A Model for Signature Verification

Trevor Hastie and Eyal Kishon
AT&T Bell Laboratories,
600 Mountain Ave
Murray Hill, NJ 07974
Malcolm Clark
Monash University
Clayton, Victoria, Australia
Jason Fan
Department of Electrical Engineering
Stanford University, CA 94305

February 5, 1992

Abstract

We propose a statistical model for signature verification by computer. Our model recognizes that repeated signatures by the "owner" are similar but not identical. Our model consists of a *template* signature for each individual, and several factors which allow for variations in each rendition of this template. These variations include the *speed* of writing, as well as slowly varying affine transformations such as size, rotation and shear. The estimated template represents the "mean" of a sample of signatures from an individual, and the variations in the factors can be used to establish several measures of "variance". These quantitative measures are essential for reliable signature verification.

1 Introduction

In this paper we propose a statistical model for human signature recognition and verification by computer. Our model recognizes that repeated test signatures by the "owner" are similar but not identical. The model involves a *template* signature which is fixed for an individual. Each time he signs he traces out this template, not exactly but with small stochastic variations on each occasion. These variations include the speed of writing, as well

as slowly varying affine transformations such as rotation, scale, and shear. While the estimated template serves as the "mean" of a writer's signature, the variations about this mean can be used to establish the "variance".

The signature data is obtained on-line from a digitizer, instrumented pen or similar device. Typically, the output of such a device gives the position (in (X, Y) co-ordinates relative to an arbitrary origin) of the pen at equi-spaced time intervals. Along with each location measurement, our digitizer also records the downward pressure on the pen, or the force exerted on the tip of the pen.

In the prototype system already developed, a database is constructed from say 5 to 10 specimen signatures from each of W writers. For each writer, a template signature is estimated, and for each the variations of their sample signatures about this template are recorded. Given a new signature claiming to belong to writer w, our system compares it to the template signature for writer w. If the variations in the factors are beyond the ranges established for these factors for writer w, the verification fails.

There is an extensive literature on automatic signature verification and writer identification by computer (see [P89] for a recent review).

The methodology for dynamic signature verification falls into two main groups:

- methods that use functions to represent an entire signature. Here the complete signal is represented as a function of time, and the values of the function at different time points are considered to be the features.

- methods that summarize the original signal by a number of low-dimensional parameters or features. These parameters represent the signature for the purpose of representation and recognition.

An enormous number of signature features have been proposed. In [CO 83] a sequence of 44 different features are used, derived from 3 signals generated by a force-sensitive pen. Some papers have also applied specific transformations to the signals and then used as features a truncated set of coefficients from these transformations. For example, [ZV 85] used the first 40 low-frequency coefficients of the power spectra obtained using a Walsh transform. At first it seems very attractive to reduce the task of signature verification to the two steps of low-level feature detection, followed by some standard method of feature-vector comparison. The price one pays for this simplification is that the overall result is only as good as the features selected. It is difficult to find a set of features that adequately describe all

the important aspects of the behavior of a signature. Given such a (large) set, there is the additional problem of assimilating these features into a classification scheme. We believe these feature extraction methods are limited in scope, because too much information is discarded during the initial processing stages.

Our work is more aligned with the first class of methods mentioned above, methods that deal with entire functions, and make use of all the available data. A large volume of data is recorded for each signature. One not only gets a reproducible rendering (shape) of each signature, but also the speed and pressure with which they were produced. It soon became apparent that with the abundance of data, we needed a model to bind all the information together. The model should conform with the physical process that takes place when someone signs their name, and the individual components and parameters should be accessible and useful.

Our model draws on two sets of methodology in the literature. The first is a general non-linear correlation method called Dynamic Time Warping (DTW) which is extensively used in speech recognition. The second is the area of geometric shape analysis.

Handwriting, much like speech, is time dependent. No two signatures will have exactly the same timing pattern, and these timing differences will not be linear. Time warping allows one to get a point-to-point correspondence between two signals which is relatively insensitive to small differences in their timing patterns. Our model uses DTW to match the speed signals of two signatures. We do not believe, however, that DTW should be used to compensate for other variations between signatures (such as Euclidean shape transformations). In [H 78] and [Y 77] DTW was used to match the (X_t, Y_t) co-ordinates of two signatures. In doing so they introduced nonlinear distortions in the time domain to compensate for rotation, scaling and shear which are linear transformations in the X, Y domain. The speed signal, on the other hand, is effectively invariant under these transformations. We deal with the shape analysis separately.

The second area which is of relevance to this work is concerned with the analysis of geometric shapes. There is a vast amount of literature in differential geometry and computer vision that deals with the problems of shape similarities under different transformations, such as projections, perspective and affine transformations. However, in the signature verification literature this problem has received hardly any attention ([S 82], [P 87]).

Our model represents each of several renditions of an individuals signature by a single template, together with an individual part that describes

deviations from the template. These individual parts are in turn modeled using some natural rules that describe how different versions of a person's signature might be expected to differ, i.e. in size, orientation and shear. The speed with which the signature is rendered is a separable part of the model; so even if a forger can mimic the shape of his victim's signature very accurately, it is unlikely that he will have mastered the relative speed with which the letters are traversed.

Here follows a broad overview of an algorithm to fit our model to a sample of size J of an individual's signatures (in practice $J = 5$ seems reasonable). We first smooth the (X, Y) pairs against *time* using a cubic smoothing spline; not so much to smooth out rough anomalies but rather to have a differentiable-function representation of the entire curve. Next we align and segment each signature into a series of corresponding sub-curves or *letters*. This is achieved in several steps. First the speed signals of the signatures are *time-warped* against a reference curve. The particular signature (and its associated speed signal) that is chosen as a reference is somewhat arbitrary; we choose the "best behaved" in a sense described below. This time warping establishes a correspondence between points on the reference signature and each of the other signatures. The speed signal of the reference signature is then used to segment it into distinct pieces or *letters*, and using the correspondences the other signatures are segmented accordingly. The template is computed at the letter level: for each set of J letters, an average shape is estimated up to an affine transformation, which allows for differences in location, scaling, orientation, and shear. These template letters are glued together to form the template signature.

When a new signature arrives for verification, it needs only to be compared to the template signature. The same set of techniques are used to validate a new signature as are used to fit the model to the training signatures:

- its speed signal is time-warped against the reference;
- it is segmented into letters;
- each letter is affinely transformed to match the template letter as well as possible.

Typically a forger can be identified at the time-warping stage, although the deviations that accumulate at each of the steps outlined above produce overwhelming evidence against authenticity. Figure 1 shows three signatures

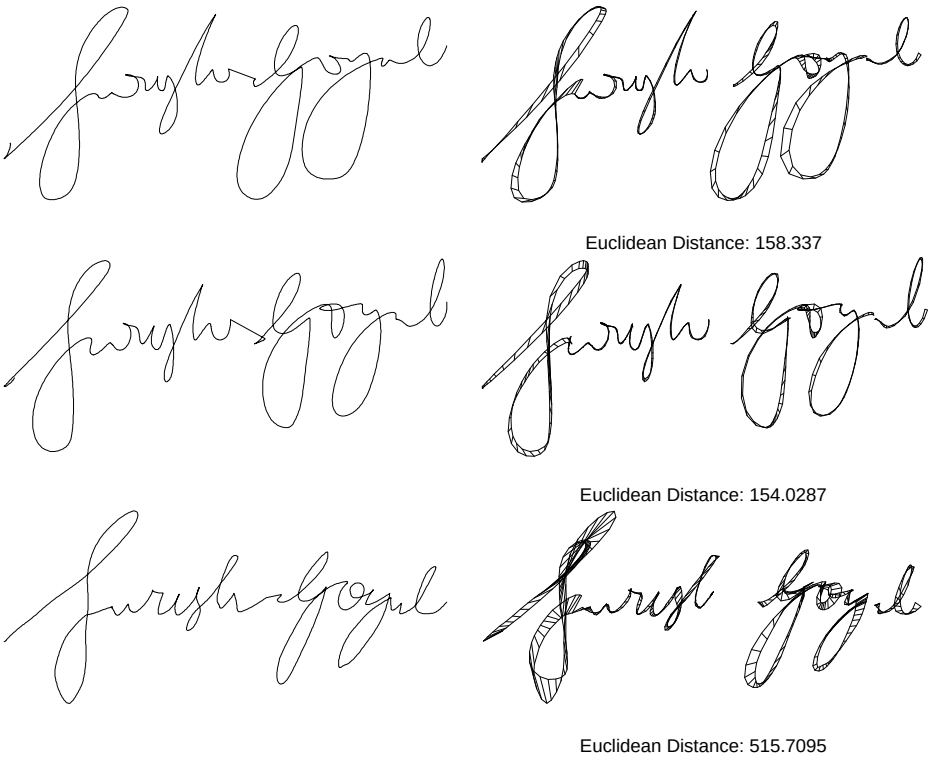

Figure 1: *The top two signatures are genuine, and the bottom one is a forgery of* Suresh Goyal's *signature. Superimposed on each is the the template signature* $\mathbf{F}$ *(dashed curves), after an optimal, piece-wise-constant, affine transformation in each case. This superimposition is much worse for the third signature.*

of the name *Suresh Goyal*, the last of which is a forgery. Superimposed on each is the template derived from *Suresh's* 5 training signatures. There is a clear mismatch in the case of the forgery.

This paper is organized as follows. Section 2 describes our stochastic signature model, and Section 3 gives a step by step description of our algorithm for fitting the model. We deliberately avoid excessive technical details in this section, with the hope that it reads easily. Section 4 gives some of the technical details and justifications omitted in section 3. The model and fitting process are illustrated using examples throughout the paper.

We do not provide extensive test results in this paper. Our experience so far has been limited to about 10 samples each from 20 colleagues, who also provided the forgeries for each others' signatures. Although these preliminary results appear promising, we are accumulating a much more extensive data base to exercise and calibrate our model.

2 Statistical Model

The two-dimensional image of a signature may be regarded as a piecewise continuous and locally smooth path, i.e. a vector function

$$\mathbf{Y}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}$$

parametrized by some scalar parameter u. Although the choice of the parameter is arbitrary, we typically identify it with time: the time taken as the signature is traced out. Since the timing is likely to be different for two different renderings of exactly the same signature, it is useful in some expressions to use the *same* parametrization throughout; for these purposes we regard u as a *universal parameter*.

Successive signatures by the same writer are not identical but will differ, both globally and locally, in location, scale and orientation. A plausible model for this process is

$$\mathbf{Y}(u) = \mathbf{A}(u)\mathbf{F}(u) + \boldsymbol{\mu}(u) + \mathbf{e}(u), \quad 0 \leq u \leq 1. \tag{1}$$

where $\mathbf{Y}$, $\mathbf{F}$, $\boldsymbol{\mu}$ and $\mathbf{e}$ are each 2-vectors, whose elements are functions of u. Such vector functions are known as parametrized curves. Similarly $\mathbf{A}$ is a 2×2 affine transformation matrix whose elements are functions of u. These components have the following interpretation:

- the fixed non-random function $\mathbf{F}(\cdot)$ may be regarded as the idealized template or "mean" signature for the writer in question.

- the functions $\mathbf{A}(\cdot)$ and $\boldsymbol{\mu}(\cdot)$ vary more slowly than $\mathbf{F}(\cdot)$ and are stochastic in nature.

- $\mathbf{e}(\cdot)$ is a stochastic vector of departures from the model; we expect $\|\mathbf{e}\|$ to be much smaller than $\|\mathbf{F}\|$, where $\|\cdot\|$ is the Euclidean norm.

Model (1) describes the *rendered* signature, as we might see it on a piece of paper, and does not have any information about *how* it was signed. Next we describe the dynamic aspect of our model; an aspect that traditionally has not been a worry to forgers. In practice, we do not record the signature $\mathbf{Y}(\cdot)$ as a function of u. Instead, our data consists of N points $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$ of the signature sampled at discrete time points $t_1, t_2, \ldots, t_N$ (usually equi-spaced), where

$$\begin{aligned} \mathbf{X}_i &= \mathbf{X}(t_i), \quad i = 1, 2, \ldots, N \\ &= \mathbf{Y}[h(t_i]) + \boldsymbol{\epsilon}_i \\ &= \mathbf{A}[h(t_i)]\mathbf{F}[h(t_i)] + \boldsymbol{\mu}[h(t_i)] + \mathbf{e}[h(t_i)] + \boldsymbol{\epsilon}_i \end{aligned} \tag{2}$$

Here, the monotone function $h(t)$ defines the transformation, known as a *time-warp*, needed to convert from the time parametrization of the recorded signature to units of the universal parameter u. This time-warp function is related to the speed at which the signature is written; as such it will change from one rendition to the next, and should also be regarded as stochastic. Finally, the $\boldsymbol{\epsilon}_i$'s denote the errors in measuring the signature at times $\{t_i\}$. In order to spare the reader from "his/her" contortions, we hereafter assume that our generic writer is male.

The non-random template function $\mathbf{F}(\cdot)$ characterizes the intrinsic *shape* of the signature, involving both the spelling of the writer's name and his style of writing (e.g. curly versus straight capital letters, curly or crossed *t*'s, etc.). The random functions $h(\cdot)$, $\mathbf{A}(\cdot)$ and $\boldsymbol{\mu}(\cdot)$ characterize *how* he writes his signature, and how it changes from one occasion to the next:

- relative to some fixed axis on an electronic pad, he will start at a different point each time, and at a particular angle. The model is invariant under such rigid body transformations, which are captured in the transformations $\mathbf{A}$ and $\boldsymbol{\mu}$.

- his writing speed will change from one signature to the next, as will its consistency over different regions of the signature; the function h reflects these speed variations
- he might sign slightly bigger or smaller on different occasions, and
- when he signs faster, the letters may slant more (in matrix language this is known as shear)

The last two items are also captured by the affine transformation functions $\mathbf{A}(\cdot)$ and $\boldsymbol{\mu}(\cdot)$, which allow these deformations to vary slowly along the signature.

Apart from providing the mechanism for deforming the template $\mathbf{F}$, the functions $h(\cdot)$ and $\mathbf{A}(\cdot)$ usefully focus the deviations between each signature and the template. For example, a sample of realizations of the time-warping functions h_j for J different signatures might exhibit more variation in some regions of the signature than in others. Similarly, the variations in the transformations $\mathbf{A}_j(\cdot)$ can be characterized, and broken down even further to pure scale changes, rotations and shear. These functions and knowledge of their their typical behavior can be used to detect forged signatures, since forgers are usually unable to reproduce these dynamic features of the genuine signature.

Our data-base for any given writer is based on J specimen signatures $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_J$ by that writer, where

$$\mathbf{Y}_j(u) = \mathbf{A}_j(u)\mathbf{F}(u) + \boldsymbol{\mu}_j(u) + \mathbf{e}_j(u) \tag{3}$$

$$0 \leq u \leq 1, \quad j = 1, 2, \ldots, J$$

There is one template $\mathbf{F}(\cdot)$ for all specimen signatures by that writer, whereas the $\mathbf{A}_j$'s, $\boldsymbol{\mu}_j$'s and $\mathbf{e}_j$'s are assumed to be independent realizations of the corresponding random functions.

Once again, we do not observe the functions $\mathbf{Y}_j(\cdot)$ directly. Instead, we observe the j-th signature $\mathbf{Y}_j(\cdot)$ at N_j discrete time points $\{t_{ji}\}$, i.e., we observe

$$\mathbf{X}_{ji} = \mathbf{Y}_j(h_j(t_{ji})) + \boldsymbol{\epsilon}_{ji} \tag{4}$$

$$i = 1, 2, \ldots, N_j, \quad j = 1, 2, \ldots, J$$

where for $j = 1, 2, \ldots, J$, the monotone functions $\{h_j(\cdot)\}$ convert the times t_{ji} to equivalent values of the universal parameter u.

The model proposed is very general—too general in fact to be useful without some additional restrictions. The location functions $\boldsymbol{\mu}_j$, if allowed to vary in an arbitrary fashion, would be sufficient to model any given signature, and therefore any deviation from the template $\mathbf{F}$. The restriction that occurs naturally in our method of estimation is to assume that both $\mathbf{A}_j(u)$ and $\boldsymbol{\mu}_j(u)$ are piecewise constant over segments of the signature. For convenience we will refer to these segments as *letters*, although they may encompass more or less than a single letter of the alphabet. In retrospect, this piecewise-constant restriction seems natural from a physical point of view as well. The physical action of sweeping out a single letter, such as an ℓ, is ballistic in nature and is likely to be smooth. The size, orientation and slant may differ slightly at each attempt, but the concept of a fixed template seems plausible. The change-points between letters, on the other hand, may exhibit some discontinuities.

3 Fitting the Model

In this section we give a more detailed and illustrated account of the steps in our algorithm for fitting (1) or (2). Most of the discussion concerns fitting the model to a set of J training signatures for a given person. To help tie the somewhat complicated sequence of operations together, we then give a summary of how each step is focused at a particular aspect of the model. Finally we describe how a new candidate signature is compared to the model signature. We deliberately leave out excessive technical details in this section to make it more readable; subsequent sections will hopefully fill in some of these technical gaps.

Step 1: Smoothing

Each signature is recorded as a sequence of (X, Y) coordinate pairs recorded at a uniform frequency of about 300 points per second. Along with the spatial coordinates, a third coordinate, pressure, is recorded at the same time points. First the pressure signal is used to segment the signature into separate words, based on the pen up-down sequences. In Figure 2 the pressure signal easily separates the two words *"Suresh"* and *"Goyal"*. The first two signatures are genuine, the third a forgery.

Although the recorded data is typically smooth, there are occasional gross outliers and noisy points which need to be removed (less than 1% of the total). These are easily identified by their large Euclidean distance from the

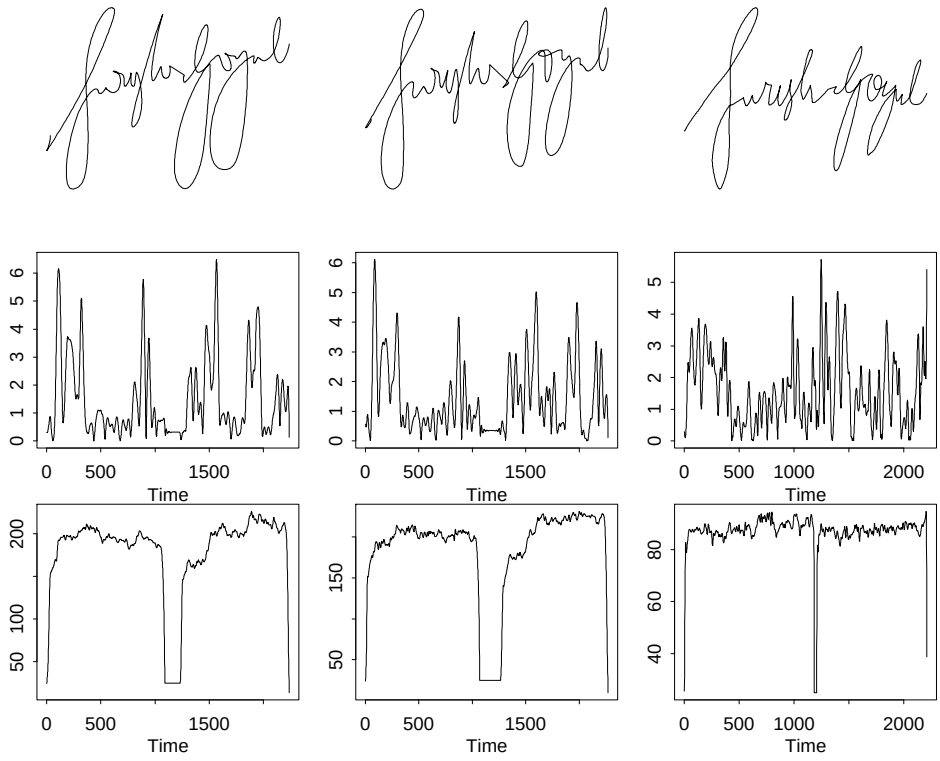

Figure 2: *The first two signatures are genuine reproductions of* Suresh's *signature, while the third is a forgery . The top row shows the signatures themselves, the second row shows the speed signal, and the third the pressure signal for each signature*

neighboring points. We then fit a smooth path through the remaining (X, Y) coordinates for each word in each signature. This is done by smoothing each coordinate separately against time using a cubic smoothing spline.

The amount of smoothing is chosen automatically by global cross-validation of the integrated Euclidean distance between the observed and fitted points. Typically very little smoothing is performed, which is what is intended, and the fitted curves usually come close to interpolating the observed sequences. There are three reasons for smoothing the signature sequences in this way:

- even though the amount of smoothing is small, it tends to eliminate small discontinuities introduced by measurement error due to the discretization during the recording process, or small movements during the signing.
- the cubic spline representation turns the sequence into a function that can be evaluated at any point t; this has consequences later on in the algorithm.
- the cubic spline has two continuous derivatives, the first of which is used in the speed computation.

Section 4.1 gives more details on the use of smoothing splines in this context.

Step 2: Computing the Speed Signal

We compute the speed signal $V_j(t)$ as a function of t for each smoothed signature using the derivatives available from the cubic spline representation. Our plan is to use the speed signal for time warping and further segmentation, but first a bit of background.

An essential aspect in fitting our model is in estimating the time-warp functions $h_j(t)$, which establish a correspondence between the parametrization of the recorded signatures $\mathbf{X}_j$ and the universal parameter u. They also establish a similar correspondence between two different recorded signatures. Estimating these h_j boils down to establishing correspondences between segments on two different curves. For example, suppose a particular piece of the S on the first signature consists of the point sequence recorded between times 25 and 39 milliseconds; the corresponding piece on the second signature, although slightly tilted and larger, might consist of the sequence recorded between 33 and 54 milliseconds. To establish such a correspondence, we need some function defined along each signature that

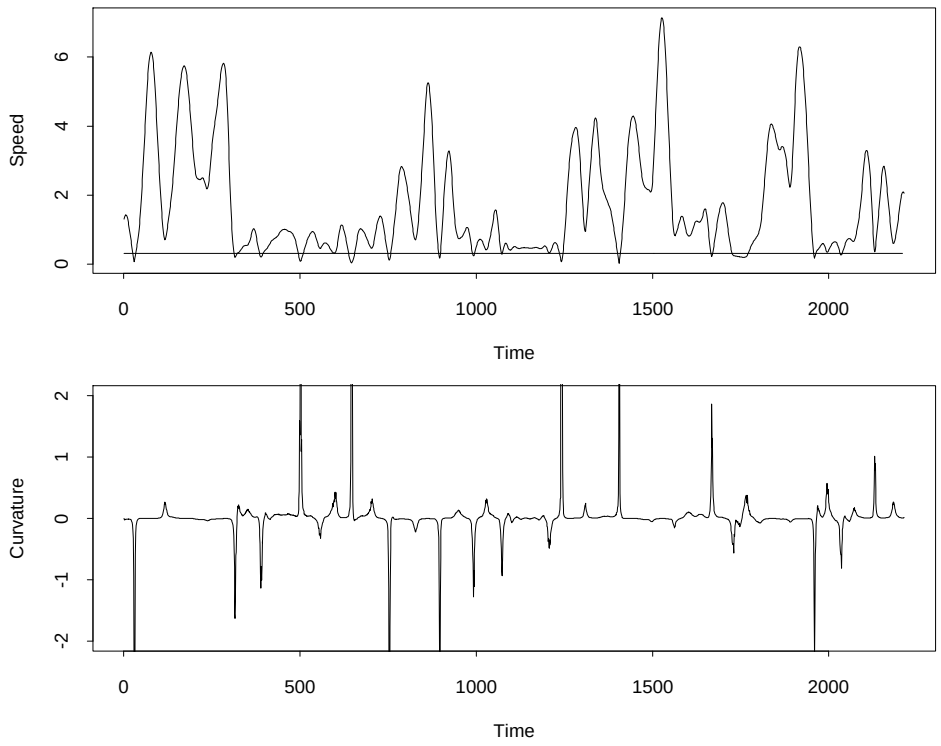

Figure 3: *The top panel shows the speed function computed from the cubic spline representation of one of* Suresh Goyal's *signatures, the bottom panel shows the curvature.*

reflects their intrinsic shape, irrespective of their orientation and location in the Euclidean plane defined by the recording pad.

The curvature function of a signature seems to fit the bill. Not only is it location and rotation invariant, but it does not depend on the particular parametrization used in defining the signature. Points of high curvature indicate big changes in shape, and should therefore be useful for segmenting a signature into logical pieces, irrespective of its orientation. By matching the curvature functions of two similarly shaped curves, each measured using different time parametrizations, one should be able to extract the time correspondences or time-warp functions. Figure 5 shows the results of such a matching.

In practice the curvature signal tends to be numerically unstable because

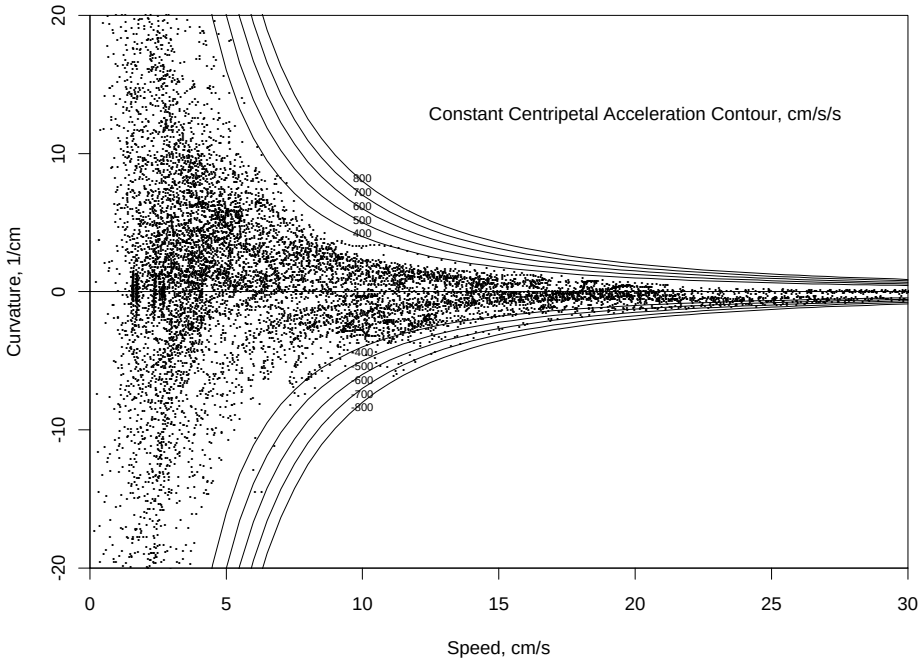

Figure 4: *The inverse relationship between speed and curvature based on the first five signatures of* Suresh Goyal *(taken from [N 91]).*

it is based on second derivative information, despite the fact that it is derived from the smoothed signature curves. This is specially true around points of high curvature, which are the areas of interest for segmentation. The speed function, on the other hand, involves first rather than second derivatives, and behaves in a more stable fashion. The speed function of a signature tends to have an inverse relationship with the curvature, such that points of high curvature correspond to points of low speed. Figure 4 shows an example of this inverse relationship for the speed and curvature signals of Figure 3. It seems then that we can use speed as a surrogate for the more invariant but noisy curvature. For each signature the speed is sampled from the spline representation of its coordinates uniformly in time. Section 4.2 gives a more rigorous justification for using the speed signal rather than curvature for time warping.

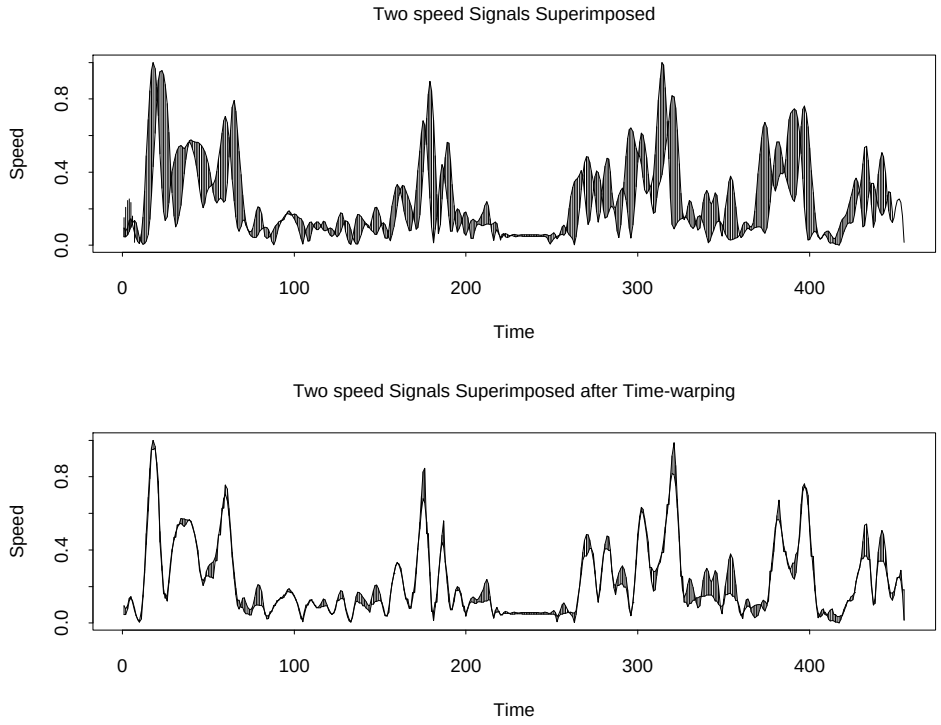

Figure 5: *The top panel shows the speed signals for the two genuine signatures in Figure 1 superimposed. The curves are connected pointwise by lines to emphasize their differences in height and phase. The bottom panel shows them time-warped against each other.*

Step 3: Time-Warping Speed Signals

The next step is to use the speed functions from any two signatures (or corresponding words from two signatures) to align them in such a way that corresponding points and regions on each can be identified. The technique we use is known as *Dynamic Time-Warping*, and is heavily used in speech recognition ([S 83]). It is a dynamic programming method used to obtain a nonlinear, *rubber-sheet* transformation between two p-dimensional signals ($p = 1$ in our case). The transformation consists of local stretching or compression of the time-axis of either signal relative to the time-axis of the other. The aim of these local adjustments is the minimization of some global measure used to quantify the difference between the two signals, such

as the least-squared error summed over the transformation path in the case of discrete time units. The transformation path represents the point correspondences established between the two signals.

Apart from providing the correspondences between two signatures, the time-warping algorithm produces an important distance measure in the speed domain between the two signatures.

All possible pairs of the J signatures for a given person are time-warped against each other, producing $J(J-1)/2$ sets of time correspondences, and their associated least-squares distances. From these the single most representative signature is selected: we choose the one whose mean time-warped speed distance from all the other $J-1$ signatures is smallest. We use a trimmed mean in this calculation, in the event that the individual perhaps had one bad signature in the batch. Note that this representative signature is not what we have been calling the template signature; it is simply a base signature for establishing correspondences. Suppose this representative signature is the first ($j=1$). Figure 6 shows a piece of a typical correspondence curve, and we see it fairly rough; in fact close inspection shows that it is piecewise linear. Each speed signal is represented at a set of time points, not necessarily the same number. Two or more adjacent points on one signature may be found to correspond to a single point on the other, which explains the sometimes flat regions in both the vertical and horizontal directions. Each of the $J-1$ correspondence curves is similarly plotted as a function of the time values for the representative signature, and each is smoothed to break the ties. For each time point of the representative speed curve V_1, the corresponding times for each of the other $J-1$ speed curves V_j can be read off from these smoothed correspondence curves. If there are N_1 speed values sampled from signature 1 at N_1 times, we get N_1 corresponding time points from each of the other $J-1$ signatures. These can then be used to retrieve the (X,Y) coordinates for the signatures themselves from their cubic-spline representations. If the time-warping did a good job, we might now expect to have corresponding sets of (X,Y) pairs for each of the J signatures for the individual.

Step 4: Segmentation

The representative curve is segmented at regions of low speed (high curvature). Figure 7 shows the points of segmentation based on the speed signal for the representative curve for the *Suresh* word. The lower speed threshold is chosen to be 15% of the mean speed, and as can be seen in the

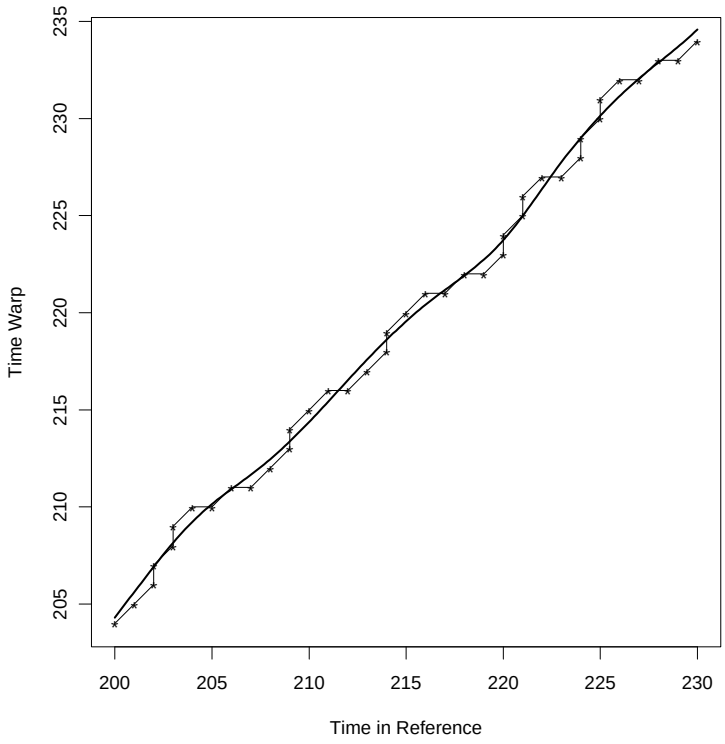

Figure 6: *The correspondences between two signatures is a piece-wise linear function, giving for each time point on one signature, the corresponding point(s) on the other. A smooth curve is fit to the correspondences to break the ties, and corresponding time points (the* time warp function*) can simply be read off this smooth curve. Here we show a small portion of the time-warp function between one of* Suresh Goyal's *signatures and his reference signature.*

figure, the portion of the signature associated with the local minima below this threshold are removed during the segmentation. This has little spatial consequence, since by definition points of low speed are close together in space. We have implemented a dynamic modification to this 15% threshold, which checks if any new local minima are discovered between the 15% and $(15+1)$% threshold; if so the threshold is incremented to 16% and the step is repeated. In practice this results in thresholds between 15–18%, and the segmentation process is more stable. The lower panel in figure 7 shows the segments themselves, separated by some white space. This is known as "self-segmentation" in the speech recognition literature ([R 82]). Once the segmentation has been performed for the representative signature, it is automatically defined for all the other signatures because of the correspondences. We refer to each of the segments as *letters*; in many cases they are single letters, but this is not a requirement.

Step 5: Affine Invariant Averaging

In this step we estimate the template signature at the letter level, and the optimal affine transformation between each letter and its template.

From the previous step we have a parallel sequence of letters for each of the J signatures; Figure 8 (right three panels, thicker curves) shows three S's from *Suresh Goyal's* signature, which were obtained from the automatic segmentation in the way described.

We denote these J renditions of the ℓth letter by $\mathbf{L}_{j\ell}$, each an $m_\ell \times 2$ matrix of (X, Y) coordinates ordered by row, and with a correspondence across rows. We define the average letter $\bar{\mathbf{L}}_\ell$ to be the minimizer of

$$\sum_{j=1}^{J} \left\| (\mathbf{L}_{j\ell} - \mathbf{O}_{j\ell})\mathbf{B}_{j\ell} - \bar{\mathbf{L}}_\ell \right\|^2 \tag{5}$$

over $\bar{\mathbf{L}}_\ell$, $\mathbf{B}_{j\ell}$ and $\mathbf{O}_{j\ell}$, where $\bar{\mathbf{L}}_\ell$ is an $m_\ell \times 2$ matrix, $\mathbf{O}_j = \mathbf{1}\boldsymbol{\mu}_j^T$ a location shift and the $\mathbf{B}_{j\ell}$ are 2×2 nonsingular transformations. The solution has a concise mathematical description in terms of the smallest eignevectors of an average projection operator (see section 4.4). There are two ways to compare the individual letters to the average. We can either superimpose $\mathbf{L}_{j\ell}$ and $\mathbf{O}_{j\ell} + \bar{\mathbf{L}}_\ell \mathbf{B}_{j\ell}^{-1}$ for each j separately, or else superimpose all the $(\mathbf{L}_{j\ell} - \mathbf{O}_{j\ell})\mathbf{B}_{j\ell}$ on $\bar{\mathbf{L}}_\ell$. Figure 8 compares the S's with the average S in both ways, and we see the correspondence is close.

At this point the average letters $\bar{\mathbf{L}}_\ell$ can be joined up sequentially and saved as the template signature $\mathbf{F}$ for *Suresh*.

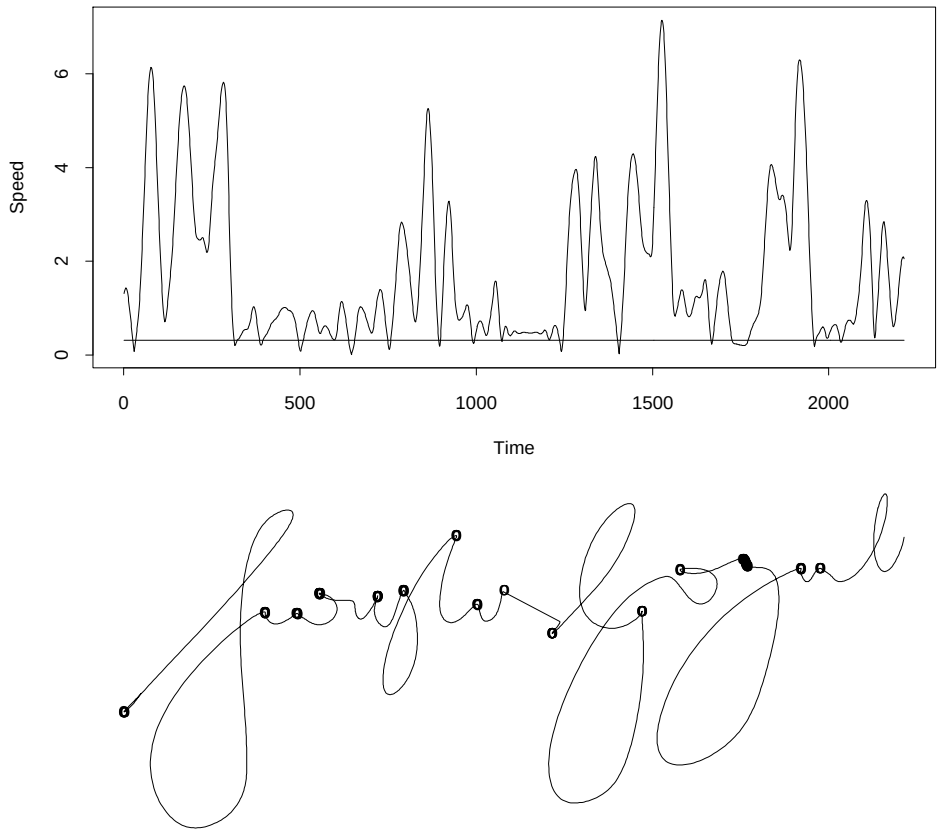

Figure 7: *The top panel shows the speed signal derived from the representative* Suresh Goyal *signature, and indicates the segmentation threshold. The lower panel shows the segments themselves.*

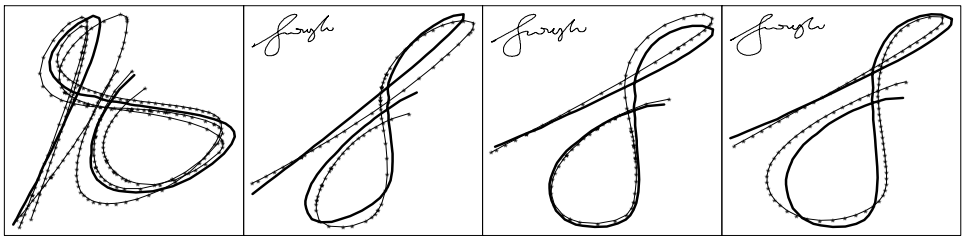

Figure 8: *The left most panel shows the affinely averaged template "S" letter (thick curve) from* Suresh Goyal's *signature, with the projections of three of his signatures superimposed. The remaining panels show the individual signatures, with the projected template superimposed.*

Relating the Five Steps to the Model

We now summarize the five steps described above according to their role in estimating the components of the model 2.

Step 1: Smoothing Although this smoothing step is primarily aimed at getting a function representation of the signature curves, it also serves to average out the measurement errors $\boldsymbol{\epsilon}_i$ in the recorded signatures.

Step 2: Speed The speed function for each signature is an intermediate step towards computing the transformations $h_j(t)$; they can also be viewed as an additional component of the model, as is the pressure signal.

Step 3: Time Warping This step produces estimates of the functions $h_j(t)$ which for each signature convert time to a value for the universal parameter u that indexes the template signature $\mathbf{F}$. The universal parameter is identified with the time parameter of the representative signature in step 3. The correspondences derived there provided a mapping from time points in that signature to time points on each of the other signatures; these mappings are then estimates of h_j^{-1}.

Step 4: Segmentation The reference signature is segmented at points of low speed into a sequence of *letters*. Using the correspondences established in step 3, all the signatures are similarly segmented.

Step 5: This step computes the template $\mathbf{F}$ and the piecewise constant functions $\mathbf{A}_j(u)$ and $\boldsymbol{\mu}_j(u)$ for each signature. For values of u corresponding to the ℓth letter, $\mathbf{A}_j^T(u) = \mathbf{B}_{j\ell}^{-1}$ and $\boldsymbol{\mu}_j(u) = \boldsymbol{\mu}_{j\ell}$. The

residuals from the affine averaging in step 5 are the remainder terms $\boldsymbol{e}_j$.

Verification of a New Signature

When a new signature presents for verification, steps 1–4 are repeated: It is smoothed and its speed signal is computed. This speed signal is then time warped against the speed signal for the template signature, and the correspondence is established. Typically a forgery can be identified at this stage because of the large speed discrepancy reported by the time warping procedure. If not, the new signature is segmented, the letters are affinely transformed to match the template, and least square distances are computed on a letter basis between the two. These distance statistics can be compared to the J similar distances for the original signatures from the template. It is extremely unlikely that a forger can mimic both the shape and relative speed with which a person signs their name. Figure 1 on page 5 shows the results of such a verification using a forgery.

4 Technical Details

In this section we elaborate on some of the technical details glossed over in the previous section.

4.1 Smoothing

We smooth the signature by smoothing each coordinate separately using a cubic smoothing spline smoother [W 90]. If the observed signature sequence is denoted by $\mathbf{X}_i$, $i = 1, \ldots, N$, measured at time points t_i, then the smoothed signature $\mathbf{S}(t)$ minimizes the criterion

$$\sum_{i=1}^{N} \|\mathbf{X}_i - \mathbf{S}(t_i)\|^2 + \lambda \int \|\mathbf{S}''(t)\|^2 \, dt \tag{6}$$

over a suitable Sobolev space of functions, and for some value of the smoothing parameter λ. This criterion and its solution is interesting for several reasons:

- the first part of the criterion encourages fidelity to the data, while the second part encourages smoothness by penalizing regions of high

curvature; the penalty λ measures these two conflicting goals against each other.

- The criterion is rotation invariant, and hence its solution is rotation equivariant. So if we rotate the points $\mathbf{X}_i$ prior to smoothing, we get exactly the rotated smooth of the unrotated points. Similarly, the solution is equivariant under uniform scale changes.

- The criterion separates into the sum of two criteria, one for each coordinate. These each have as a solution the univariate smooth of their coordinate values against time.

- The solution coordinate functions are cubic splines with knots at each of the time points t_i; as such they can be evaluated at ANY time points, not just the original t_i. Since they are piecewise cubic polynomials, their first and second derivatives are available everywhere as well.

The solution varies dramatically with the value of the smoothing parameter, which has to be supplied. We use the cross-validated, integrated, Euclidean squared distance

$$CV(\lambda) = \sum_{i=1}^{N} \left\| \mathbf{X}_i - \mathbf{S}^{\lambda}_{(i)}(t_i) \right\|^2 \tag{7}$$

as a criterion for selecting λ. Here $\mathbf{S}^{\lambda}_{(i)}(t_i)$ is the value of the smooth curve evaluated at t_i; the subscript (i) indicates that the ith point itself was omitted in the fitting of the curve. This criterion serves us well, since it recognizes the signal in the signature, and selects a value for λ such that only enough smoothing is performed to eliminate the small amount of measurement error.

4.2 Using Speed for Time-Warping

Here we give additional justification for using the speed signals extracted from two signatures as a basis for establishing a correspondence, or time-warping.

The speed of a differentiable curve parametrized by u is the amount of arc-length covered per small unit of u, and is computed as the norm of the coordinate derivative function:

$$V(t) = \left\| \dot{\mathbf{F}}(t) \right\|$$

The curvature function measures the normal component of acceleration relative to speed, and involves the second derivatives of the coordinate functions ([T 79], page 62):

$$\kappa(t) = \frac{\langle \ddot{\mathbf{F}}(t), \mathbf{N}(t) \rangle}{\left\| \dot{\mathbf{F}}(t) \right\|^2}. \tag{8}$$

Here $\mathbf{N}(t)$ is a normal vector to the curve at t. For a plane curve, the curvature at a point is also given by the inverse radius of the circle, tangent to the curve at that point, that matches the curve to second order (or more simply, that fits snugly into the curve at that point). As can be seen from their formulae the speed and curvature of a parametrized curve are invariant under rigid body transformations (location shifts and rotations). What is not so apparent from (8) is that the curvature function is independent of the parametrization of the curve $\mathbf{F}$; this is not true for the speed function in general. This invariance is clearly a desirable feature for a candidate function for time warping. For example, one can reparametrize a smooth curve to have *constant* speed; although the shapes of the curves would be identical, their speed functions would be useless for the purpose of time-warping.

Next we introduce scale in addition to the rigid-body transformations. Suppose then that signature 2 is simply a scaled, rotated and translated version of signature 1, at least over a small segment of each.

$$\begin{array}{rcl} \mathbf{Y}_1(u) & = & \mathbf{F}(u) \\ \mathbf{Y}_2(u) & = & \rho \mathbf{R} \mathbf{F}(u) + \boldsymbol{\mu} \end{array} \tag{9}$$

where $\mathbf{R}$ is orthogonal. Note that ρ, $\boldsymbol{\mu}$ and $\mathbf{R}$ do not depend on u.

It is easy to check that both their speed and curvature functions are proportional

$$\begin{array}{rcl} V_{Y_1}(u) & = & \rho V_{Y_2}(u) \\ \kappa_{Y_1}(u) & = & \kappa_{Y_2}(u)/\rho \end{array} \tag{10}$$

Now, if instead of observing the $\mathbf{Y}_j$ as a function of u, we observe $\mathbf{X}_j(t) = \mathbf{Y}_j[h_j(t)]$, $j = 1, 2$, then from the above

$$\kappa_{X_1}(t_1) = \kappa_{X_2}(t_2)/\rho \tag{11}$$

if $h_1(t_1) = h_2(t_2)$. In other words the curvatures at the two corresponding points on the signatures remain the same (up to the factor ρ), despite the fact that they have been parametrized differently. It also seems intuitive that, if we know ρ and if κ is non-constant, we should be able to solve

$t_1 = h_1^{-1}(h_2(t_2))$ for t_2 given any value of t_1. This operation of solving such equations for all values of t_1 by matching values of a common function is time-warping.

As mentioned the speed function does not in general have this parametrization invariance,

$$\begin{aligned} V_{X_1}(t) &= \left\|\dot{\mathbf{F}}(h_1(t))\right\| \left|\frac{\partial h_1(t)}{\partial t}\right| \\ V_{X_2}(t) &= \rho\left\|\dot{\mathbf{F}}(h_2(t))\right\| \left|\frac{\partial h_2(t)}{\partial t}\right| \end{aligned} \tag{12}$$

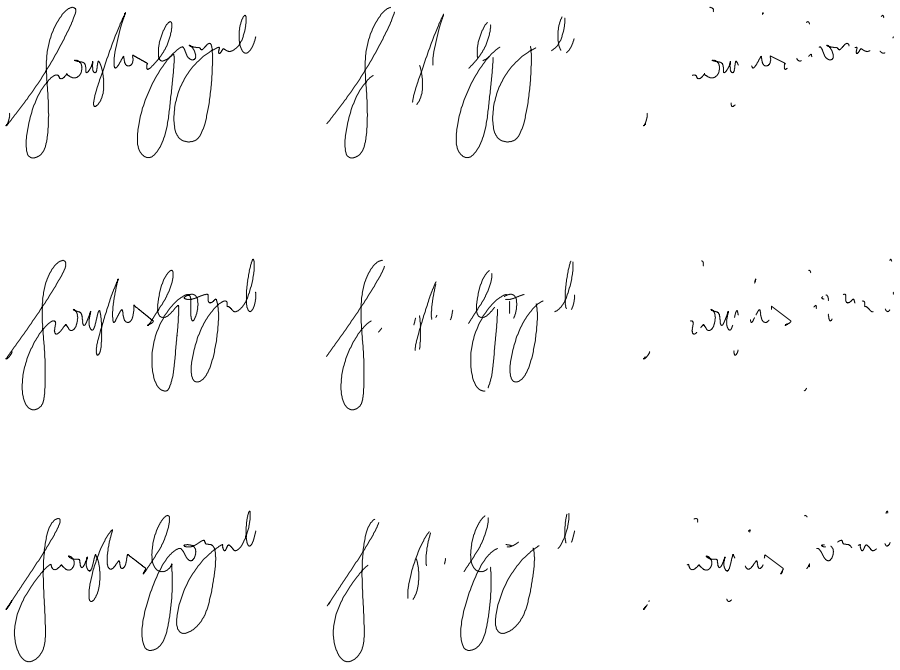

Figure 9: *For three of* Suresh Goyal's *signatures, we show the effect of speed segmentation. The left column of plots has the signatures themselves, the middle column the parts of the signature for which $V_{X_j} > 6cm/s$, and the right column those pieces for which $V_{X_j} \leq 6cm/s$. The figure gives additional evidence that the high speed portions are smooth, while the low speed portions correspond to high curvature pieces of the signatures.*

However, empirical evidence, as exhibited in figure 4, suggests that for

signatures, parametrized by writing time, speed and curvature are inversely related (curvature has a sign, which accounts for the symmetric pair of relationships with speed in figure 4). This makes intuitive sense: we need to slow down to turn sharp corners. Earlier we suggested that the action of signing is ballistic in nature, and we can push this argument a bit further here. When we swing a weight tied to the end of a string in a circle, the normal acceleration is constant and is proportional to the constant force we need to exert in maintaining the motion. By analogy, if we exert a nearly constant force (normal to the curve) when we sweep out the arc of a letter, the normal acceleration will be constant. But from (8) we see that since the numerator is precisely this normal acceleration, this implies $\kappa_X(t) \propto 1/V_X^2(t)$. This argument is presented in more detail in [N 91].

Of course real signatures do not behave exactly in the manner idealized here. Our model (2) allows other deformations besides rotation and global scaling, and allows them to vary with u as well. Our model also allows for errors $\mathbf{u}_j(u)$ which we ignored here, and the constant acceleration concept will at most be locally true. Fortunately, the time-warping algorithm operates in a local fashion as well.

4.3 Time Warping

Suppose we have two signatures (curves)

$$\begin{aligned} \mathbf{X}_1(t) &= \mathbf{Y}_1(h_1(t)) \\ \mathbf{X}_2(t) &= \mathbf{Y}_2(h_2(t)) \end{aligned}$$

where both $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are parametrized by u. We wish to identify "matching points" on both signatures, i.e., pairs of points with the same u-value. We need to do this on a reasonably-fine grid of u-values, not just for a few isolated "anchor-points" or "landmarks". This amounts to estimating the time-warping functions $h_1(\cdot)$ and $h_2(\cdot)$. This will only be possible if the two curves $\mathbf{Y}_1$ and $\mathbf{Y}_2$, although different, share some intrinsic characteristic that varies in parallel along their respective paths, that can serve as a basis for matching. In the previous section we argued that speed and curvature are such characteristics, given the sorts of differences we anticipate between the $\mathbf{Y}_j$. However, an additional requirement is that this characteristic should be invariant to the parametrization, so it can be computed using the $\mathbf{X}_j$ rather than the $\mathbf{Y}_j$ representation. Although speed fails in general in this regard while curvature does not, we argued that for signature sequences

$\mathbf{X}_j$ parametrized by time, curvature is approximately inversely proportional to the square of speed. This means that speed is also eligible as a basis for matching, and since it is more reliably computed, it is the measure we chose.

Symmetric Dynamic Time Warping

Dynamic time-warping (DTW) is a dynamic programming method used to obtain a nonlinear, *rubber-sheet* transformation between two p-dimensional, time-varying signals ([S 83]). The transformation consists of local stretching or compression of the time-axis of either signal relative to the time-axis of the other. The aim of these local adjustments is the minimization of some global measure used to quantify the difference between the two signals, such as the least-squared error summed over the transformation path in the case of discrete time units. The transformation path, which is simply a sequence of ordered pairs of corresponding time instances in the two functions, may easily be determined in a dynamic fashion so that the global measure is minimized.

In speech recognition asymmetric DTW has been the method of choice; the word *asymmetric* in this context means that one signal is being warped to fit a predetermined reference signal. However, we have chosen to use symmetric DTW for our work. Symmetric DTW allows for simultaneous stretching and compression of the two signals being compared; the mapping obtained may be applied to either signal and result in a good match to the other. We prefer the symmetric approach because this method does not assume in any given comparison that either signal is the template. A better correspondence may also be achieved due to the fact that both signals are being simultaneously adjusted to fit each other in the symmetric case.

Dynamic time warping has already been used in a limited capacity in signature verification systems([H 78] and [Y 77]), to match the $X(t)$ and $Y(t)$ signals. The results of these algorithms were less than satisfactory (see [PS 89]), due to the fact that the $X(t)$ and $Y(t)$ representation of the signature is not invariant under rotation and scaling. Hence, the time warping algorithm would try to compensate for deviations in rotation and scaling by successive modifications in the time axis. We on the other hand use the speed function of the signature; it can be computed in a stable fashion, is invariant under rotation, and equivariant to global scaling. Since the time-warping algorithm allows for slowly varying scale differences in the signals, the scale equivariance is not an issue.

Implementation

The two speed signals are evaluated on a grid of time points (we use a grid with resolution one fifth that of the original data). This results in two sequences V_{1i} and V_{2i}, of lengths n_1 and n_2. Typically the numbers of sampled points n_j are within 15% of each other. The goal of the algorithm is to find a sequence-correspondence mapping between the two signals. We denote this correspondence-mapping by a sequence of ordered pairs (j, k).

The symmetric DTW algorithm which we use is a basic one. Given a correspondence (i, j) between V_{1i} and V_{2j}, allowable future correspondences include those between $V_{1,i+1}$ and $V_{2,j}$, $V_{1,i+1}$ and $V_{2,j+1}$, and $V_{1,i}$ and $V_{2,j+1}$. As a global constraint we enforce the correspondence between sequence values $(1, 1)$ and (n_1, n_2). The desired overall correspondence is determined using dynamic programming to minimize the total weighted least-squared error between the speed signal over the correspondence path, allowing for local scale adjustment. Each pair (i, j) therefore has a scaled squared distance associated with it, and these are simply added over the path. These squared distances in our case are simply $(V_{1i} - V_{2j})^2$ adjusted in two ways:

- The distances associated with correspondences of the form $(i + 1, j)$ or $(i, j + 1)$ are weighted by a factor 0.5, while those of the form $(i + 1, j + 1)$ are not, during the computation of the weighted least-squared error. This weighting guarantees that correspondence paths which traverse routes such as $(i, j) \rightarrow (i + 1, j) \rightarrow (i + 1, j + 1)$ or $(i, j) \rightarrow (i, j + 1) \rightarrow (i + 1, j + 1)$ will not be penalized relative to the direct path $(i, j) \rightarrow (i + 1, j + 1)$ (see [K 83]).

- The speed signals are initially scaled to lie in $(0, 1)$. Furthermore, a local rescaling takes place before distances are computed: in effect one entire speed curve is scaled so that their local means (computed over a window around the target points) coincide, before the distance between the speeds in question is computed.

This latter adjustment reflects our interest in the rises and falls of the speed function as a basis for matching, rather than the actual levels attained.

We have altered the basic algorithm by adding additional local constraints, which prohibit more than one horizontal or vertical move in a row. The purpose of these local constraints is to prevent the mapping of a single time instant in one signal to many time instants in the other signal. In particular, having these restrictions allows for much improved segmentation and

shape analysis due to the fact that the mapping function does not distort the original functions to the same degree as in the case of warping without local constraints.

4.4 Affine Invariant Averaging

Here we give more details on the computations in step 6 in computing the average letter. Recall that the $\mathbf{L}_j$ are $m \times 2$ matrices representing different versions of the same letter, and with a correspondence between each of their rows. For notational simplicity we drop the subscript ℓ. We define the average (preshape) letter $\bar{\mathbf{L}}$ to be the minimizer of

$$\sum_{j=1}^{J} \left\| \mathbf{L}_j \mathbf{B}_j - \bar{\mathbf{L}} \right\|^2 \tag{13}$$

over $\bar{\mathbf{L}}_{m\times 2}$ and $\mathbf{B}_j$. We have omitted the terms $\mathbf{O}_j$, since they are trivially estimated as $\mathbf{1}\boldsymbol{\mu}_j^T$ using the centroid of $\mathbf{L}_j$ for $\boldsymbol{\mu}_j$. The standard practice in shape analysis ([G 75]) is to allow at most similarity transformations $\beta\boldsymbol{\Gamma}_j$ rather than general non-singular transformations $\mathbf{B}_j$, where $\boldsymbol{\Gamma}_j$ are orthogonal rotation matrices. Forcing the $\mathbf{B}_j$ to be similarity transformations seems too restrictive, since an individual may not only sign bigger or smaller, but may also squash or stretch his signature. Allowing him to alter the shear as well seems to call for unrestricted $\mathbf{B}_j$. Although it would seem more natural to have applied the transformation $\mathbf{A}_j$ to $\bar{\mathbf{L}}$ rather than $\mathbf{B}_j$ to $\mathbf{L}_j$, in keeping with our model, the form we have chosen admits a far cleaner solution. To avoid degeneracies we impose the constraint $\bar{\mathbf{L}}^T\bar{\mathbf{L}} = \mathbf{I}$, although other equivalent constraints are possible.

If $\mathbf{L}_j = \mathbf{Q}_j\mathbf{R}_j$ is the Q-R decomposition of the jth letter, then it is easy to show that the optimal $\mathbf{B}_j$ is given by $\mathbf{B}_j = \mathbf{R}_j^{-1}\mathbf{Q}_j^T\bar{\mathbf{L}}$ and hence $\mathbf{L}_j\mathbf{B}_j = \mathbf{Q}_j\mathbf{Q}_j^T\bar{\mathbf{L}}$. So at the minimum (13) is

$$\begin{aligned} \sum_{j=1}^{J} \left\| (\mathbf{Q}_j\mathbf{Q}_j^T - \mathbf{I})\bar{\mathbf{L}} \right\|^2 &= \sum_{j=1}^{J} \mathrm{tr}(\bar{\mathbf{L}}^T\mathbf{M}_j\bar{\mathbf{L}}) \\ &= J\mathrm{tr}(\bar{\mathbf{L}}^T\bar{\mathbf{M}}\bar{\mathbf{L}}) \end{aligned}$$

where $\bar{\mathbf{M}}$ is the average of the residual projection operators $\mathbf{M}_j = (\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^T)$. Since each of the $\mathbf{M}_j$ are symmetric and nonnegative, so is $\bar{\mathbf{M}}$.

Now minimizing $\mathrm{tr}(\bar{\mathbf{L}}^T \bar{\mathbf{M}} \bar{\mathbf{L}})$ subject to $\bar{\mathbf{L}}^T \bar{\mathbf{L}} = \mathbf{I}$ is a well known eigenvector problem, with solution $\bar{\mathbf{L}}$ being a basis for the eigenspace corresponding to the two smallest eigenvalues of $\bar{\mathbf{M}}$. Since each of the projection operators $\mathbf{Q}_j \mathbf{Q}_j^T$ has eigenvalues of zero or one, their average $\bar{\mathbf{P}}$ has eigenvalues in $[0, 1]$, and thus the solution also corresponds to the two *largest* eigenvalues of $\bar{\mathbf{P}}$.

Figure 10 shows the template estimated in this fashion from five of *Suresh Goyal's* signatures.

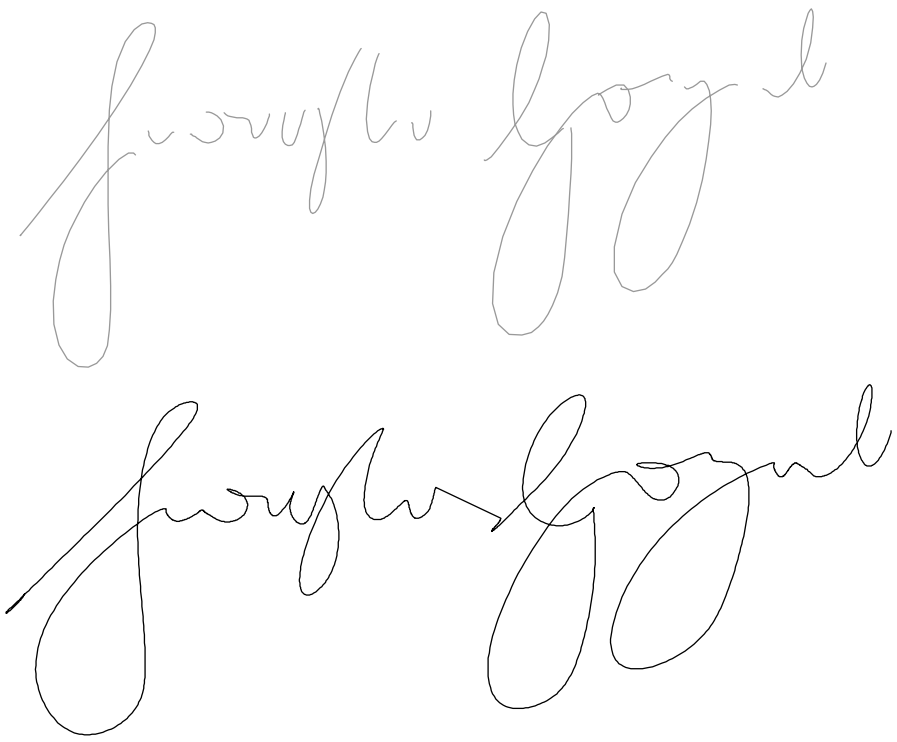

Figure 10: *The top panel shows the estimated template signature for* Suresh Goyal*, based on five of his signatures. The template letters are simply placed alongside each other, since the affine-invariant averaging leaves them location invariant. The lower panel shows, for comparison, one of the signatures from the training set.*

Some additional features of this approach are worth noting:

- Our problem has a closed form solution; had we used the similarity

transformations, an iterative solution would have been required (although admittedly eigendecompositions are iterative).

- The solution $\bar{\mathbf{L}}$ is a matrix, and represents the average letter at time points implicit for each row of the individual letters $\mathbf{L}_j$. In practice we need to represent the letter as a function. A simple solution that we have adopted is to smooth the average using the cubic smoothing spline smoother.

- Although each of the individual signatures is smooth, it may happen that the affine invariant average is not as smooth. One can impose a smoothness penalty on the criterion (13), by adding a quadratic roughness penalty of the form $\lambda \mathrm{tr} \bar{\mathbf{L}}^T \boldsymbol{\Omega} \bar{\mathbf{L}}$. A natural candidate for $\boldsymbol{\Omega}$ is the integrated, second-squared-derivative matrix corresponding to a smoothing spline. In this case the solution is a basis for the eigenspace corresponding to the two smallest eigenvalues of $J\bar{\mathbf{M}} + \lambda \boldsymbol{\Omega}$. In practice our unconstrained solutions are smooth enough not to require this additional constraint. Of course the simpler but less elegant solution is to once again smooth the columns of $\bar{\mathbf{L}}$.

- Although our average requires an eigendecomposition of an $m \times m$ matrix, the problem lends itself naturally to a simplified iterative solution. Iterative algorithms for finding eigen-subspaces of an operator (matrix) typically operate by repeatedly applying the matrix to the current guess, followed by a renormalization ([GvL 83]). Although $\bar{\mathbf{P}}$ is $m \times m$, it is the sum of projection operators (onto a 2-dimensional space), and thus each can be applied in $O(m)$ operations. The same is true for the constrained problem outlined in the previous item, since $\boldsymbol{\Omega}$ is composed of banded matrices. Our implementation uses this trick and results in a dramatic speedup in computation time.

5 Discussion

We have tested our model on a small sample of signatures, taken from a group of ten colleagues at AT&T Bell laboratories. Although the results should be regarded as preliminary, they have been promising. Most forgers will be detected using the dynamic aspect of the model alone—using the distances reported by the time-warping routine. An experienced static-signature forger has little concern for the dynamic aspect of the signature,

and is likely to be foiled when his speed function is compared to that of the owner of the signature. This is especially true when he has to learn to reproduce exotic flourishes, often illegible, but consistent in the victims signature. The signature of *Suresh Goyal*, on the other hand, is very even and neatly written. A forger might do well to simply write this name in their own natural style, and thereby mimic the speed function inherent in cursive handwriting. The forger in the examples used in this paper did a reasonable job, and in fact his signature segmented in much the same places as did those of his victim. However, to mimic both the speed and shape of the signatures proved too much in this case.

We need still to experiment with our model in many ways. Some of the items on our agenda are:

- we need to gather statistics on a letter basis in forming our rules for discrimination:
  - Some *letters* have longer arc-lengths, and/or involve more recorded points, than others. The shorter and simpler the letter, the more easy it is to transform it using the affine-transformations. Such letters are likely to appear stable in the training set, and are less likely to be detectable in the forgery.
  - Some letters will vary little in the training set (even longer, less simple ones), and will be strong candidates for discrimination; alternatively, others will have high variance and are likely to be poor discriminators.
  - The 2×2 transformation matrices contain additional information about the variability in each signature. They can be further decomposed into rotation, scale and shear factors, and analyzed separately.
- We are currently exploring alternative time-warping schemes, especially candidates that can be computed more efficiently.
- We plan to exercise the model on a large database of signatures, in order to evaluate its performance with some confidence.

Bibliography

[C 83] H. D. Crabe and J. S. Ostrem, "Automatic signature verification

using a three-axis-force-sensitive pen," *IEEE Trans. Syst. Man. Cybernetics, 13, pp. 329-337,* 1983.

[GvL 83] G. Golub, C. van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[H 78] M. Hanan, N.M. Herbst and J.M. Kurtzberg, "Finite state machine decision procedure model for signature verification", *IBM tech. Disclos. Bull. 20 3355-3360*, 1978.

[N 91] W. Nelson and E. Kishon, "Use of Dynamic Features for Signature Recognition," AT&T Bell Labs Technical Memorandum, 1991.

[P 89] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification - The state of the art", *Pattern Recognition, Vol. 22, pp. 107-131*, 1989.

[R 82] L. R. Rabiner A. E. Rosenberg J. G. Wilpon and T. M. Zampin, "A bootstrapping training technique for obtaining demisyllable reference patterns," *J. Acoust. Soc. Am. 71(6),* June 1982

[P 87] R. Plamondon, "What does differential geometry tell us about handwriting generation", *Proc. 3rd Int. Symp. on Handwriting and Comput. Applic., pp. 11-13, Montreal*, 1987.

[RS 90] J. Rice, and B. Silverman, "Estimating the Mean and Covariance Structure Nonparametrically when the Data are Curves", *JRSSB, to appear*

[S 82] Y. Sato and K. Kogure, "On-line signature verification based on shape, motion and handwriting pressure", *Proc. 6th Int. Conf. on Pattern Recognition, Vol. 2, pp 823-826, Munich*, 1982.

[S 83] D. Sankoff and J.B. Kruskal editors, "Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison", *Addison-Wesley Publishing Company Inc., pp. 125-160*, 1983.

[W 90] G. Wahba, *Spline Functions for Observational Data*, CBMS-NSF Regional Conference Series, SIAM, Philadelphia.

[Y 77] M. Yasuara and M. Oka, "Signature verification experiment based on non-linear time alignment," *IEEE Trans. Sys. man Cybernetics, 17, pp. 212-216*, 1977.

[ZV 85] K. P. Zimmermann and C. L. Varady, "Handwriter verification from one-bit quantized, pressure patterns," *Pattern Recognition, 18, pp. 63-72,* 1985.