PRE-PROCESSING THE DYNAMICS OF ON-LINE HANDWRITING DATA, FEATURE EXTRACTION AND RECOGNITION

Homayoon S.M. Beigi
IBM T.J. Watson Research Center P.O. Box 218 - Yorktown Heights, New York 10598 USA
EMail: beigi@watson.ibm.com

Most of the dynamic information present in an on-line handwriting signal is often ignored by on-line handwriting recognition systems. It is shown here that the dynamic information is complementary to the shape information and may be used to improve the accuracy of the recognition system.

1 Introduction

This paper presents results of an effort to use the dynamic information present in an on-line handwriting signal. On-line handwriting recognition systems often ignore most of the dynamic information available in the signal. They commonly go to the extent of retaining the order of points being sampled and throwing away all other dynamic (speed) information through a resampling of some sort. The system of [1], developed by our group at IBM Research. In the algorithm used by the system in [1], the writing is re-sampled to produce equidistant points; then a segment of the writing with a fixed number of points (hence the same Euclidean Length) is used to produce the feature vector. This feature extraction technique is used for both training and decoding. With a system of this type, the characters should be formed at a nominal size to get an acceptable comparison. For this reason, a size normalization is very important before training or decoding. [2]

To improve the recognition results of our current system [1], an effort was initiated to look at features which would be generated based on the dynamics of the handwriting data under the assumption that parts of the information obtained through these features would be complementary to those generated by existing features (referred to here as *static features* for the purpose of distinction). It is hoped that the union of these features in the form of complementary codebooks will increase the overall accuracy of the handwriting recognition system. The system of interest recognizes unconstrained handwriting (any combination of cursive or discrete writing styles). Size normalization is not as important with these dynamic features; although, a velocity (time) normalization would be required.

2 Handwriting Model

The general equation of motion of most rigid-body systems, with the consideration of dynamics such as inertia, gravity, Coriolis, Centrifugal, and other forces may be written as follows:

$$T = M(\alpha)\ddot{\alpha} + C(\alpha, \dot{\alpha}) + F(\dot{\alpha}) + G(\alpha) + T_d \tag{1}$$

with the following vectors defined: generalized forces supplied by the actuators (muscles), T, generalized coordinates, α, equivalent mass matrix, $M(\alpha)$, generalized forces due to Coriolis and Centrifugal accelerations, $C(\alpha, \dot{\alpha})$, generalized viscous friction forces, $F(\dot{\alpha})$, generalized gravitational forces and other potential energy such as stiffness, $G(\alpha)$, and disturbances, friction, and other unmodeled forces, T_d. In equation 1, all vectors are of dimension $nx1$ and $M(\alpha)$ has dimension nxn.

Equation 1 may be linearized about a set of reference coordinates at each time instant, hence approximating the non-linear equations of motion by a set of second order linear differential equations. Considering the governing equations of motion for the human-hand motor control, used for handwriting, we may make further simplifications by ignoring all forces but the D'Alembert forces and spring stiffness. These approximations only hold true under the conditions that the speed of writing is within some nominal range and that the hand moves in a two dimensional trajectory related to producing legible handwriting. Hollerbach, in his 1980 Doctoral Thesis [3], made even greater simplifications to this model by assuming a decoupled mass matrix. However, the mass and spring stiffness would still be functions of time even if we ignore all other forces. Another assumption which also holds pretty well is that the values for mass and spring stiffness are piecewise constant along a pen trajectory. These pieces along which the system parameters remain nearly constant are bordered at points of the extrema in the x and y velocities. These points coincide with zero velocity points for the corresponding coordinate, extrema of the dynamics (figure 1). Under certain assumptions such as minimum energy, minimum jerk, etc., these boundaries and the models may change. [3, 4]

Considering these approximations, let us then assume that the differential equations for the handwriting generation process may be approximated by a two dimensional second order equation with linear time-invariant coefficients along a piece of writing between any two consecutive velocity extrema in each coordinate (x and y), given by figure 1. Under these assumptions, the solution of the approximate differential equation, in the approximation region, would be of the usual sine and cosine form. In fact the velocity in each coordinate will also have the same form. For the sake of modeling handwriting it is better to consider the velocity rather than the position for the apparent reasons of robustness to noise and pre-emphasis. Velocities in the x and y directions under these very crude assumptions are given by equation 2.

$$\dot{x} = A_x cos(\omega_x t + \phi_x) + \bar{v}_x \quad and \quad \dot{y} = A_y cos(\omega_y t + \phi_y) + \bar{v}_y \tag{2}$$

where A_x, ω_x, ϕ_x and $\bar{v}_x$ are the amplitude, frequency, phase and mean velocity for the x direction. Also, A_y, ω_y, ϕ_y and $\bar{v}_y$ are the counterparts in the y direction. $\dot{y} - \bar{v}_y$ for a word after segmentation may be well approximated by cosines with constant parameters. This approximation, although not as good, is also acceptable for the x direction. Initially, the phase may not seem to play any important role, however, if the word is to be segmented in such a way as to use **either** $\dot{x} = 0$ **or** $\dot{y} = 0$, then the phase plays the very important role of synchronization. In fact in some cases a segment boundary is simply deleted just because too small a window is generated; in these cases, also, the phase is essential (figure 1).

3 Pre-Processing

In a practical sense, if segments of the writing are written very slowly, some problems may arise. The first problem is the computation of segmentation points where $\dot{x}$ and $\dot{y}$ are nearly zero. The finite difference approximation of velocities in x and y assumes a Δt which is small in comparison to the nominal Δx or Δy. However, in slow speeds, since the Δt is a fixed number equal to the inverse of the sampling frequency, the velocity is not correctly estimated. To alleviate this problem, a time normalization is done to generate writing with similar nominal speeds. Figures 4 through 6 show plots of x and y versus time for three different samples of the character a being written by the same writer. Shape of the character is the same for all three samples. In these figures, darker lines show the original points and lighter lines show points after time normalization. The a in figure 5 is an example written in an empirically determined nominal speed. The sample of figure 4 was written with a uniform speed, but in general much more slowly than the nominal speed. After time normalization, the plots are very similar to those shown in figure 5. The character of figure 4 is shown in figure 2. In addition to causing problems in velocity computation, slowly written characters also contain a fair amount of noise which was reduced here by using a zero phase low-pass filter. Filtration results are shown in figure 2 using a thinner line. Finally, figure 6 shows the most complicated case, namely, when the velocity of writing changes within a single stroke. This often happens when a person is preoccupied by a thought while writing. Using this time normalization the mean velocities within segments may be approximated by the mean velocity over the whole stroke, thus producing compression.

4 Parameter Estimation

Consider the spring model. Equations 2 give the solution to the x and y velocities of the signal. The $\bar{v}_x$ term of these equations is the mean x velocity which results in the separation of the characters. If this velocity were equal to zero, the hand would stay stagnant and all the characters would be formed overlapping one another. For estimating $\bar{v}_x$, we assume that the mean velocity is constant within each stroke. Therefore, the value of $\bar{v}_x$ is estimated to be the mean value of v_x computed within a stroke. Similarly, we notice that if the writer is to write on a horizontal line (rule), the mean y velocity should be zero. $\bar{v}_y$ may also be estimated to be the mean value of the y velocity. Using these estimates of $\bar{v}_x$ and $\bar{v}_y$ and subtracting these values from $\dot{x}$ and $\dot{y}$ in equations 2, the new x and y velocities will be given by:

$$\dot{\hat{x}} = A_x sin(\omega_x(t - t_0) + \phi_x) \;\; and \;\; \dot{\hat{y}} = A_y sin(\omega_y(t - t_0) + \phi_y) \tag{3}$$

Now the problem reduces to estimating the parameters of two sine curves given a few data points. An optimization problem is formulated for estimating these parameters. A very good set of initial conditions are picked for the amplitude, the frequency and the phase. Due to the reliability of these initial conditions, a penalty is imposed on deviating from the initial amplitude and frequency values. From this point on, due to the similarity of the equations for $\dot{\hat{x}}$ and $\dot{\hat{y}}$, the procedures for estimating the generic amplitude A_η, the generic frequency ω_η and the generic phase ϕ_η are presented. Here, η is considered to be a generalized coordinate which may be replaced by x or y. Consider an n point segment of a sampled stroke using the segmentation scheme of last sections. If the sample interval is denoted by Δt, the following n-dimensional vectors may be defined where $1 \leq k \leq n$:

$$\vec{t} : t_k = (k-1)\Delta t, \quad \vec{s} : s_k = sin(\omega_\eta t_k + \phi_\eta), \quad \vec{\xi} : \xi_k = t_k sin(\omega_\eta t_k + \phi_\eta)$$
$$\vec{\Xi} : \Xi_k = {t_k}^2 sin(\omega_\eta t_k + \phi_\eta), \; \vec{c} : c_k = cos(\omega_\eta t_k + \phi_\eta) \; and \; \vec{e} : e_k = \dot{\vec{\eta}} - A_\eta \vec{s} \tag{4}$$

The objective is to find a segment of a sine wave which would model the data points in this segment with minimal sum of squares of errors. The parameters to be estimated for each coordinate are the amplitude, A_η, the frequency, ω_η, and the phase, ϕ_η. Let us denote the local velocity vector (with the mean velocity subtracted) of the points in the segment as $\dot{\vec{\eta}}_d$. Then, the simplest objective function for the optimization problem of estimating the system parameters is the sum of squares of errors along the segment, given by, $E_\eta = (\dot{\vec{\eta}}_d - A_\eta \vec{s})^T (\dot{\vec{\eta}}_d - A_\eta \vec{s})$.

The problem with minimizing E_η is that by increasing the frequency, it is possible to reduce the value of E_η, however, a high frequency sinusoidal curve will not fit the general continuous signal well. It is imperative to introduce a penalty on the size of the frequency. This is especially true since a very good initial value for the frequency may be obtained by estimating the zero crossings of the data. A good estimate of the amplitude, A_η may also be obtained by using the point with the maximum absolute velocity. Since the data is segmented at the zero velocity points, an initial value of 0 is very appropriate for ϕ_η. One may be satisfied with these initial values and not want to compute the actual values. However, here is where a major contribution of this paper lies. In picking the segments, any combination of $\dot{\hat{x}} = 0$ and $\dot{\hat{y}} = 0$ may be used. Assume that a segment is between the points $\dot{\hat{x}} = 0$ and $\dot{\hat{y}} = 0$ and we are estimating the parameters for the y velocity. In this case, the y velocity does not start from 0. Therefore, the above initial estimates may be off. Note that as stated earlier, the frequency tends to converge to large numbers to reduce the error E_η, and that we have a good initial estimate of ω_η based on zero-crossings. We also have a good initial estimate of the amplitude A_η. Based on these arguments, we should not allow the frequency and amplitude to deviate from their initial conditions by a lot. Therefore,

$$E_\eta = (\dot{\vec{\eta}}_d - A_\eta \vec{s})^T (\dot{\vec{\eta}}_d - A_\eta \vec{s}) + \alpha(\omega_\eta - \tilde{\omega}_\eta)^2 + \beta(A_\eta - \tilde{A}_\eta)^2 \tag{5}$$

where $\tilde{\omega}_\eta$ and $\tilde{A}_\eta$ are the initial estimates of ω_η and A_η respectively and α and β are weighting factors. Considering the objective function of equation 5, a Newton's method may be used to iteratively solve for the set of parameters which minimize E_η. To apply Newton's method to solving this optimization problem, let us define the state vector for the parameters to be estimated, as, $\zeta_\eta = \begin{bmatrix} A_\eta & \phi_\eta & \omega_\eta \end{bmatrix}^T$ Then the gradient of E_η, g, is written as,

$$g = \nabla_\zeta E_\eta = \begin{bmatrix} -2\vec{s}^T(\dot{\vec{\eta}} - A_\eta \vec{s}) + 2\beta(A_\eta - \tilde{A}_\eta) \\ -2A_\eta \vec{c}^T(\dot{\vec{\eta}} - A_\eta \vec{s}) \\ -2A_\eta \vec{\xi}^T E_\eta + 2\alpha(\omega_\eta - \tilde{\omega}) \end{bmatrix} \tag{6}$$

Similarly, the symmetric $3x3$ Hessian matrix, G, may be written as follows,

$$\begin{bmatrix} 2(\vec{s}^T \vec{s} + 2n\beta & 2\vec{c}^T(A_\eta \vec{s} - \vec{e}_\eta) & 2\vec{c}^T(A_\eta \vec{\xi} - \vec{e}_\eta) \\ G_{12} & 2(A_\eta \vec{c}^T(A_\eta \vec{c} + \vec{e}_\eta) & A_\eta \left[A_\eta n(n+1)\Delta t + 2\vec{\xi}^T(\dot{\vec{\eta}} - 2A_\eta \vec{s})\right] \\ G_{13} & G_{23} & 2A_\eta \left[A_\eta \vec{t}^T \vec{t} - 2A_\eta \vec{\xi}^T \vec{\xi} + \vec{\Xi}^T \dot{\vec{\eta}}\right] \end{bmatrix}$$

Given the above equations and solving for G^{-1} analytically, it is simple to solve for the parameter estimate vector, $\vec{\zeta}_\eta$ using the Newton step,
$\vec{\zeta}_\eta^{(i+1)} = \vec{\zeta}_\eta^{(i)} - \gamma^{(i)} H^{(i)} g^{(i)} \quad where \quad i = 0, 1, 2, \cdots.$

Here, H is the inverse Hessian and γ is a weight which may be computed using any line search technique such as Golden Section, Fletcher, etc. [5]

5 Reconstruction and Recognition

The parameters of equation 2 may be estimated and used for several purposes including reconstruction and recognition of the writing. The fast match shape recognition techniques of [1], based on a degenerate single-state continuous density Hidden-Markov Model, were used for obtaining preliminary shape recognition results, using these estimated parameters. In these tests, the digits from 0 to 9, upper and lower case letters were trained and tested. The system was trained on a total of 2303 characters and tested on a total of 833 characters. Characters were not at all uniform and are not presented here due to space restrictions. The same data files were also used for training and testing of the fast-match version of the system with static features [1]. Figures 3, 7 and 8 show the results of these runs. Based on these results, performances of the two systems are quite complementary, presenting ideal conditions for the combination of these two codebooks. For example, digit 1 and characters I, J, j and n get zero accuracy with the static features and an average of 76% accuracy with the dynamic features. Likewise, F, K and X get zero percent accuracy with the dynamic features and an average of 50% accuracy with the static features. The overall character accuracy of the system using dynamic features is 47% where that of the system using static features is 66%.

Part of the reason for a worse total character accuracy of the system using the dynamic features is due to the lack of context. The system using static features, uses the window edges (see figure 1) as its window centers and looks to the left and right of these centers for a fixed length. These windows are overlapping one another, hence capturing some local context. [1] In the dynamic feature case, only the information within a window (between two consecutive segmentation points) is used to generate features and therefore no local context is available. It is quite acceptable to assume that by using a multi-state Hidden Markov Model, for both cases, the two performances should converge each other. However, the important point to realize is the complementary nature of the two systems and the fact that they may be used together to produce better results. This experiment is being worked on at the present time and no results are available yet. The delay in obtaining such results is partly due to the different front ends (windowing paradigms). Some writer-dependent fast match results are also available for unconstrained handwriting using the dynamic features. These results were obtained using a lexicon of 368 words related to computer applications. Writers 1 and 2 had word accuracies of 91.5% (184 / 201) and 75.8% (229 / 302) respectively. The two writers' training files were combined into one and the system was trained tested on the writers to show word accuracies of 87.1% (175 / 201) and 70.9% (214 / 302) respectively.

6 Conclusions and Future Research

If the combination of training files is continued further, as discussed at the end of the previous section, a write-independent prototype set will be established. Preliminary attempts at doing this have shown that the features should be tranformed (engineered) to produce better generalization capability through some normalization. Normalization procedures presented in the Pre-Processing section of this paper have been considered to achieve such results. These normalization techniques have not been used in the results shown in the paper. This is an on-going effort and new results will be presented in the near future.

It is also notable that the correlation coefficient between the frequencies in the x and y direction is on the order of 0.7 which suggests that one of the two frequencies may be omitted for the sake of recognition without any large performance loss. In fact, the removal of the ω_x resulted in a slightly higher accuracy for both writers of the last section. This is due to the reduction of the dimension of the Gaussian mixtures used for representing the dynamic features [1] from 6 to 5 which makes the means and variances of these Gaussians more accurate given the fixed amount of training data. For this reason, experiments have been made to reduce the dynamic features further down to 4 without any loss of accuracy. Due to the different window definition for computing the features in the new system and the system of [1], the two codebooks are not easily combined. An attempt is being made to generate outputs from both fast match techniques (static and dynamic) and then pass a combination of words proposed by these systems to the detailed match scheme of [1].

References

1. K. Nathan, Homayoon S.M. Beigi, G. J. Clary, J. Subrahmonia and H. Maruyama, "Real Time On-Line Unconstrained Handwriting Recognition using Statistical Methods," ICASSP, Detroit, MI, May, 1995.
2. Homayoon S.M. Beigi, K. Nathan, G. J. Clary, and J. Subrahmonia, "Size Normalization in On-Line Unconstrained Handwriting Recognition," ICIP Proc., Nov. 13-16, 1994.
3. John M. Hollerbach, "An Oscillation Theory of Handwriting," PhD Thesis, MIT, 1980.
4. A.M. Alimi and R. Plamondon, "Performance Analysis of Handwritten Strokes Generation Models," IWFHRIII, NY, May 1993, pp.272-283.
5. Homayoon S.M. Beigi, "Neural Network Learning and Learning Control Through Optimization Techniques," Doctoral Thesis, SEAS, Columbia University, NYC, NY, 1991.

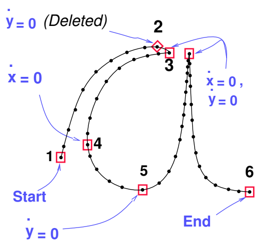

Figure 1: Segmentation of char. *a*

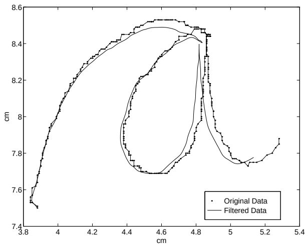

Figure 2: *a* written slowly

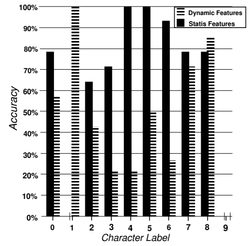

Figure 3: Accuracy Comparison for Digits

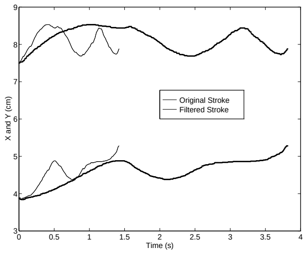

Figure 4: *a*: written slowly

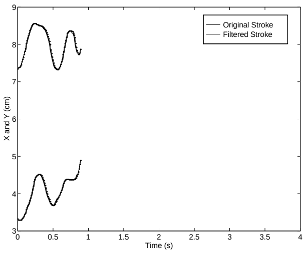

Figure 5: *a*: nominal speed

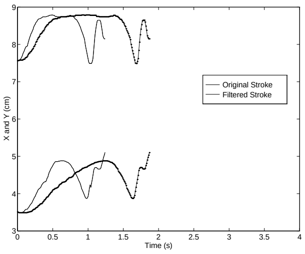

Figure 6: *a*: Variable Speed

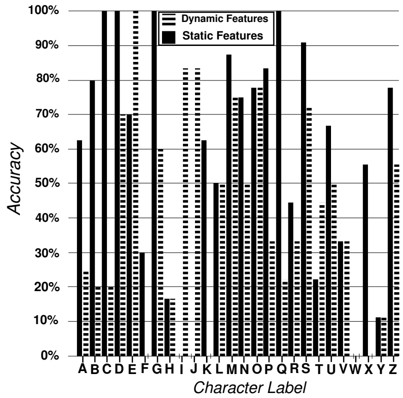

Figure 7: Upper Case Acc. Comparison

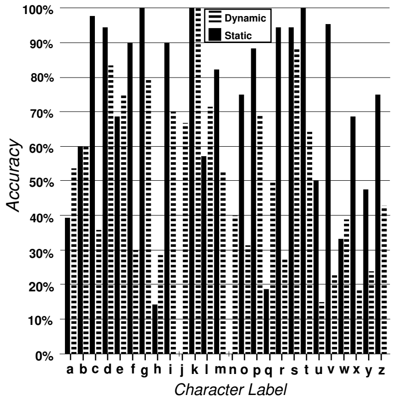

Figure 8: Lower Case Acc. Comparison