

Stat 992: Lecture 31

Curve Modeling IV.

Moo K. Chung mchung@stat.wisc.edu

April 11, 2004

1. *Modeling curves.* We show how to simulate realistic curve. We propose the following model of curve. Let $X = (x(t), y(t)) \in \mathbb{R}^2$ be the arclength parameterization of curve C .

$$x(t) = \mu_x(t) + \epsilon_x(t) + w_x(t),$$

$$y(t) = \mu_y(t) + \epsilon_y(t) + w_y(t).$$

where ϵ_x, ϵ_y are independent zero mean Gaussian random fields and w_x, w_y are independent Gaussian white noises which is again Gaussian random fields. This is a standard modeling technique in dependent time series and fMRI signals. So we may let $\epsilon_x + w_x \rightarrow \epsilon_x$ and $\epsilon_y + w_y \rightarrow \epsilon_y$. There is a couple of ways to estimate the mean coordinate functions μ_x and μ_y . See principal curve modeling in Hastie and Tibshirani's *The Element of Statistical Learning* (shape analysis version of PCA). For our simple minded example, we estimate them via iterated-kernel or diffusion smoothing, i.e. $\hat{\mu}_x = K_{\sigma_x} * x$ and $\hat{\mu}_y = K_{\sigma_y} * y$. However, if the bandwidth is too large there will be the problem of shrinkage. See Taubin's seminal paper on the shrinkage effect of kernel smoothing. Now we take $\hat{\mu}_x, \hat{\mu}_y$ to be the mean coordinate functions. If there are multiple subjects, we need to perform a curve registration to estimate the mean shape.

```
cx=[x(171);x;x(1)]
cy=[y(171);y;y(1)]
for j=1:10
    for i=2:172
        nbr_dist=[dl(i-1) 0 dl(i)];
        W= K(0.2,nbr_dist);
        tx(i-1) = dot(cx((i-1):(i+1)),W);
        ty(i-1) = dot(cy((i-1):(i+1)),W);
    end;
    cx=[tx(171) tx tx(1)];
    cy=[ty(171) ty ty(1)];
end;
```

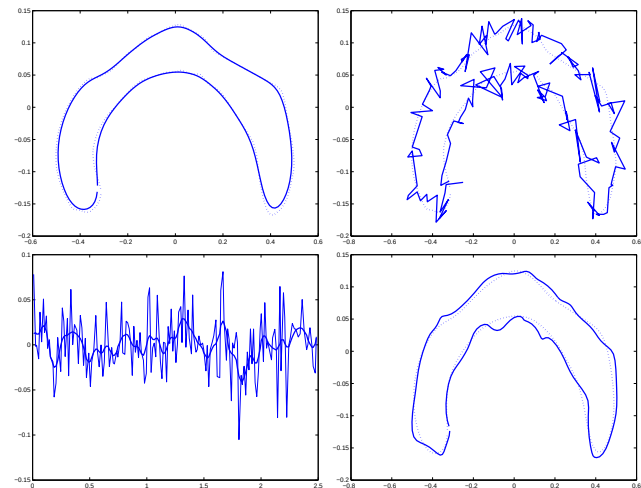


Figure 1: Top left: iterated-kernel smoothing ($\sigma = 0.2, m = 10$). Top right: white noise added curve. The white noises have been exaggerated. Bottom left: Iterated-kernel smoothing of x-component of noise ($\sigma = 0.1, m = 5$). Bottom right: the dotted line (μ_x, μ_y), the bold line is a simulated curve.

2. *Simulating curves.* In order to simulate curves, we need to simulate correlated noise which follows Gaussian random fields. If we simply add independent Gaussian noise to the coordinates $w_x(t_i) \sim N(0, \sigma_1^2)$ and $w_y(t_i) \sim N(0, \sigma_2^2)$ at each point t_i the curve may not be topologically correct (Figure 1). So we add correlated noise.

$$\epsilon_x(t) = K * w_x(t), \epsilon_y(t) = K * w_y(t).$$

Then simulated curve is given by

$$x(t) = \hat{\mu}_x + K * w_x, y(t) = \hat{\mu}_y + K * w_y.$$

Note that $\mathbb{E}x = \hat{\mu}_x, \mathbb{E}y = \hat{\mu}_y$. These guarantee $\hat{\mu}_x$ and $\hat{\mu}_y$ are in fact the mean coordinate functions.

Generalizing this idea, smoothing and simulating random surfaces follow similarly (Figure 3). Spline-based smoothing and simulation can be performed as well (Figure 2). But most 2D curve and surface smoothing methods has the shrinkage problem when the bandwidth is large.

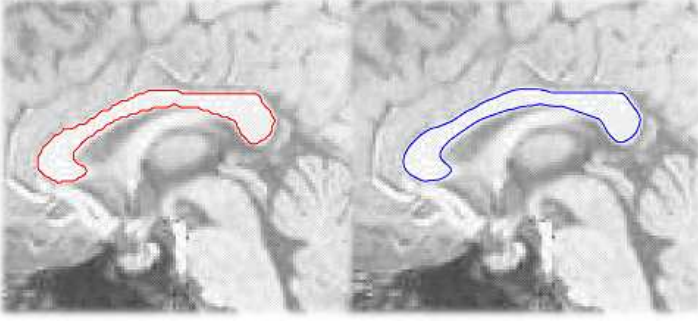


Figure 2: Spline smoothing by Thomas Hoffmann.

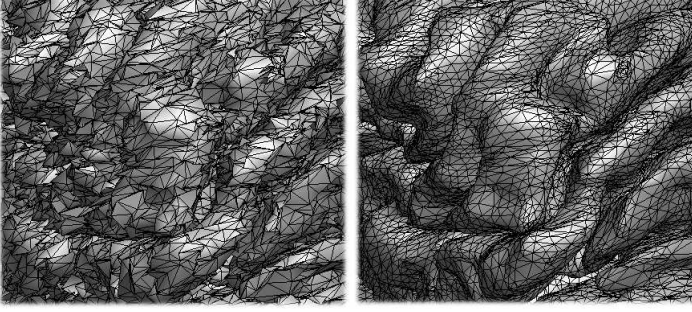


Figure 3: Iterated-kernel smoothing applied to noisy surface.

3. *Different method.* The above approach uses the fixed Cartesian coordinate system to add noise components. A different approach is to add noises along the normal direction. For the arclength parameterization $X(t) = (x(t), y(t))$, $X'(t)$ and $X''(t)$ give the unit normal and tangent vectors respectively. The normalized tangent vector is $N(t) = X''(t)/\|X''(t)\|$. The Cartesian coordinate representation is

$$N(t) = \frac{(x''(t), y''(t))}{\sqrt{x''(t)^2 + y''(t)^2}}.$$

If t is interpreted as time, X, X', X'' can be viewed as the position, velocity and acceleration of a particle. Estimating normal vectors in both noisy curves and surfaces is a nontrivial problem. This is equivalent to the second order derivative estimation which is extremely sensitive to noise. In computer vision literatures, there is a normal vector and curvature estimation technique called *vector voting* or *tensor voting*.

For given mean curve $\mu(t) = (\mu_x(t), \mu_y(t))$, let $N(t) = (\cos \theta(t), \sin \theta(t))$ be the normal vector of $\mu(t)$. Add white noise $\|w\| \sim N(0, \sigma_N^2)$ along the direction of $N(t)$. Then we are adding random vector $w = (w_x, w_y) = (\|w\| \cos \theta(t), \|w\| \sin \theta(t))$. Note $w_x \sim N(0, \sigma_N^2 \cos^2 \theta(t)), w_y \sim$

$N(0, \sigma_N^2 \sin^2 \theta(t))$. Unlike the previous approach, w_x and w_y are correlated. Assuming the mean curve gives somewhat smooth normal vector, we add noise by applying iterated-kernel smoothing on $\|w\|$. So our stochastic curve model is

$$x(t) = \mu_x(t) + \cos \theta(t) K_\sigma * \|w\|(t),$$

$$y(t) = \mu_y(t) + \sin \theta(t) K_\sigma * \|w\|(t).$$

Again this also gives $\mathbb{E}x = \mu_x, \mathbb{E}y = \mu_y$.

Problem 45. Based on the above method of adding noise in the normal direction, simulate a couple of corpus callosum curves.