Curve Registration by Local Regression

Alois Kneip, l'Université Catholique de Louvain
Xiaochun Li, Eli Lilly Co.
Brenda MacGibbon, l'Université du Québec à Montréal
J. O. Ramsay, McGill Unversity

June 12, 1998

The research of the first author was supported by the contract Projet d'Actions de Recherche Concertées (PARC No. 93/98-164) of the Belgian Government. This research was also supported by grant A320 from the Natural Sciences and Engineering Research Council of Canada, and was completed while the second author was a Postdoctoral Fellow at the Centre de Recherches Mathématiques and the Institut des Sciences Mathématiques de Montréal.

Abstract

Functional data analysis involves the extension of familiar statistical procedures such as principal components analysis, linear modeling, and canonical correlation analysis to data where the raw observation x_i is a function.

An essential preliminary to a functional data analysis is often the registration or alignment of salient curve features by suitable monotone transformations h_i of the argument t, so that the actual analyses are carried out on the values $x_i[h_i(t)]$. In effect, this conceptualizes variation among functions as being composed of two aspects: phase and amplitude. Registration aims to remove phase variation as a preliminary to statistical analyses of amplitude variation

A local nonlinear regression technique is described for identifying the smooth monotone transformations h_i, and is illustrated by analyses of simulated and actual data.

Keywords: time warping, dynamic time warping, monotone functions

1 Introduction

An observation $x_i(t)$ in functional data analytic situations is a curve, and as such displays a sequence of events such as peaks, valleys, zero crossings, flat regions, and so forth. These events can vary in two ways: amplitude and phase or timing, and we normally see both types of variation. Consider, for example, the 50 curves displayed in the top panel of Figure 1. These are estimated magnitudes of the acceleration vector of the pen tip in a study of handwriting (Ramsay, 1998b). Each curve has around 50 sharp peaks separated by deep valleys approaching zero acceleration. But it is difficult to disentangle them because phase and amplitude variation are mixed.

If we want to get a good estimate of average amplitude, phase variation becomes a problem. In the second panel of Figure 1 we see the pointwise average of the curves, along with a randomly selected individual curve. The average has lost a great deal of the variability of individual curves, and consequently is a poor descriptor of amplitude. This is due to the average at any time t being taken over curves where different events are taking place.

Because of these and similar problems, the registration or alignment of salient curve features by suitable monotone transformations $h_i(t)$ of the argument is essential. Subsequent analyses are then carried out on the curve values considered as a function of $h_i(t)$, that is, $x_i \circ h_i$. This process effectively aims to remove phase variation so that subsequent analyses can focus on amplitude effects.

The classic procedure is marker registration, involving identifying the timing of specified features in the curves and transforming time so that these marker events occur at the same time. A comprehensive reference is Bookstein (1991). Dynamic time warping is a marker registration technique frequently cited in the engineering literature (Sakoe and Chiba 1978), and Kneip and Gasser (1988,1992) studied its statistical aspects. These methods, however, can be sensitive to errors in feature location, and these features may be even missing in some curves. Moreover, substantial phase variation may remain between widely separated markers.

Silverman (1995) developed an alternative technique that did not require markers, and Ramsay and Li (1998) extended Silverman's method by using a flexible smooth monotone transformation family developed by Ramsay (1998a). These approaches involve optimizing a fitting criterion with respect to the parameters defining an entire warping curve $h_i(t)$ for each observation. Numerical optimization methods are used, and consequently these

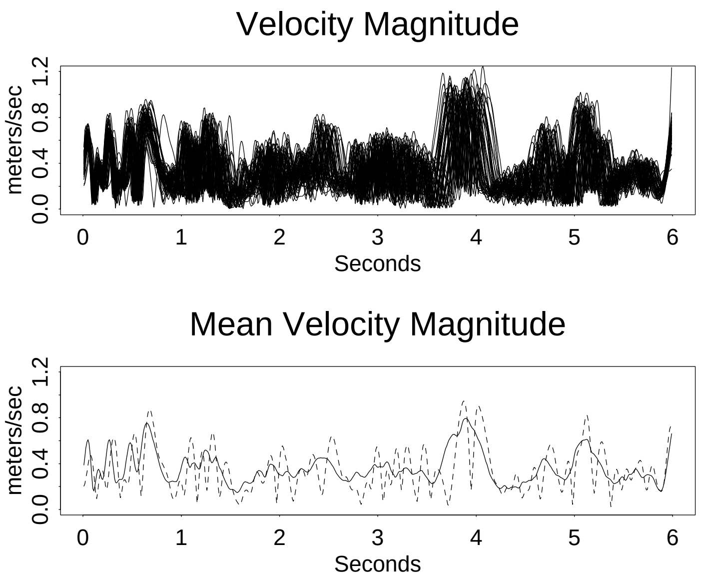

Figure 1: The top panel displays 50 acceleration magnitude functions for pen position in a handwriting experiment. The solid curve in the bottom panel is the cross-sectional mean of these curves, and the dashed line is a randomly selected curve. The average is a poor representation of curve shape because phase variation is confounded with amplitude variation.

approaches can be somewhat complex to program. Moreover, it does seem obvious how these methods can be generalized to multidimensional argument domains

Our goal here to develop a computationally convenient method for curve registration using locally estimated monotone transformations. As is the rule for localized curve estimation methods in general, the needed flexibility can be achieved while keeping the algorithm simple and using routines available in standard matrix calculation packages. Although this paper considers only one-dimensional arguments, we also seek a method that seems capable of generalization to higher dimensions. For an example of registration of three-dimensional brain images, see Collins, Holmes, Peters and Evans (1996).

2 The Algorithm

Let y be a target function to which each of the functions x_i is to be registered. The target may be (i) a selected and representative sample curve, (ii) a preliminary estimate of the average curve to be updated after registration, or (iii) some prototype curve available from some other source. In the following discussion, the subscript i is dropped, so that x refers to an arbitrary curve to be registered to y.

The algorithm proceeds by a series of iterations, such that on iteration ν we seek a warping function $g^{(\nu)}$ such that

$$y \approx x^{(\nu-1)} \circ g^{(\nu)}, \tag{1}$$

the approximation being taken in a sense that is specified below, and $x^{(\nu-1)}$ being the output of the previous iteration, and the algorithm being initialized by setting $x^{(0)} = x$.

Iteration ν consists of two phases, each of which is described in greater detail below, but a short summary of which is

- a cautious estimate of the warping function $g^{(\nu)}$ is constructed by a local regression algorithm, and
- in preparation for the next iteration, an update of $x^{(\nu-1)}$ to the revised estimate $x^{(\nu)} = x^{(\nu-1)} \circ g^{(\nu)(-1)}$, where $g^{(\nu)(-1)}$ refers to the functional inverse of $g^{(\nu)}$ rather than to its reciprocal.

The second update phase replaces the problem (1) by the progressively updated and hopefully increasingly easy problem $y \approx x^{(\nu)} \circ g$. When the iterations converge, we can expect that $g^{(\nu)}(t) = t$ to a satisfactory degree. An initial bandwidth or window width b is chosen for the first local regression step; we tend to start with one quarter of the width of the interval spanned by the arguments. This is subsequently decreased as detailed below. The estimate of the warping function h with respect to the original x on iteration ν is then

$$h = g^{(\nu)} \circ g^{(\nu-1)} \circ \ldots \circ g^{(1)}.$$

2.1 The Local Regression Phase

For simplicity, omit the superscript indicating iteration number ν. Let $\boldsymbol{s} = s_j, j = 1, \ldots, n$, be the sequence of argument values at which we desire the values $g(\boldsymbol{s})$. Denote by $g_j(s)$, the localized warping function in a neighborhood U_j of fixed argument value s_j. The local regression algorithm is based on a small family of very simple and low-frequency models for g_j, along with a local model for the relationship between y and $x \circ g_j$ in this neighborhood. For example, we may use as g_j the positive-slope linear transformation

$$g_j(t) = e^{\gamma_j}(t + \delta_j)$$

and the direct relation

$$y(t) \approx x[g_j(t)] = x[e^{\gamma_j}(t + \delta_j)], \tag{2}$$

over the neighborhood $U_j = [t_j - b, t_j + b]$. Our starting value for b implies that the width of U_j is half of the width of the argument domain.

We fit this model and similar models over U_j by local weighted least squares. Let $\boldsymbol{t} = t_i, i = 1, \ldots, N$ be a reasonably fine sequence of N argument values, and let w_{ij} be a set of weights that vanish outside of U_j. These weights may, for example, be defined by the sequence $(1 - (t_i - t_j)^2/b)_+$. Then, taking first-order expansion of g_j about $\gamma_j = \delta_j = 0$ and dropping higher-order terms yields the linearized least squares problem

$$(\gamma_j, \delta_j) = \text{argmin} \sum_i^N w_{ij}[y(t_i) - x(t_i) - \gamma_j t_i Dx(t_i) - \delta_j Dx(t_i)]^2 \tag{3}$$

Solving this problem for γ_j and δ_j is equivalent to a single iteration of the Gauss-Newton algorithm. This iteration needs to be stabilized to protect it against possible singularity of the stationary equations, so a positive ridge constant Δ can be added to the diagonal entries of the cross-product matrix, yielding

$$\left[\begin{array}{ll} \sum_i w_{ij}[t_i Dx(t_i)]^2 + \Delta & \sum_i w_{ij} t_i [Dx(t_i)]^2 \\ \sum_i w_{ij} t_i [Dx(t_i]]^2 & \sum_i w_{ij} [Dx(t_i)]^2 + \Delta \end{array}\right], \tag{4}$$

thus changing the problem to one of local ridge regression (Hoerl & Kennard, 1970), and consequently smoothing the parameter estimates towards zero. We routinely use $\Delta = N$.

While satisfactory results are possible with (2), we find that extending this model to

$$h_j(t) = e^{\gamma_j}(t + \delta_j) \quad \text{and} \quad y(t) \approx e^{\beta} x[e^{\gamma_j}(t + \delta_j)] \tag{5}$$

gives better results when there is amplitude variation that is sufficiently smooth that it can be approximated locally by a simple change of scale. The corresponding Gauss-Newton step updates the log-scale parameter β as well as the other two. The matrix (4) becomes

$$\left[\begin{array}{lll} \sum_i w_{ij} x(t_i)^2 + \Delta & \sum_i w_{ij} t_i x(t_i) Dx(t_i) & \sum_i w_{ij} x(t_i) Dx(t_i) \\ \sum_i w_{ij} t_i x(t_i) Dx(t_i) & \sum_i w_{ij} [t_i Dx(t_i)]^2 + \Delta & \sum_i w_{ij} t_i Dx(t_i)^2 \\ \sum_i w_{ij} x(t_i) Dx(t_i) & \sum_i w_{ij} t_i Dx(t_i]^2 & \sum_i w_{ij} Dx(t_i)^2 + \Delta \end{array}\right], \tag{6}$$

While one might elaborate this three-parameter model even further by allowing for a local vertical shift as well as a local scale change, our experience indicates that the resulting four-parameter model is too often unstable.

The local regression step can be denoted by the operation

$$\boldsymbol{f}^{(\nu)} = \mathtt{LR}_b[\boldsymbol{s}|\boldsymbol{t}, x^{(\nu-1)}(\boldsymbol{t}), Dx^{(\nu-1)}(\boldsymbol{t})] \tag{7}$$

where $\boldsymbol{f}$ contains the n estimated values of the warping function corresponding to the argument vector $\boldsymbol{s}$.

2.2 The Update Phase

Let the notation

$$\boldsymbol{c} = \mathtt{Sm}_m(\boldsymbol{z}|\boldsymbol{a}, \boldsymbol{b})$$

indicate the operation of interpolating or smoothing the pairs $(\boldsymbol{a}, \boldsymbol{b})$, $\boldsymbol{a}$ and $\boldsymbol{b}$ being assumed to be of the same length, so as to evaluate the derivative of order m at the argument values in $\boldsymbol{z}$. In terms of this notation, the second phase operations that follow the local regression step are

$$\begin{aligned} \boldsymbol{g} &= \mathtt{Sm}_0(\boldsymbol{t}|\boldsymbol{s}, \boldsymbol{f}) \\ \boldsymbol{g}^{(-1)} &= \mathtt{Sm}_0(\boldsymbol{t}|\boldsymbol{g}, \boldsymbol{t}) \\ \boldsymbol{h}^{(\nu)} &= \mathtt{Sm}_0(\boldsymbol{t}|\boldsymbol{g}^{(-1)}, \boldsymbol{h}^{(\nu-1)}) \\ x^{(\nu)}(\boldsymbol{t}) &= \mathtt{Sm}_0(\boldsymbol{t}|\boldsymbol{g}^{(-1)}, x^{(\nu-1)}(\boldsymbol{t})) \\ Dx^{(\nu)}(\boldsymbol{t}) &= \mathtt{Sm}_1(\boldsymbol{t}|\boldsymbol{t}, x^{(\nu)}(\boldsymbol{t})) \end{aligned} \tag{8}$$

The first operation estimates the current warping function g, the second its inverse $g^{(-1)}$, the third updates the global warping function, and the final two update the registered curve and its derivative.

In addition to providing reasonable interpolated values, the `Sm` operator must also have the capability of making these values strictly monotone in the first three steps. We obtained reasonable performance by using the SPLUS linear interpolation function `approx` for $\mathtt{Sm}_0$ and central difference estimates computed from `approx` combined with forward and backward differences for the initial and final values, respectively, for $\mathtt{Sm}_1$. A more sophisticated option making smoothing possible is the Pspline module for SPLUS (Statistical Sciences, 1995) described in Ramsay and Silverman (1997) and available at Web site `www.psych.mcgill.ca/faculty/ramsay.html`. We suggest a light roughness penalty applied to the third derivative. If the output of the $\mathtt{Sm}_0$ function revealed some nonmonotonicity, these values were fitted by an isotonic regression to render them strictly increasing.

2.3 Bandwidth Reduction

This local regression/update algorithm can be repeated with smaller and smaller bandwidths b, and we simply halve the bandwidth for each cycle. However, care must be taken to avoid making the bandwidth so small that either too few points are contained within the interval, or that the two functions are essentially parallel within one or more neighborhoods U_j. In fact, we find that the algorithm is effective for even very large bandwidths, so that small neighborhoods are not really needed. This is consistent with the principle that it is only reasonable to seek warping functions h which vary slowly with respect to the variation in x and y.

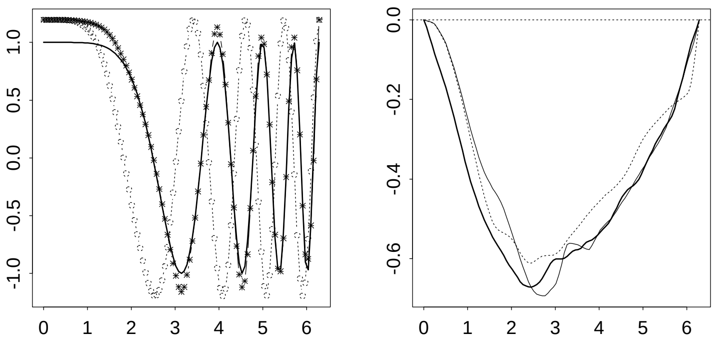

Figure 2: The left panel displays a typical registration problem, along with the results of four local registration iterations with bandwidth $b = \pi/2$. The heavy solid line is the target function $y = \cos(t^3/\pi^2)$, the dotted line with open circles is the function x requiring registration to y. The solid line with stars is the registered function $x \circ h$. Circles and stars indicate argument values t_i. The right panel displays the true warping function as a heavy dotted line in terms of values $h(t) - t$, the estimated values $\hat{h}(t) - t$ after a single block of four iterations as a dotted line, and the estimate after three blocks of four iterations, with bandwidth halving, as the thinner solid line.

2.4 An Illustration

Figure 2 displays a typical problem. The target is $y = \cos(t^3/\pi^2)$, and is displayed as the heavy solid line without symbols in the left panel. The arguments $\boldsymbol{t}$ are 101 equally spaced values spanning $[0, 2\pi]$. The true warping function is displayed as a heavy solid line in the right panel in terms of $h(t)-t$, and we see that the largest amount of warping occurs in the middle of the interval, where the peaks in x and y are becoming progressively sharper. Some amplitude scaling was applied to generate the warped target, so that the function x, plotted with open circles in the left panel, has amplitude as well as phase variation.

Figure 2 also displays the results of four iterations using the initial bandwidth $b = \pi/2$, and a ridge parameter $\Delta = N = 101$. The function $x \circ h$ is plotted with stars, and is now well registered to the target function. The dotted line in the right panel is the estimated warping function, and we see that it has captured the main features of the true function. The estimated warping function after three blocks of four iterations, with the bandwidth halved at the beginning blocks 2 and 3, is shown as a regular solid line in the right panel. We see that the two additional blocks with finer bandwidths have made the estimate very accurate on the right where the presence of sharp features allows the smaller values of b to improve the fit, but the fit deteriorates somewhat in the left half, where the low-frequency variation does not require small bandwidths. We may conclude from this experiment that bandwidth b should be large enough to ensure that the cross-product matrix (6) does not collapse to $\Delta \boldsymbol{I}$, and in practice this means that the two curves are not approximately linear within the region defined by the bandwidth.

3 A Monte Carlo Performance Trial

Although the problem presented above illustrates some of the aspects and difficulties presented by registration situations, we need some indication of how the method works over a wider range of problems. To this end, we generated 100 random problems as follows. Let $\boldsymbol{z}(\sigma)$ indicate a vector of length $N = 101$ random normal deviates sampled from an $N(0, \sigma^2)$ population. Let the operators

1. `Trap` indicate the indefinite integral of this sequence using the trapezoidal rule,

$$\texttt{Trap}(\boldsymbol{z})_i = \Delta_t[\sum_{j=1}^{i} z_j - (z_1 + z_i)/2],$$

   where Δ_t is the difference between adjacent members of an equally spaced sequence $\boldsymbol{t}$,

2. `Detrend` indicate the operation of removing any linear trend from a sequence by linear regression, and

3. `Norm` indicate the operation of linearly transforming a sequence $\boldsymbol{h}$ so that $h_1 = 0$ and $h_N = t_n$.

Then a random function was generated by

$$\boldsymbol{y} = 10^4[\texttt{Detrend} \circ \texttt{Trap}^3][\boldsymbol{z}(1)],$$

where $\texttt{Trap}^3$ implies applying the trapezoidal integration three times. A random warping sequence was generating using a method described in Ramsay (1998a) by

$$\boldsymbol{h} = [\texttt{Norm} \circ \texttt{Trap} \circ \exp \texttt{Trap}][\boldsymbol{z}(5)]$$

and a random amplitude modulation sequence $\boldsymbol{a}$ by

$$\boldsymbol{a} = [\exp \circ 10^2 \texttt{Detrend} \circ \texttt{Trap}^3][\boldsymbol{z}(1)].$$

Finally, the vector $\boldsymbol{x}$ of values of the random carrier function x was

$$\boldsymbol{x} = \boldsymbol{a} * \texttt{Sm}_0(\boldsymbol{t}|\boldsymbol{h}, \boldsymbol{y}),$$

where the product indicated by $*$ is pointwise multiplication. Some idea of the nature of the registration problems that this procedure generates can be gained by inspecting Figure 3, which displays six randomly sampled problems.

The registration was applied in the same manner as for the previous illustration to each of the 100 random problems. We looked at two measures of lack of performance of the algorithm computed in terms of the target sequence and the zero-intercept regression approximation,

$$\begin{aligned} \hat{\boldsymbol{y}} &= a\hat{\boldsymbol{x}} \\ \hat{\boldsymbol{x}} &= \texttt{Sm}_0(\boldsymbol{t}|\hat{\boldsymbol{h}}^{(-1)}, \boldsymbol{x}) \\ a &= (\boldsymbol{y}'\hat{\boldsymbol{x}})/(\hat{\boldsymbol{x}}'\hat{\boldsymbol{x}}). \end{aligned} \tag{9}$$

The first measure is

$$1 - R_1^2 = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 / \|\boldsymbol{y} - \bar{\boldsymbol{y}}\|^2, \tag{10}$$

where $\bar{\boldsymbol{y}}$ is the mean of the y-values. The second performance measure indicates failure to improve the fit to the target offered by the unregistered function:

$$1 - R_x^2 = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 / \|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{11}$$

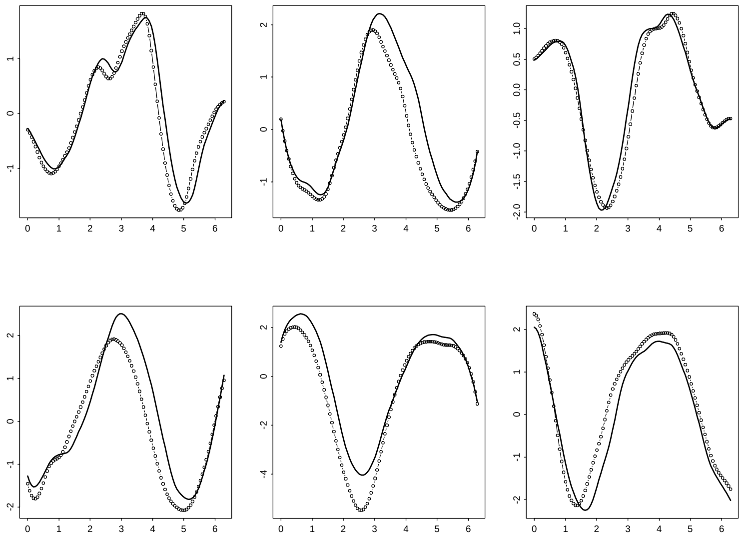

Figure 3: Each panel shows a randomly generated registration problem. The target function is shown as a heavy solid line, and the carrier function to be registered as a dotted line with open circles.

Figure 4 plots these measures against each other. We see that the geometric averages are 0.0023 and 0.017, respectively. Thus, the average congruence between the target and the registered function is what most applications would require. We also see that there are no catastrophic failures among these 100 analyses; the worst measures are 0.028 and 0.43, respectively, and the latter value is in part due to the fact that the unregistered carrier x was already well-registered.

4 Registering Handwriting Velocity

We return now to the curves displayed in Figure 1. A study of handwriting reported in Ramsay (1998b) required a single subject to write the three Chinese characters for the phrase "statistical science" 50 times, and the X-, Y-, and Z-coordinates of pen position were recorded with high accuracy 600 times per second. Here we register the velocity norm records, $\sqrt{DX_i^2 + DY_i^2 + DZ_i^2}$ of each record i, sampled at 601 equally spaced time points. This process required two cycles: In the first, the target function y was the the average

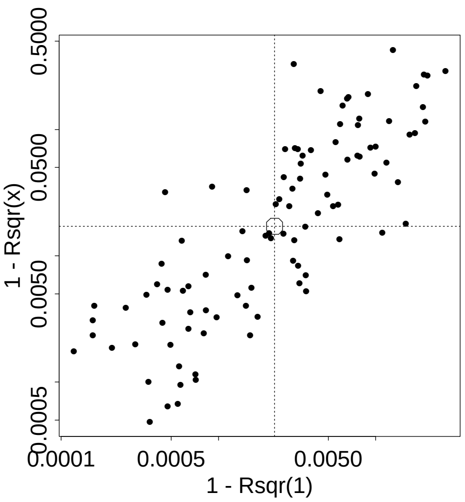

Figure 4: Failure of fit measures (10) and (11) for 100 random registration problems. The open circle and dotted lines indicate the geometric means of these measures.

of these records, and in the second cycle, y was the average of the registered functions computed in the first cycle. The default settings of the algorithm used in the previous illustrations were used here. The warping functions were estimated at 120 equally spaced time points.

Figure 5 displays the registered curves after the first and second cycle. With one or two exceptions, the curves appear well-registered at the end of the second cycle. Further cycles did not result in appreciable change in the degree of registration. The few failures could have been avoided by carrying out an initial marker registration using as landmarks three or four of the largest peaks.

5 Conclusion

These examples and our other experiences with the method encourage us to believe that this local regression algorithm is an effective registration tool that is fairly easy to implement in various computing environments. Our own software, written in a mixture of SPLUS and Fortran 77, is available by

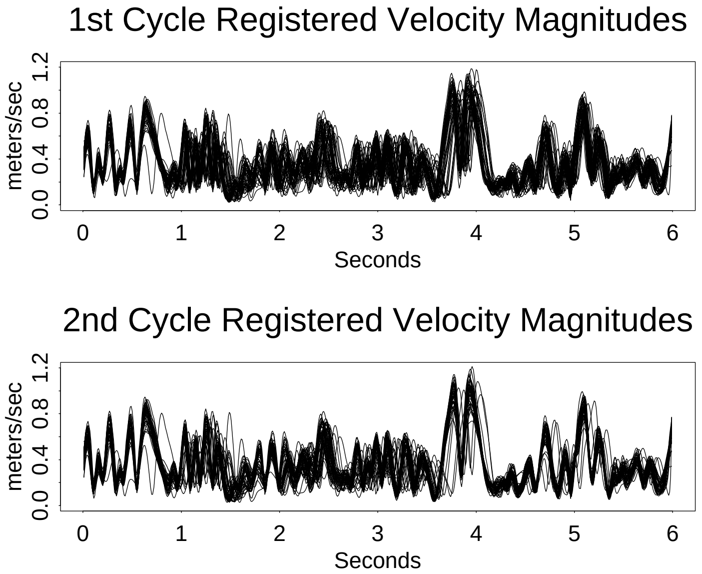

Figure 5: The 50 velocity magnitudes for handwriting in Figure 1 registered to their cross-sectional mean are in the first panel, and the second panel contains the magnitudes registered to the updated cross-sectional mean.

anonymous ftp from `ego.psych.mcgill.ca/pub/ramsay/register`.

We are also working on versions of this algorithm for higher dimensional problems, where the need for convenient and reasonably fast registration methods is acute. The local warping functions (2) and (5) generalize in a straightforward manner to higher dimensional versions.

It seems clear that amplitude and phase variation can only be separated if the latter varies slowly with respect to the former. In many problems, including brain imaging, the amount of variation in amplitude will vary greatly from one part of the domain to another. For example, a slice through the middle of the brain shows high frequency variation in various types of neuro-images in the cortical regions near the outer boundaries of the brain, but only low-frequency variation in central regions, where much of the space if taken up by the ventricular chambers. This suggests that local algorithms will have to vary bandwidths b to adapt to the frequency of amplitude variation.

References

Bookstein, F. L. (1991) *Morphometric Tools for Landmark Data: Geometry and Biology.* Cambridge: Cambridge University Press.

Collins, D. L., Holmes, C. J., Peters, T. M., and Evans, A. C. (1996) Automatic 3-D model-based Neuroanatomical segmentation. *Human Brain Mapping,* **3,** 190-208.

Hoerl, A. E. and Kennard, R. W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics,* **12,** 69-82.

Kneip, A. and Gasser, T. (1988) Convergence and consistency results for self-modeling nonlinear regression. *Annals of Statistics,* **16,** 82-112.

Kneip, A. and Gasser, T. (1992) Statistical tools to analyze data representing a sample of curves. *Annals of Statistics,* **20,** 1266-1305.

Ramsay, J. O. and Li, X. (1998) Curve registration. *Journal of the Royal Statistical Society, Series B,* **60,** 351-363.

Ramsay, J. O. (1998a) Estimating smooth monotone functions. *Journal of the Royal Statistical Society, Series B,* **60,** 365-375.

Ramsay, J. O. (1998b) Functional components of variation in handwriting. McGill University, unpublished manuscript.

Ramsay, J. O. and Silverman, B. W. (1997) *Functional Data Analysis.* New York: Springer-Verlag.

Salkoe, H. and Chiba, S. (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions, ASSP-26,* **1,** 43-49.

Silverman, B. W. (1995) Incorporating parametric effects into functional principal components analysis. *Journal of the Royal Statistical Society,* Series B, **57**, 673–689.