

Stat 992: Lecture 40

Image Registration II.

Moo K. Chung mchung@stat.wisc.edu

December 5, 2003

This method is still under formulation. :)

1. *Function matching by iterative shifting.* We present a curve matching methodology that can be extended in higher dimension. Let $f(x)$ and $g(x)$ be two functions from $\mathbb{R}$ to $\mathbb{R}$. Suppose that the two functions are similar in shape such that

$$f(x) \approx g(x + u(x))$$

where u is the displacement function. The reasonable stochastic model would be

$$f(x) = g(x + u(x)) + \epsilon_0(x)$$

where ϵ_0 is zero mean Gaussian fields. Linearize $g(x + u(x))$ via Taylor expansion:

$$g(x + u(x)) = g(x) + g'(x)u(x) + O(u^2(x))$$

We assume small displacement $u(x)$ otherwise algorithm based on linear approximation may fail. We have

$$f(x) - g(x + u(x)) \approx f(x) - g(x) - g'(x)u(x) = \epsilon(x)$$

where ϵ is the stochastic noise and error of linear fit combined together. At each point x , u is estimated by

$$u_0(x) = \frac{g(x) - f(x)}{g'(x)}. \quad (1)$$

To stabilize the algorithm, we use $K'_\sigma(x) * g(x)$ in the denominator so that

$$u_0(x) = \frac{K_{\sigma_0} * (g(x) - f(x))}{K'_{\sigma_0}(x) * g(x)}.$$

So our first estimation is $u_0(x) = \frac{K_{\sigma_0} * (g(x) - f(x))}{K'_{\sigma_0} * g(x)}$. Based on this initial guess, we shift function $g(x)$ to $g(u(x) + u_0(x))$ and perform the above procedure again and get the new estimate

$$u_1(x) = \frac{K_{\sigma_1} * (g(x + u_0(x)) - f(x))}{K'_{\sigma_1} * g(x + u_0(x))}.$$

The total amount of displacement is $u_0 + u_1$. Following this idea, we have iterative shifting algorithm

$$u_{n+1}(x) = \frac{K_{\sigma_n} * (g(x + u_0 + u_1 + \dots + u_n) - f(x))}{K'_{\sigma_n} * g(x + u_0 + u_1 + \dots + u_n)}.$$

We decrease the scale at each iteration

$$\sigma_0 > \sigma_1 > \sigma_2 \dots$$

to have a coarse-to-fine approach. So the final estimation is

$$\hat{u} = u_0 + u_1 + u_2 + \dots$$

In order for the algorithm to work, we need to show the convergence of the iteration. If we have $|u_{n+1}| \leq c|u_n|$, $0 < c < 1$, the algorithm will converge due to contraction.

2. *Improvement via least squares estimation.* The estimation (1) seems nice except it may not be very stable since it is based on a single data point. A better estimation would be to use a least squares estimation of u requires a couple of points around x . Suppose we use three points x_- , x , x_+ and set up a system of linear equations

$$f(x_i) - g(x_i) - g'(x_i)u(x_i) = \epsilon(x_i).$$

We minimize the sum of squared residual of the 1st order Taylor expansion:

$$u_0 = [g'(x)]^- [f(x) - g(x)]$$

where $[g'(x)] = (g'(x_-), g'(x), g'(x_+))'$ denotes a vector and similarly for $[f(x) - g(x)]$ and $^-$ is the generalized Moore-Penrose inverse. It is trivial to show that for unit column vector V , $V^- = V'$. Hence $[g'(x)]^- = [g'(x)]^T / \|g'(x)\|^2$. This gives an weighted average estimation of the displacement. Again the derivative g' may not be well defined so we take the derivative of kernel smoothing. So we estimate

$$u_0(x) = [K_{\sigma_0} * g'(x)]^- [K_{\sigma_0}(f(x) - g(x))].$$

Note that this is a weighted least squares estimation (0-th order weighted polynomial regression with Gaussian weight).

Similar coarse-to-fine *multiscale* approaches have been used in many image registration algorithms. It basically saves a lot of computational time. Given data Y , multiscale data can be defined as the collection of data

$$\{K_\sigma * Y : \sigma \in \mathbb{R}^+\}.$$

Since the scale σ can vary continually, the multiscale representation of image can contain global coarse-to-fine geometric features. For instance $K_0 * Y = Y$ is the original unsmoothed data while $K_\infty * Y = \bar{Y}$ (assuming data is centered) is the mean image. The advantages of the multiscale data representation are the differentiability, compactness and reduction on computational time.

3. *Derivative matching.* It is also possible to formulate curve matching by matching higher order derivatives. It may give more smooth displacement. This idea has been implemented by Thomas Hoffmann via the dynamic time warping algorithm (Wang, K. and Gaser, Y. Alignment of Curves by Dynamic Time Warping. Annals of Statistics, 25:1251-1276, 1997).

Following the previous method for matching the curve functions,

$$f'(x) = g'(x) + g''(x)u(t) + \epsilon(t)$$

Afterwards it follows similarly.

The similarity measure we have used is based on function difference. In the case of higher dimensional image data, it is intensity difference. If we use more complex similarity measure like mutual information, the problem becomes nonlinear optimization. Other people used basis function expansion approach such as polynomial basis: $u(x) = \sum_{i=1}^p \beta_i x^i$ (AIR) or cosine basis (SPM'99).