

Image Alignment With Thin-Plate Splines

NELS GREVSTAD

Department of Statistics, Purdue University, West Lafayette, IN 47907-1399, USA



ABSTRACT

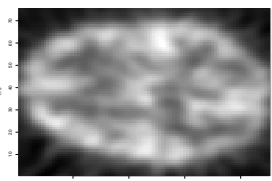
To compare medical images of the same subject taken at slightly different orientations, the first step is to align the images. In this article, an approach to such image alignment using thin-plate splines is presented. The pixels are assumed to be noisy samples of a underlying smooth function on continuous domains, and the aim is to warp the coordinates of one image into those of the other; no pixel imputation is performed as in popular image alignment algorithms that work on discrete domains. Among issues addressed are the isotropic invariance of the approach, the computation, and the smoothing parameter selection. Simulated and real data examples are used to illustrate the technique.

This work is not to be confused with the thin-plate spline warping algorithms appearing in recent literature on the modelling of biological shape deformations, which operate by interpolating a small number of selected landmarks.

1 Introduction

1.1 Human Brain Imaging

- Grayscale images used to locate 'activated' regions of the brain by comparing images under different experimental conditions.
- Modalities
 - Positron Emission Tomography (PET) images depict cerebral blood flow.
 - Functional Magnetic Resonance Imaging (fMRI) images depict blood oxygenation levels.
- Typically, pixelwise t or F statistics are computed, and significant 'activations' are identified based on the resulting t or F image using one of several methods for determining a threshold value.



1.2 Misaligned Images

- The multiple images are generally misaligned:
 - Due to orientation variations between scans
 - Due to anatomical variations between individuals
- Various alignment methods exist:
 - Landmark based methods match anatomically homologous points (e.g. Bookstein's Thin-Plate Warp).
 - Non-landmark based methods minimize of some measure of misalignment (e.g. the least squares approach of Frackowiak, et al.).

2 Background and Formulation

2.1 Thin-Plate Splines

- A thin-plate spline is the minimizer of the functional

$$\frac{1}{n} \sum_{i=1}^n (Y_i - f(x_i))^2 + \lambda J_m^d(f), \quad (1)$$

where $x_i = (x_{i(1)}, \dots, x_{i(d)}) \in (-\infty, \infty)^d$ and

$$J_m^d(f) = \sum_{\alpha_1 + \dots + \alpha_d = m} \frac{m!}{\alpha_1! \dots \alpha_d!} \int \dots \int \left(\frac{\partial^m f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} \right)^2 dx_{(1)} \dots dx_{(d)}$$

- The null space \mathcal{N}_J of $J_m^d(f)$ consists of polynomials of up to $(m-1)$ total order and has dimension $M = \binom{d+m-1}{d}$.
- The functional $J_m^d(f)$ is invariant under shift and rotation of the coordinates, so thin-plate splines are very well suited for the estimation of 2-D and 3-D spatial functions.
- The minimizer of (1) in a space $\mathcal{H} = \{f : J_m^d(f) < \infty\}$ has an expression

$$f(x) = \sum_{\nu=1}^M d_\nu \phi_\nu(x) + \sum_{i=1}^n c_i R_J(x_i, x). \quad (2)$$

where $\{\phi_\nu\}_{\nu=1}^M$ is a basis of the null space \mathcal{N}_J of $J_m^d(f)$.

2.2 Approximation and Efficient Computation

- With the expression (2), the computation reduces to the calculation of the coefficients d_ν and c_i , which takes $O(n^3)$ flops.
- To ease the computation for n large, one may take a subset $\{z_j\}_{j=1}^q \subset \{x_i\}_{i=1}^n$ and minimize (1) in the space $\mathcal{N}_J \oplus \text{span}\{R_J(z_j, \cdot)\}$, which takes $O(nq^2)$ flops; see, e.g., Gu, Kim, 2001.
- Let S be the $n \times M$ matrix with the (i, ν) th entry $\phi_\nu(x_i)$, R be the $n \times q$ matrix with the (i, j) th entry $R_J(z_j, x_i)$, and Q be the $q \times q$ matrix with the (j, k) th entry $R_J(z_j, z_k) = J_m^d(R_J(z_j, \cdot), R_J(z_k, \cdot))$. Minimizing (1) for

$$f(x) = \sum_{\nu=1}^M d_\nu \phi_\nu(x) + \sum_{j=1}^q c_j R_J(z_j, x),$$

one solves for coefficients \mathbf{d} and \mathbf{c}

$$\min (\mathbf{Y} - \mathbf{S}\mathbf{d} - \mathbf{R}\mathbf{c})^T (\mathbf{Y} - \mathbf{S}\mathbf{d} - \mathbf{R}\mathbf{c}) + n\lambda \mathbf{c}^T \mathbf{Q}\mathbf{c}. \quad (3)$$

2.3 Image Alignment

- Model
 - Consider two images, one with pixel values Y_i at pixel locations x_i , $i = 1, \dots, n$, the other with pixel values \tilde{Y}_j at pixel locations \tilde{x}_j , $j = 1, \dots, \tilde{n}$
 - Model the images as noisy samples of smooth functions on continuous domains, $Y_i = g(x_i) + \epsilon_i$, $\tilde{Y}_j = f(\tilde{x}_j) + \tilde{\epsilon}_j$.
 - Assume that $g(x) = f(\eta(x))$, where η maps the coordinates x of one image to \tilde{x} of the other.
- Estimation of the mapping η : two stage procedure.
 - First, $f(\tilde{x})$ is estimated using a standard thin-plate spline. with the smoothing parameter λ selected by say the generalized cross-validation of Craven and Wahba.
 - Given the estimate of $f(\tilde{x})$, η can be estimated by minimizing

$$\frac{1}{n} \sum_{i=1}^n (Y_i - f(\eta(x_i)))^2 + \lambda_1 J_m^d(\eta_{(1)}) + \dots + \lambda_d J_m^d(\eta_{(d)}); \quad (4)$$

- It can be shown that (4) is isotropically invariant with respect to \tilde{x} .

3 Computation

- Our primary concern is the computation of (4).
- Given $f(\tilde{x})$ and the expressions

$$\eta_{(k)} = \sum_{\nu} d_{\nu(k)} \phi_\nu + \sum_j c_{j(k)} R_J(z_j, \cdot), \quad (5)$$

the least squares term in (4) can be minimized using Gauss-Newton iteration.

- To perform the iteration, replace $f(\eta(x_i))$ in (4) with its first-order Taylor expansion at the current estimate.
- Analytical formulas of $\partial f / \partial \tilde{x}$ exist, but are messy.
- The Gauss-Newton iteration typically converges in 10-15 steps.

4 Smoothing Parameter Selection

4.1 Cross-Validation

- For a given λ Would like to assess the performance of the estimate η_λ by a mean square error loss at the pixel points

$$L(\lambda) = \frac{1}{n} \sum_{i=1}^n \{f(\eta_\lambda(x_i)) - f(\eta_0(x_i))\}^2, \quad (6)$$

where η_0 is the true mapping.

- Fixing λ , on convergence of the Gauss-Newton iteration, define the "fitted value" $\hat{\mathbf{Y}} = \sum_{k=1}^d G_k (\mathbf{S}\mathbf{d}_{(k)} + \mathbf{R}\mathbf{c}_{(k)}) = \sum_{k=1}^d R_k \mathbf{b}_{(k)}$, where G_k are diagonal matrices with diagonals $\hat{g}_{i(k)} = \partial f / \partial \tilde{x}_{i(k)}|_{\tilde{x}=\eta_\lambda(x_i)}$, $\hat{\eta}(x)$ being the current estimate of $\eta(x)$, and write $\hat{\mathbf{Y}} = \mathbf{A}(\lambda)\hat{\mathbf{Y}}$.

- A cross-validation score can be defined by

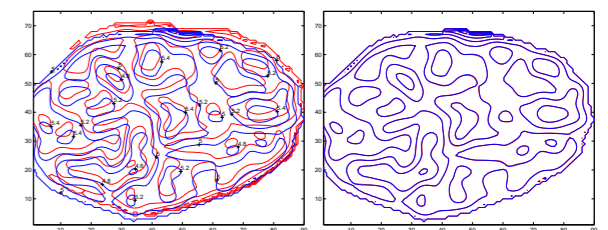
$$V(\lambda) = \frac{\hat{\mathbf{Y}}^T (\mathbf{I} - \mathbf{A}(\lambda)) \hat{\mathbf{Y}}}{\{n^{-1} \text{tr}(\mathbf{I} - \mathbf{A}(\lambda))\}^2}; \quad (7)$$

- In simulations performed so far, it $V(\lambda)$ tracks $L(\lambda)$ well, with the minimizer of $V(\lambda)$ delivering near optimal λ for use in (4).

5 Synthetic Examples

5.1 Alignment to Known $f(\tilde{x})$

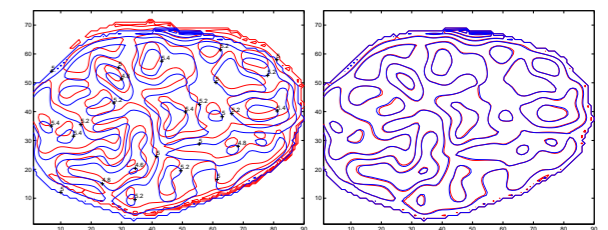
- In this first example, we assess the performance of the alignment procedure in the case $f(\tilde{x})$ is a known function.
- The known function $f(\tilde{x})$ is generated by fitting a thin-plate spline to the log of the $n = 4492$ pixel values of a 2D PET image of a subject under resting condition.
- We then generate the second image Y_i by adding noise to $g(x_i) = f(\eta_0(x_i))$, $i = 1, \dots, n$, where $\eta_0(x)$ consists of a translation plus slight bending of the x domain.
- The figures below show the results of warping $f(\tilde{x})$ to the simulated pixel values Y_i via an the estimate of $\eta(x)$ using our method described above, with the subset $\{z_j\}_{j=1}^q$ is of size $q = 94$.



The figure on the left shows the contours of $f(\tilde{x})$ (red lines) superimposed on $g(x) = f(\eta_0(x))$ (blue lines). On the right, contours of $f(\hat{\eta}(x))$ (red), with $\hat{\eta}(x)$ the estimate of $\eta_0(x)$ using our method, superimposed on $g(x)$ (blue).

5.2 Alignment to Estimate of $f(\tilde{x})$

- In this second example, we assess the performance of the alignment procedure in the case $f(\tilde{x})$ is estimated with a standard thin-plate spline $\hat{f}(\tilde{x})$ from a simulated PET image generated by adding noise to the $f(\tilde{x})$ described in section 5.1.
- Using the same Y_i described in section 5.1, the figures below show the results of warping $\hat{f}(\tilde{x})$ to the pixel values Y_i via an the estimate of $\eta_0(x)$ using our method.



The figure on the left shows the again the contours of $f(\tilde{x})$ (red lines) superimposed on $g(x) = f(\eta_0(x))$ (blue lines). On the right, contours of $f(\hat{\eta}(x))$ (red), with $\hat{\eta}(x)$ the estimate of $\eta_0(x)$ using our method, superimposed on $g(x)$ (blue).

6 Conclusions

- In simulations so far, our method performs admirably.
- It has the desirable property of isotropic invariance.
- It does not require identification of landmarks, i.e. it's non-label based alignment. This eliminates the the error associated with misidentification of the landmarks.
- The problem of smoothing parameter selection is solved.
- However, many more simulations are needed to further assess it's performance, and we are continuing in this direction.

References

- Bookstein, F. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations, *IEEE Trans. Pattern Anal. and Mach. Intell.*, Vol11, No. 6, 567-585, June 1989.
- Craven, and Wahba, G. Smoothing Noisy Data with Spline Functions: Estimating the degree of smoothing by cross-validation, *Numer. Math.* 31, 377-403, (1979).
- Frackowiak, Friston, et al., *Human Brain Function*, Academic Press, (1997).
- Gu, *Smoothing Spline Anova Models*, Springer-Verlag, (2002).
- Gu, Kim, Penalized Likelihood Regression: General Formulation and Efficient Computation, *Can. J. Statist.* 29, (2002).