{v}$ and $\boldsymbol{w}$. However, we must evolve the particles as in step 1 of the particle filter somehow. We propose the following method: In order to get a good indication of the rigid velocity $\boldsymbol{w}$, we use an $L+1$ order least-squares method. I.e. for some number K of previous observations, we compute the centroid of the contour $\boldsymbol{m}_k = \oint_{Y(s)} Y(s)\mathrm{d}s$. Using these approximations of the centroid locations, we use a first-order forward finite-difference to compute velocity approximations:

$$\widetilde{w^i}(t_k) = \frac{m_{k+1}^i - m_k^i}{t_{k+1} - t_k}$$

  We now have a collection of K points for a function $\widetilde{w^i}(t)$. We then fit these points to an Lth order polynomial in the least-squares sense. This polynomial $\widetilde{w^i}(t)$ is used to approximate $w^i(t)$ in the interval $[t_n, t_{n+1}]$. We still need to approximate $\boldsymbol{v}$. Without going into details, we remark that a similar procedure is done with respect to the deformations. I.e. we use a high-dimensional spline, fit in an L_2 sense, to predict the deformation in the interval $[t_n, t_{n+1}]$.
  We also remark that noise is introduced into the predicted values of $\boldsymbol{v}$ and $\boldsymbol{w}$. The variance of the noise is proportional to the L_2 norm of the residual of the least-squares fit. P independent noises of this type are generated and then added to each of the samples.

- The algorithm currently has no rigorous way to compute $\pi(Y_n|\theta_n^{(i)})$ or w_i. At this point, we simply use the asymmetric Hausdorff distance $d(F,G) = \sup_{x\in G} \inf_{y \in F} |x-y|$ for two contours F and G. We set $w_i = \exp\left(-\frac{d(Y^n, \theta_n^{(i)})^2}{k\, A_n}\right)$ where k is some constant and A_n is the area enclosed by the contour Y_n. This admittedly is not the best way to compute this probability and we have listed suggestions for improvement following this algorithm development discussion.

We have now dealt with how quantites are computed in our algorithm. We present the simple formulation of it below:

1. Evolve $\theta_{n-1}^{(i)} \to \theta_n^{(i)}$ using (2) where the $\boldsymbol{w}_{n-1}^{(i)}$ and $\boldsymbol{v}_{n-1}^{(i)}$ are computed using the least-squares method.
2. Compute the 'distance' $d^{(i)} = d(Y_n, \theta_n^{(i)})$ and the area A_n enclosed by the contour Y_n.
3. Let $\theta_n^{(i)}$ form the discrete state space for the conditional density π_n and endow those states with probability $\pi_n(\theta_n^{(i)}|Y_n) = \exp\left(-\frac{(d^{(i)})^2}{k\,A_n}\right)$.
4. Resample $\theta_n^{(i)}$ using the recently calculated density $\pi_n(\theta|Y_n)$. Go back to step 1.

6 Comments on the tracking algorithm using Geometric Active Contours

Rathi et al. proposed a particle filtering algorithm for tracking moving objects based on the Chan-Vese energy functional. In their algorithm, the underlying hidden Markov process is modelled as the pair consisting of an affine transformation A_t, and the edge of the object θ_t. The observation Y is the physical image, whose likelihood function is

$$p(Y|\theta) = e^{-E_{image}(\theta,Y)} \tag{15}$$

where $E_{image}(\theta, Y)$ is the Chan-Vese energy functional.

The parameters of the affine transformation follow the Markov relationship

$$A_{t+1} = B_1 A_t + B_2 A_{t-1} + B_0 \xi_{t+1} \tag{16}$$

with randomness incorporated via $\xi_{t+1} \sim \mathcal{N}(0,1)$. Once A_{t+1} is predicted using (16), the predicted sample θ_{t+1} is then obtained in 2 steps:

1. Evolving/deforming θ_t to more closely match the previous observation Y_t. This is achieved by gradient descent to lower the energy $E_{image}(\theta_t, Y)$.
2. Applying the affine transformation to the deformed θ_t to obtain θ_{t+1}.

The definition of the likelihood function in (15) immediately raises the question of how it is connected to the model relationship $Y = h(\theta) + \xi$. We are unable to mathematically justify its definition, and we note that if one admits this choice of likelihood function, then one can equally well use any energy functional. Depending on the type of image characteristics one wants to emphasize - sharp edges, or uniformity of intensity within the object - the model relationship is modelled such that $p(Y|\theta)$ is a decreasing function of $E(\theta, Y)$, for appropriate energy functional E.

It is instructive to check the affine transformation model against our SPDE model. As a first step, we consider a simple case of equation (2) using one mode of $\boldsymbol{w}$ in the sine basis,

$$\frac{\partial \theta}{\partial t} = -\boldsymbol{w}(t) \cdot \nabla\theta, \qquad \boldsymbol{w}(t) = \xi_k \sin(kt) \tag{17}$$

$$\theta(0,x) = \theta_0(x) \tag{18}$$

where $\xi_k \sim \mathcal{N}(0,1)$. The solution to (17) is given by $\theta(t,x) = \theta_0(X_0^{t,x})$, where the characteristic equation and solution are

$$\frac{\mathrm{d}X_s^{t,x}}{\mathrm{d}s} = \xi_k \sin(ks), \qquad X_t^{t,x} = x \tag{19}$$

$$X_s^{t,x} = \frac{\xi_k}{k}(\cos kt - \cos ks) + x \tag{20}$$

Since we are interested in finding an affine relationship between θ are time step t and $t + \Delta t$, we let $A : x \mapsto \alpha x + \beta$ be an affine transformation and seek the relationship $\theta(t + \Delta t, x) = \theta(t, Ax)$. Simple calculations yield

$$
\begin{aligned}
X_0^{t,Ax} &= \frac{\xi_k}{k}(\cos kt - 1) + \alpha x + \beta && (21)\\
X_0^{t+\Delta t,x} &= \frac{\xi_k}{k}\big(\cos k(t+\Delta t) - 1\big) + x \\
&\approx \frac{\xi_k}{k}(\cos kt - 1) + x - \xi_k k \Delta t \sin kt + \xi_k \mathcal{O}(\Delta t^2) && (22)
\end{aligned}
$$

by using the approximation from the Taylor series expansion, $\cos k(t + \Delta t) = \cos kt - k^2 \Delta t \sin(kt) + \mathcal{O}(\Delta t^2)$. Thus we should take $\alpha = \boldsymbol{I}$ and $\beta = -\xi_k k \Delta t \sin kt + \xi_k \mathcal{O}(\Delta t^2)$. If we choose $t_i = i\Delta t$, then again using the approximation $\sin k(t + \Delta t) = \sin(kt) + \mathcal{O}(\Delta t)$, we calculate

$$
\begin{aligned}
\beta_{t_{i+1}} &= -\xi_k k \Delta t \sin kt + \xi_k k \Delta t \mathcal{O}(\Delta t) + \xi_k \mathcal{O}(\Delta t^2) \\
&= \beta_{t_i} + \xi_k k \mathcal{O}(\Delta t^2)
\end{aligned}
$$

Hence we have a Markov process on the affine parameter β_{t_i}, with a Gaussian transition kernel. If we further assume Δt to be sufficiently small, then we can write down a specific computable (albeit now an approximate) recursive dependence for β_{t_i}:

$$
\beta_{t_{i+1}} = \beta_{t_i} + \xi_k k^3 \Delta t^2 \cos kt_i, \qquad \xi_k \sim \mathcal{N}(0,1) \tag{23}
$$

Two points of note:

(1) A similar analysis also works for a *finite* expansion of $\boldsymbol{w}$ with sufficiently small Δt. In this case, (23) simply becomes

$$
\beta_{t_{i+1}} = \beta_{t_i} + \sum_{k=1}^{M} \xi_k k^3 \Delta t^2 \cos kt_i, \qquad \xi_k \sim \mathcal{N}(0,1), \quad k = 1, \ldots, M
$$

Note the noise term's cubic dependence on k. From here we see the impact of the highly erratic behaviour caused by the high modes.

(2) Because $\boldsymbol{w}$ only encodes for irrotational rigid motion, only the translational term β_t changes in time while the linear term $\alpha = 1$ remains constant. Analysis for the deformation noise $\boldsymbol{v}$ is considerably more difficult, due to the more intractable form of the characteristic ODE

$$
\frac{\mathrm{d}X_s^{t,x}}{\mathrm{d}s} = \boldsymbol{v}(X_s^{t,x}, s) = \xi_{k,l} \sin(ks) \sin(lX_s^{t,x})
$$

It is likely that the affine model alone will not be able to fully account for random deformations caused by the $\boldsymbol{v}$ term. Hence the curve evolution step may prove essential in the prediction step.

7 Further Work

There are many improvements that can be made to this algorithm. To begin with, we are not sure if the models assumed in (2) and (8) are valid ones. Perhaps it is not the case the proper model is a colored/white noise for the advection. A more troubling thought is that if we use (8) to generate observations and we insist f to have nonsmooth form, then we can no longer blindly use the method of characteristics. A more thorough study of how well this model reflects real-life moving objects should be conducted, especially in the context of images and not just contours.

In addition, the prediction step used to predict the velocities can be generalized. The affine transformation model seems to provide a method that works for a simple case of the SPDE model, and also in the numerical experiments conducted by Rathi et al. But as noted above, the affine model alone may not be able to account for the random deformations caused by the $\boldsymbol{v}$ term. More work needs to be done to understand how integrating the curve evolution step before applying the affine transformation can account for the random deformations caused by the $\boldsymbol{v}$ term.

The method we have presented for the velocity prediction uses a least-squares polynomial fit. The way in which this fit is done is not the most robust, and can be improved. Also, the current methods used to predict the velocities are not very robust to perturubations in rigid motion. We can observe this in numerical experiments and we note that there are good ways to mitigate this problem which we have not yet explored.

Also, there remains the problem of defining a noisy contour in a sensible way. In (8), it is not clear how to justify the form for the noise term f, or whether it is more appropriate to write the noise as a perturbation on the tangent space of the contour θ. In the alternative model described by (10), the obvious problem is that the resulting contour no longer maintains a simple non-self-intersecting curve, even if one were to restrict the perturbation to the tangent space of the curve. (In the algorithm, we had overlooked this problem by setting the noise so small that the resulting observation, while intersecting itself at points, still has a discernable interior.)

Also, we are not clear on how appropriate the model we have presented in equations (10) and (11) are. We are unsure if that is the correct probability density to use, or if there are better options.

One of the major drawbacks of the algorithm we've presented is the lack of rigor in the calculation of the importance weights w_i. The formula given by equation (14) should be implemented, but it is unclear how to proceed in evaluating the distributions in that expression. They depend on some of the distributions that are created by the SPDE, and these need to be studied further.