Segmentation and Reconstruction of On-line Handwritten Scripts*

Xiaolin Li†, Marc PARIZEAU‡ and Réjean Plamondon§

printed August 4, 1997

Abstract

On-line handwritten scripts consist of sequences of components that are pen tip traces from pen-down to pen-up positions. This paper presents a segmentation and reconstruction procedure which segments components of a script into sequences of static strokes, and then reconstructs the script from these sequences. The segmentation is based on the extrema of curvature and inflection points in individual components. The static strokes are derived from the delta log-normal model of handwriting generation and are used in component representation and reconstruction. The performance of the procedure is measured in terms of root mean square reconstruction error and data compression rate.

1 Introduction

On-line handwritten script segmentation is an active research topic since it promises a structural description for the underlying shape.(1–6,8,11,13,15) An on-line handwritten

*This work will appear in *Pattern Recognition*, probably late 1997 or early 1998. It was supported in part by NSERC grant OGP0155389 to M. Parizeau and in part by NSERC grant OGP00915 and FCAR grant ER-1220 to R. Plamondon.

†Xiaolin Li was with the Département de génie électrique et de génie informatique, Université Laval. He is now with the Département de génie électrique et de génie informatique, École Polytechnique de Montréal, Montréal (Québec), Canada H3C 3A7. E-mail: *xiaolin@scribens.polymtl.ca*

‡Marc Parizeau is with the Département de génie électrique et de génie informatique, Université Laval, St-Foy (Québec), Canada, G1K 7P4. E-mail: *parizeau@gel.ulaval.ca*

§Réjean Plamondon is with the Département de génie électrique et de génie informatique, École Polytechnique de Montréal, Montréal (Québec), Canada H3C 3A7. E-mail: *rejean@scribens.polymtl.ca*

script captured by the digitizing tablet consists of sequences of components that are pen tip traces from pen-down to pen-up positions. As a component can be segmented into a sequence of primitives at its characteristic points, a script can be described by sequences of attributed pattern primitives. This kind of representation can be used for data compression,[5] ink data search,[11] and on-line recognition.[1–3,8,10,13,16]

The definition and extraction of characteristic points within a component is very diverse in different systems. For example, some systems use the minima of curvilinear velocity to characterize the pen-tip trace,[1,3] some systems use the local extrema of X and Y coordinates as the characteristic points,[5,11,13] and some systems use the geometric parameters[6] or the extrema of curvature[8] to segment the component. In general, different definition and extraction leads to a different primitive representation.

With the electronic ink data increasing dramatically in some applications such as electronic ink search within a large database,[11] the issue of ink data compression and reconstruction has arisen recently. In this paper we present a pragmatic approach derived from the delta log-normal model of handwriting generation[4,12,14,17] for on-line handwritten script segmentation and reconstruction. The procedure detects the local extrema of curvature in individual components. It uses the pen tip traces on the $x-y$ plane for curvature analysis. Unlike the techniques for stroke corner detection based on eight-neighbor chain code,[8] this technique uses interpolated boundary points to compute the sequence of changes in angles. The extrema of curvature are detected based on this information. In our procedure, each component of a script is segmented into a sequence of static strokes at its landmark points (i.e. pen-down and pen-up points, local extrema of curvature, inflection points, and middle points of circular shape). Then the script is reconstructed based on this representation. The performance of the procedure is measured in terms of root mean square reconstruction error and data compression rate.

The remainder of this paper is organized as follows. Section 2 summarizes the delta log-normal model and some of its properties. Section 3 presents our algorithms for landmark point detection and script segmentation. Section 4 describes script reconstruction using sequences of static strokes. Section 5 presents our experimental results concerning not only script segmentation and reconstruction but also the root mean square reconstruction error and data compression rate. Finally, section 6 concludes this paper.

2 Delta Log-Normal Model

2.1 Simple stroke

Delta log-normal model is a powerful tool in analyzing rapid human movements.[7,15] It describes a neuromuscular synergy in terms of the agonist and antagonist systems involved in the production of these movements.[14] With respect to handwriting generation, the movement of a simple stroke is controlled by velocity.[4,15] $v_\xi(t)$, the magnitude of that velocity can be described analytically by:

$$v_\xi(t) = D_1\Lambda(t;t_0,\mu_1,\sigma_1^2) - D_2\Lambda(t;t_0,\mu_2,\sigma_2^2) \tag{1}$$

where

$$\Lambda(t;t_0,\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}(t-t_0)} \exp\left\{-\frac{[ln(t-t_0)-\mu]^2}{2\sigma^2}\right\} \quad t_0 \le t \tag{2}$$

is a log-normal function. In the above equations, t_0 represents the activation time, D_i are the amplitude of impulse commands, μ_i are the mean time delay, and σ_i are the response time of the agonist and antagonist systems, respectively, on a logarithmic scale axis.

The angular direction of the velocity can be expressed as

$$\theta(t) = \theta_0 + \int_{t_0}^{t} c_0 v_\xi(u)du \tag{3}$$

where θ_0 is the initial direction, c_0 is a constant representing the curvature of the stroke, and the derivative of $\theta(t)$ is referred to as the angular velocity:

$$v_\theta(t) = c_0 v_\xi(t) \tag{4}$$

Thus, the trace $\bar{v}$ of the stroke is characterized by a vector:

$$\bar{v}(t-t_0) =< v_\xi(t), \theta(t) >, \quad t_0 \le t \tag{5}$$

From the above definition, it readily follows that given $v_\xi(t)$, the curvature along a stroke is time invariant. Thus, the static shape of the stroke is an arc, and can be characterized by:

$$S = <\theta, c, D> \tag{6}$$

where $\theta = \theta_0$ is the initial angular direction, $c = c_0$ is the constant curvature, and $D = D_1 - D_2$ is the arc length. We call this a static stroke because there is no time variable in this expression.

2.2 Vectorial summation of simple strokes

Now consider a complex movement. Let $\bar{v}^{(1)}(t - t_0^{(1)})\bar{v}^{(2)}(t - t_0^{(2)}) \cdots \bar{v}^{(n)}(t - t_0^{(n)})$ be an n-stroke sequence, where the ith stroke is characterized by:

$$\bar{v}^{(i)}(t - t_0^{(i)}) = < v_\xi^{(i)}(t), \theta^{(i)}(t) >, \quad t_0^{(i)} \le t \tag{7}$$

where $t_0^{(i)}$ is the activation time, $v_\xi^{(i)}(t)$ is the magnitude, and $\theta^{(i)}(t)$ is the angular position. The movement of a component can be considered to be the vectorial summation of the n strokes in the time domain:

$$\bar{v}(t) = \sum_{i=1}^{n} \bar{v}^{(i)}(t - t_0^{(i)}) \tag{8}$$

In the above summation, each stroke in the sequence has an effect on all subsequent strokes. Since the n strokes are superimposed on one another, it is very difficult to recover them given only the summation observation $\bar{v}(t)$. However, if one assumes that the magnitude of each stroke decreases very fast, i.e.

$$v_\xi^{(i)}(t) \approx 0 \quad \text{when} \quad T^{(i)} < t \tag{9}$$

and that for $1 < i < n$,

$$t_0^{(i)} < T^{(i-1)} < t_0^{(i+1)} < T^{(i)} \tag{10}$$

then it follows that

1. when $T^{(i-1)} \leq t < t_0^{(i+1)}$

$$\begin{aligned} \bar{v}(t) &\approx \bar{v}^{(i)}(t - t_0^{(i)}) \\ c(t) &\approx c_0^{(i)} \end{aligned} \tag{11}$$

   $c(t)$ is approximately time invariant.

2. when $t_0^{(i)} \leq t < T^{(i-1)}$

$$\begin{aligned} \bar{v}(t) &= < v_\xi(t), \theta(t) > \\ &\approx \bar{v}^{(i-1)}(t - t_0^{(i-1)}) + \bar{v}^{(i)}(t - t_0^{(i)}) \end{aligned} \tag{12}$$

   where

$$\begin{aligned} v_\xi(t) &\approx v_\xi^{(i-1)(i)}(t) \\ &= \{[v_\xi^{(i-1)}(t)]^2 + [v_\xi^{(i)}(t)]^2 + 2v_\xi^{(i-1)}(t)v_\xi^{(i)}(t)\cos[\theta^{(i-1)}(t) - \theta^{(i)}(t)]\}^{\frac{1}{2}} \end{aligned} \tag{13}$$

$$\begin{aligned} \theta(t) &\approx \theta^{(i-1)(i)}(t) \\ &= \tan^{-1}\left[\frac{v_\xi^{(i-1)}(t)\sin\theta^{(i-1)}(t) + v_\xi^{(i)}(t)\sin\theta^{(i)}(t)}{v_\xi^{(i-1)}(t)\cos\theta^{(i-1)}(t) + v_\xi^{(i)}(t)\cos\theta^{(i)}(t)}\right] \end{aligned} \tag{14}$$

   and the curvature satisfies

$$\begin{aligned} c(t) &= \frac{d\theta(t)}{d\xi(t)} \\ &\approx \frac{d\theta^{(i-1)(i)}(t)}{d\xi^{(i-1)(i)}(t)} \end{aligned} \tag{15}$$

   where

$$\xi^{(i-1)(i)}(t) = \xi^{(i-1)}(t_0^{(i)}) + \int_{t_0^{(i)}}^{t} v_\xi^{(i-1)(i)}(u)du \tag{16}$$

   Thus, $c(t)$ is time variant.

In the following sections, we will describe a pragmatic segmentation and reconstruction procedure based on the above description and assumption, using the static curvature information.

3 Script Segmentation

3.1 Landmark points

Landmark points refer to points of the following categories in a component: 1) pen-down and pen-up points, 2) local extrema of curvature, and 3) inflection points of curvature.

3.1.1 Extrema of curvature

In differential calculus, the curvature c at a point p on a continuous plane curve C is defined as

$$c = \lim_{\Delta s \to 0} \frac{\Delta \alpha}{\Delta s} \tag{17}$$

where s is the distance to the point p along the curve and $\Delta\alpha$ is the change in the angles of the tangents to the curve at distance s and $s + \Delta s$, respectively. Note that the direction of the tangent line corresponds to the direction of the velocity. Since the sign of curvature is related to the curve direction, one can define convex and concave curvature to reflect this relationship:

- c is *convex* if and only if $c < 0$;
- c is *concave* if and only if $c > 0$.

In the light of the above definitions, the direction of the tangent line always turns clockwise with *convex curvature*; while the direction of the tangent line always turns counterclockwise with *concave curvature*.

In practical cases, it is difficult to calculate the above limit when the analytical representation of the curve is not available. However, by using a small unity interval $\Delta s = 1$ along the curve, the curvature can be approximated as $c = \Delta\alpha$. The above idea can be realized by interpolating data points along a component[1] such that $C = p_1 p_2 \cdots p_L$,

[1]This is also referred to spatial sampling.

where $d(p_l, p_{l+1}) = 1$, $1 \leq l < L$. In our case, the curvature c_l at point p_l can be computed as

$$c_l = \Delta\alpha_l \tag{18}$$

using the unity interval $\Delta s = 1$.

To estimate the above curvature, we compute the sequence of angles from point to point:

$$A = \alpha_1 \alpha_2 \cdots \alpha_L \tag{19}$$

where $\alpha_l \in [-180°, 180°]$ is the angle of the tangent line at p_l, which is estimated as

$$\alpha_l = \tan^{-1}\left[\frac{y_{l+1} - y_l}{x_{l+1} - x_l}\right] \tag{20}$$

In the light of A, we can obtain the sequence of changes in angles:

$$\Delta A = \Delta\alpha_1 \Delta\alpha_2 \cdots \Delta\alpha_L \tag{21}$$

where

$$\Delta\alpha_l = (\alpha_l - \alpha_{l-1}) \bmod 360° \tag{22}$$

Equation (22) guarantees that $\Delta\alpha_l \in [-180°, 180°]$. In the extreme cases that $\alpha_l - \alpha_{l-1} = \pm 180°$, $\Delta\alpha_l$ has the same sign as that of $\Delta\alpha_{l-1}$, since the original signal is supposed to be continuous.

It is obvious that the above sequence contains the curvature signal mixed with digitization and quantization noise. The influence of the noise can be suppressed by convolving the sequence ΔA with an appropriate filter G in the spatial domain: $\Delta A^* = \Delta A * G$. The curvature is then estimated based on the filtered signal:

$$c_l = \Delta\alpha_l^* \tag{23}$$

After the above signal is obtained, we define a local extreme of curvature based on signal intensity using the following measures:

- signal intensity:

$$I = \sqrt{\frac{1}{L}\sum_{l=1}^{L} \Delta\alpha_l^* \times \Delta\alpha_l^*} \tag{24}$$

- threshold of signal intensity:

$$T = k_S I + k_L \tag{25}$$

where k_S is the slope coefficient and k_L gives the lowest threshold value.

Using the above measures, we are now in a position to define the local extrema of curvature along a discrete component:

- If $\Delta\alpha_l^*$ is a local minimum such that $\Delta\alpha_l^* \leq -T$, then p_l is a point of *local minimum of curvature.*
- If $\Delta\alpha_l^*$ is a local maximum such that $T \leq \Delta\alpha_l^*$, then p_l is a point of *local maximum of curvature.*

All the above points constitute the points of *local extrema of curvature* in a component.

3.1.2 Inflection points

An inflection point is defined as the zero crossing point of curvature between two consecutive extrema of curvature if the convex-concave property of these extrema opposite to each other, and none of them display curvature discontinuity.

3.1.3 Special consideration

In some cases a portion of a component between two consecutive landmark points may consist of some circular parts. For example, sometimes one may write a character 'o' like a perfect circle and sometimes one may write an 'S' so smooth that the curvature along its trace is almost a constant. In the above cases there are no extrema of curvature detectable. Under such circumstances we insert a middle point as a special landmark point. To avoid uncertainty in determining the center angle of an arc (see following subsection), the above insertion is performed recursively until all arc segments are less than a semi-circle.

3.2 Component segmentation

We segment a component into a sequence of attributed strokes (arcs) at its landmark points such that

$$C = S_1 S_2 \cdots S_{N-1} \tag{26}$$

where $S_i =< \theta_i, c_i, D_i >$, $i = 1, 2, \cdots, N-1$, and N is the number of landmark points in the component. Two stroke categories are defined here for computational reason:

1) straight line segments[2]

2) arcs

For each portion of the component corresponding to stroke S_i, we use three points $p_0(x_0, y_0)$, $p_m(x_m, y_m)$, and $p_1(x_1, y_1)$ (p_0 and p_1 are landmark points and p_m is the middle point of that portion) to classify the portion and compute the stroke attribute:

$$\begin{aligned} \eta_1 &= \tan^{-1}\left[\frac{y_m - y_0}{x_m - x_0}\right] \\ \eta_2 &= \tan^{-1}\left[\frac{y_1 - y_m}{x_1 - x_m}\right] \end{aligned} \tag{27}$$

- S_i is a line segment if $\eta_1 = \eta_2$.

  In this case, the stroke attribute is computed as:

$$\begin{aligned} \theta_i &= \eta_1 \\ c_i &= 0 \\ D_i &= [(x_1 - x_0)^2 + (y_1 - y_0)^2]^{\frac{1}{2}} \end{aligned} \tag{28}$$

[2]A straight line segment can be regarded as an arc where the curvature of the arc tends to zero.

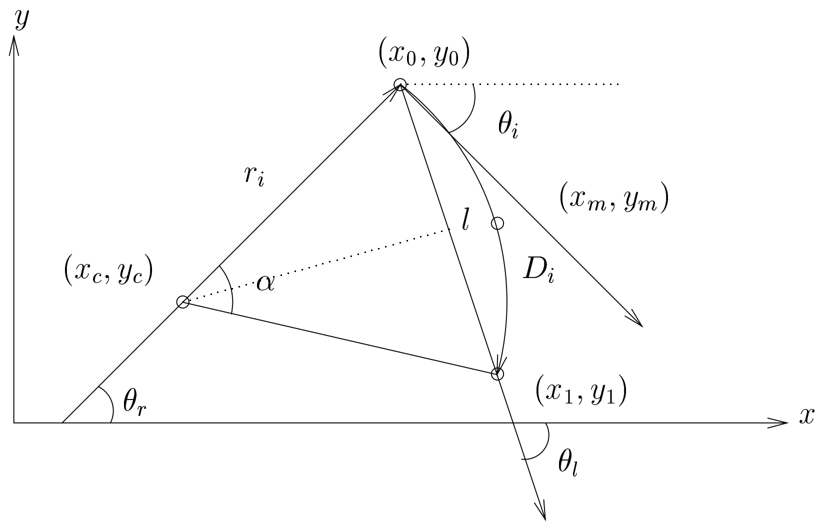


Figure 1: Computing arc attribute

- S_i is an arc if $\eta_1 \neq \eta_2$.

  In this case, the stroke attribute is computed by the following steps: (See Figure 1):

  1. determine radius r_i and center $o_i(x_c, y_c)$ by solving the following equations:

  $$\begin{array}{l} r_i = [(x_0 - x_c)^2 + (y_0 - y_c)^2]^{\frac{1}{2}} \\ 2(x_m - x_0)x_c + 2(y_m - y_0)y_c = x_m^2 - x_0^2 + y_m^2 - y_0^2 \\ 2(x_1 - x_m)x_c + 2(y_1 - y_m)y_c = x_1^2 - x_m^2 + y_1^2 - y_m^2 \end{array} \quad (29)$$

  2. determine relevant parameters l, α, θ_r, θ_l, and θ:

  $$\begin{array}{l} l = [(x_1 - x_0)^2 + (y_1 - y_0)^2]^{\frac{1}{2}} \\ \alpha = 2\sin^{-1}\left[\frac{l}{2r_i}\right] \end{array} \quad (30)$$

  $$\begin{array}{l} \theta_r = \tan^{-1}\left[\frac{y_0 - y_c}{x_0 - x_c}\right] \\ \theta_l = \tan^{-1}\left[\frac{y_1 - y_0}{x_1 - x_0}\right] \end{array} \quad (31)$$

  $$\theta = (\theta_l - \theta_r) \bmod 360^\circ \quad (32)$$

3. determine θ_i, c_i, and D_i:

$$\theta_i = \begin{cases} \left(\eta_1 + \frac{\alpha}{2}\right) \bmod 360^\circ & \theta < 0 \\ \left(\eta_1 - \frac{\alpha}{2}\right) \bmod 360^\circ & \theta \geq 0 \end{cases} \tag{33}$$

$$c_i = \begin{cases} -\frac{1}{r} & \theta < 0 \\ \frac{1}{r} & \theta \geq 0 \end{cases} \tag{34}$$

$$D_i = \alpha r_i \tag{35}$$

Now that each component of a script has been segmented into a sequence of static strokes, we can store the pen-down point and the stroke sequence of each component. The script can be reconstructed component by component based on these sequences.

4 Script Reconstruction

4.1 Component reconstruction

A component can be reconstructed stroke by stroke, starting from its pen-down position. Given a stroke $S_i =< \theta_i, c_i, D_i >$ and starting point $p_0(x_0, y_0)$, the trace of the stroke can be computed as follows.

- straight line segment reconstruction:

$$\begin{aligned} x &= x_0 + d\cos\theta_i \\ y &= y_0 + d\sin\theta_i \end{aligned} \tag{36}$$

  where $d \in [0, D_i]$.

- arc reconstruction (see Figure 1):

1. determine r_i, α, and $o_i(x_c, y_c)$:

$$r_i = \frac{1}{|c_i|} \tag{37}$$

$$\alpha = \frac{D_i}{r_i} \tag{38}$$

$$\begin{aligned} x_c &= x_0 - r_i \cos\theta_i \\ y_c &= y_0 - r_i \sin\theta_i \end{aligned} \tag{39}$$

2. reconstruction:

$$\begin{aligned} x &= x_c + r_i \cos(\theta \bmod 360°) \\ y &= y_c + r_i \sin(\theta \bmod 360°) \end{aligned} \tag{40}$$

where

$$\begin{aligned} \theta &\in [\theta_r - \alpha, \theta_r] \quad c_i < 0 \\ \theta &\in [\theta_r, \theta_r + \alpha] \quad c_i > 0 \end{aligned} \tag{41}$$

where

$$\theta_r = \tan^{-1}\left[\frac{y_0 - y_c}{x_0 - x_c}\right] \tag{42}$$

The end point of a current stroke can be easily computed. Then it will be passed to the next stroke as a starting point.

4.2 Root mean square reconstruction error

Let $P = p_1 p_2 \cdots p_K$ be a component segment associated with a static stroke S_i. We define a point sequence $P' = p'_1 p'_2 \cdots p'_K$ sampled from S_i that best correspond to P in the following manner:

1. case of straight line segment

   In this case, we locate $P' \in S_i$ such that $d(p_k, p'_k) = d(p_k, S_i)$, where $d(p_k, S_i)$ denotes Euclidean perpendicular distance between point p_k and line segment S_i.

2. case of arc

   In this case, we locate $P' \in S_i$ such that p'_k is the intersection of arc S_i and the line segment $\overline{p_k o_i}$, where $o_i(x_c, y_c)$ is the center of S_i.

Note that in either case, we always have $p_1 = p'_1$ and $p_K = p'_K$ as a result from our component segmentation.

Based on the above description, we can concatenate point sequences stroke by stroke and component by component. Then we get two point sequences that correspond to each other at script level.

Let $P = p_1 p_2 \cdots p_M$ be the point sequence of a script and $P' = p'_1 p'_2 \cdots p'_M$ be its correspondent in the reconstructed traces, the root mean square reconstruction error at script level is defined as:

$$rmse = \frac{1}{H}\sqrt{\frac{1}{M}\sum_{m=1}^{M} d^2(p_m, p'_m)} \tag{43}$$

where H is the normalized height of the script. Thus, the error can be expressed in percentage compared with H.

5 Experimental Results

So far we have described our procedure for on-line handwritten script segmentation and reconstruction. The results of our experiment using this technique are summarized in Tables 1 and 2 and are discussed in details in this section.

The experimental results shown in this section were obtained with four different data sets. The first data set contains 55 French words written by one person. The second data set contains 2600 characters (a-z) written by 10 different people (10 samples per character class per person). The third and fourth data sets named "unipen1a" and "unipen1c" are from "train_r01_v05", the fifth release of UNIPEN training data, and contain 6519 digits (0-9) and 28799 characters (a-z), respectively. Other details concerning number of

data set	scripts	components	points				strokes
			data	extreme	inflection	middle	
French words	55	170	10469	956	113	12	1251
Characters	2600	3050	89099	8788	781	185	12804
Unipen1a	6519	8181	361318	17862	2628	339	29010
Unipen1c	28799	35176	1144547	73590	7058	1796	117620

Table 1: Segmentation statistics

data set	*rmse*			bytes		compression rate
	max	min	avg	original	compressed	
French words	2.38%	0.70%	1.20%	41876	16372	60.90%
Characters	8.21%	0.09%	1.55%	356396	178048	50.04%
Unipen1a	11.65%	0.00%	1.32%	1445272	413568	71.38%
Unipen1c	9.22%	0.00%	1.45%	4578188	1692848	63.02%

Table 2: Reconstruction statistics

components and number of data points (coordinate data) of each data set are given in Table 1.

In our experiment, the first data set (55 French words) and the second data set (2600 a-z characters) were used for tuning the parameters of the segmentation algorithm, while the third data set (unipen1a) and the forth data set (unipen1c) were used for testing. The parameters of the segmentation algorithm include: 1) the normalized script height, 2) the passband of the filter, and 3) the coefficients k_S and k_L related to the threshold of signal intensity. They are discussed in the following subsection.

5.1 Script segmentation and reconstruction

Before applying this technique, each of these scripts was empirically scaled at 80 unit height ($H = 80$) with the width kept proportional to its original size. This kind of normalization is necessary because curvature is a scale dependent measure. The normalization value of 80 was found to be sufficient to maintain the correlation that is predicted by the handwriting generation model between the extrema of curvature, the maxima of angular velocity and the minima of curvilinear velocity.[15,17]

After the normalization, the data points of each component were smoothed, and then they were interpolated. For each component in all the above cases, ΔA, the sequence of changes in angles was iteratively convolved with a Gaussian function G twice[3] using the following equations:

$$\Delta\alpha_l^* = \frac{1}{W} \sum_{s=l-16}^{l+16} w_s \Delta\alpha_s \tag{44}$$

where

$$\begin{aligned} w_s &= e^{-[0.2(s-l)]^2} \\ W &= \sum_{s=l-16}^{l+16} w_s \end{aligned} \tag{45}$$

The bandwidth of the filter[4] is 0.1665 when $\Delta s = 1$, where Δs is the spatial sampling interval.

In our experiment, the extreme point detection was carried out with two steps: a main step and a supplemental step. At the main step, the slope coefficient and the bound related to thresholding were set as $k_S = 0.125$ and $k_L = 2$, respectively, and extrema corresponding to that threshold were detected. At the supplemental step, all segments that resulted from the main step were checked, and those long segments that stretched in the x direction (such as segments that looked like flat sine wave) were further processed. As the curvature signal was very weak along such segments, the threshold of extrema was tuned down by setting $k_S = 0.0625$ and $k_L = 1$.

The segmentation statistics concerning the four data sets are shown in Table 1. Figures 2 to 6 show some examples in this experiment.

[3] In the light of convolution theorem, performing $\Delta A(s) * G(s) * G(s)$ in the time-spatial domain is equivalent to performing $\Delta A(\omega) G^2(\omega)$ in the frequency domain, where $\Delta A(\omega)$ and $G(\omega)$ are the Fourier transforms of $\Delta A(s)$ and $G(s)$, respectively. $G^2(\omega)$ has a lower passband but a smaller truncation effect than that of $G(\omega)$ on $\Delta A(\omega)$ when $\Delta A(s)$ and $G(s)$ are digitized.

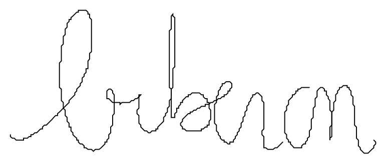
[4] This is referred to the angular frequency ω_c that satisfies $G^2(\omega_c) = 1/\sqrt{2}$.

The script in Figure 2 (a) consists of 2 components. Its landmark points are shown in Figure 2 (b) with 1) the points of local extrema of curvature marked by black circles, 2) the inflection points marked by empty circles. For simplicity the pen-down and pen-up points are not marked by any signs because they are easily located. The original curvature signal profile ΔA and the filtered curvature signal profile ΔA^* of each component are displayed in Figure 2 (d) and (e), respectively. The original curvature signal profile ΔA is an impulse-like sequence but the filtered curvature signal profile ΔA^* has clear peaks and valleys. From Figure 2 (e) one can clearly see that the first component has 15 extrema (marked by crosses) and 3 inflection points (marked by Ts), while the second one has 6 extrema and 2 inflection points. This script was segmented into 2 sequences of static strokes. The first sequence contains 19 strokes, and the second contains 9 strokes. The reconstructed script using these sequences are shown in Figure 2 (c). The *rmse* is 1.02 percent compared with the normalized height of the script.

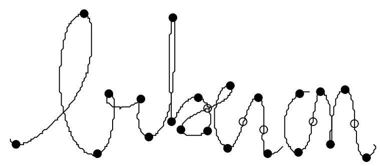
Figures 3 to 4 show a few more examples of the French word segmentation and reconstruction. In Figure 3 there are 10 French words with their landmark points displayed (the middle points are marked by empty squares where appropriate). These words were segmented and then reconstructed. The reconstruction results are shown in Figure 4. In Figure 4 the *rmse* of each word is also displayed.

Figure 5 shows some examples concerning the individual character segmentation. In Figure 5 there are 26 characters (a-z) with their landmark points displayed. These characters were segmented and then reconstructed in a similar manner as dealing with the French words. The reconstruction results are shown in Figure 6.

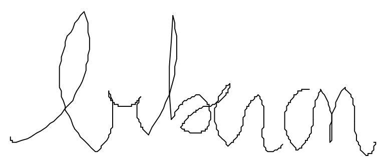
In our experiment, the root mean square reconstruction error varied from character to character and from word to word. The reconstruction statistics is shown in Table 2. with respect to the first data set, the minimum *rmse* was 0.70 percent, while the maximum *rmse* was 2.38 percent. The average *rmse* was 1.20 percent. With respect to the second data set, the minimum *rmse* was 0.09 percent, while the maximum *rmse* was 8.21 percent. The average *rmse* was 1.55 percent. With respect to unipen1a, the minimum *rmse* was 0 percent, while the maximum *rmse* was 11.65 percent. The average



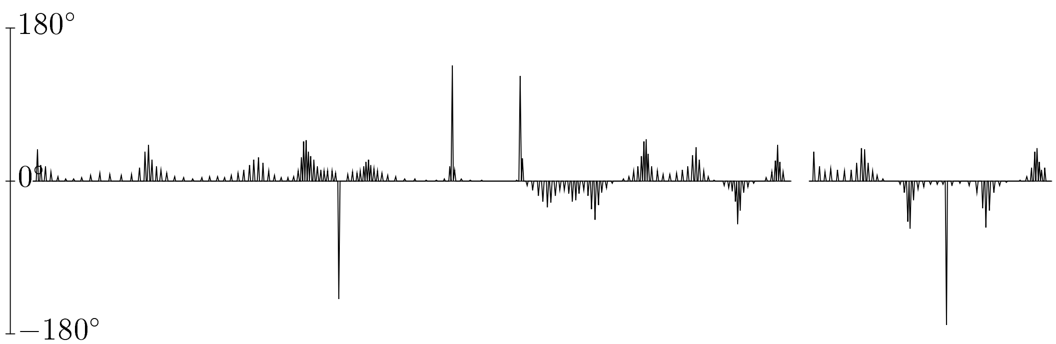
(a) Handwritten script



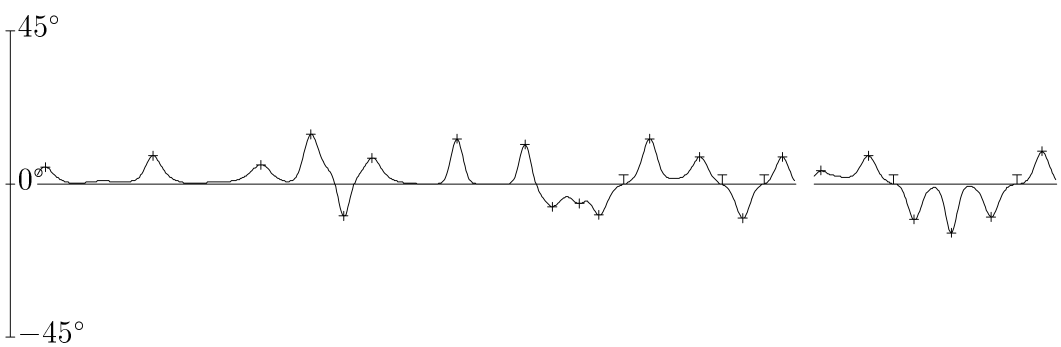
(b) Landmark points



(c) Reconstructed script ($rmse = 1.02\%$)



(d) Original curvature signal profile ΔA



(e) Filtered curvature signal profile ΔA^*

Figure 2: Example of script segmentation and reconstruction

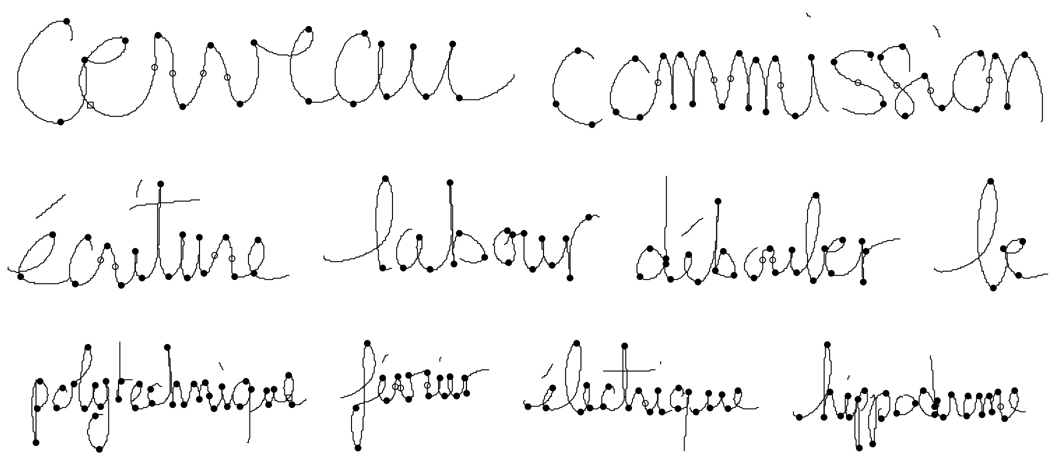


Figure 3: French word segmentation

rmse was 1.32 percent. With respect to unipen1c, the minimum *rmse* was 0 percent, while the maximum *rmse* was 9.22 percent. The average *rmse* was 1.45 percent.

From Table 1 one may also see that in our experiment the middle points detected were less than 2% of all landmark points. In other words, the pen-down and pen-up points and the extrema of curvature and inflection points were more than 98% of the case and hence they played a key role in script segmentation and reconstruction.

5.2 Data compression rate

The 55 French words sampled by the digitizing tablet contain 170 components with 10469 data points. Since each point is represented by a pair of short integers (x, y), the data points require a total number of 41876 ($10469 \times 2 \times 2$) bytes for storage. Using our procedure for script segmentation and reconstruction, these words were segmented into 170 sequences with 1251 strokes. As each component has a pen-down point and each stroke has 3 parameters and all these attributes are represented by floating numbers, the segmented data require a total number of 16372 ($170 \times 2 \times 4 + 1251 \times 3 \times 4$) bytes for storage. In this case, the data compression rate is $(41876 - 16372)/41876 = 60.90\%$.

On the other hand, the 2600 characters contain 3050 components with 89099 data

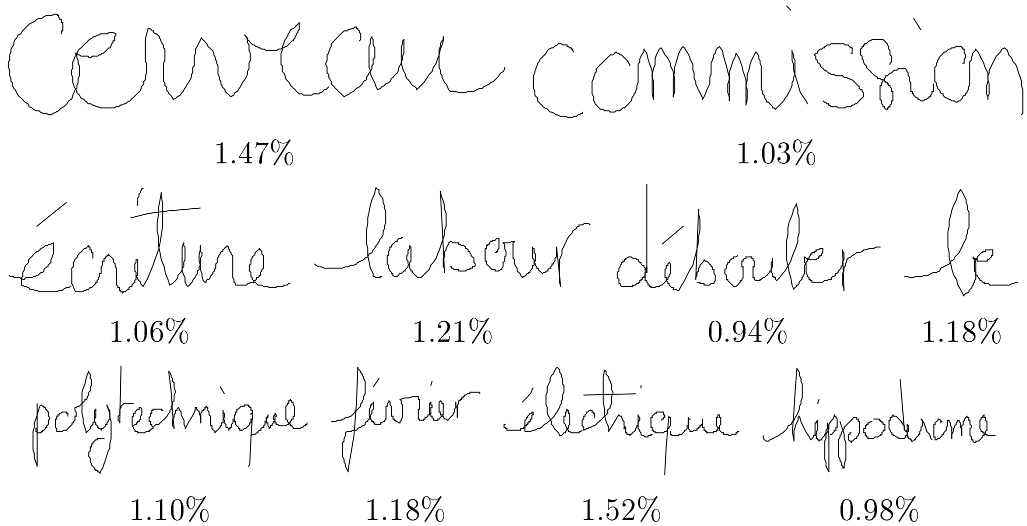


Figure 4: French word reconstruction

points. These data points require a total number of 356396 bytes for storage. When these characters were segmented into 3050 sequences with 12804 strokes, the segmented data require a total number of 178048 bytes for storage. In this case, the data compression rate is $(356396 - 178048)/356396 = 50.04\%$.

The data compression rates concerning unipen1a and unipen1c are given in Table 2. From Table 2 one can see that unipen1a has the highest data compression rate 71.38% but also the highest maximum $rmse$ 11.65%. On the other hand, the data compression rate from unipen1c is 63.02%. This is meaningful because it is the largest data set among the four.

6 Conclusions

In this paper we have presented a procedure for on-line handwritten script segmentation and reconstruction. The procedure detects the landmark points in individual components and segments each component of a script into a sequence of static strokes. The static strokes are derived from the delta log-normal model of handwriting generation.[4,17] The

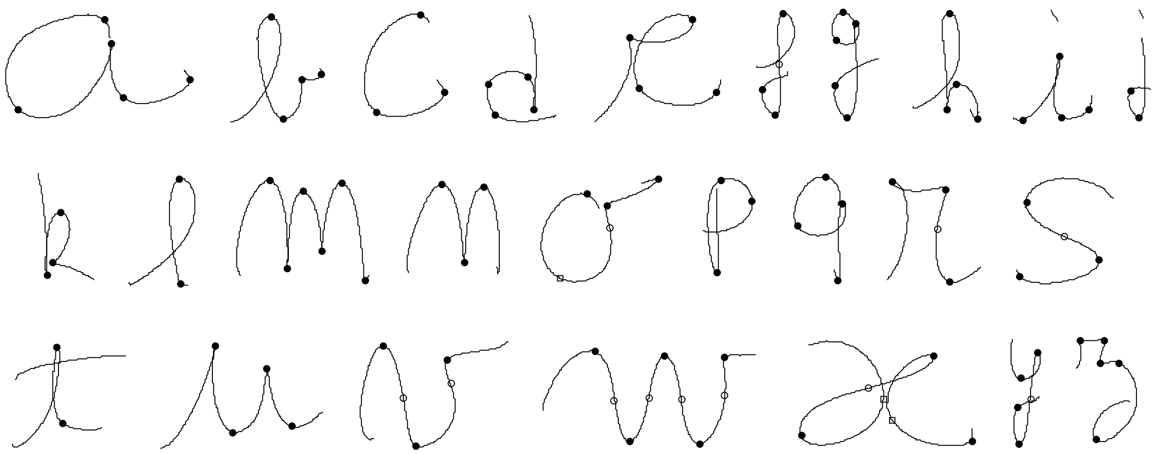


Figure 5: Individual character segmentation

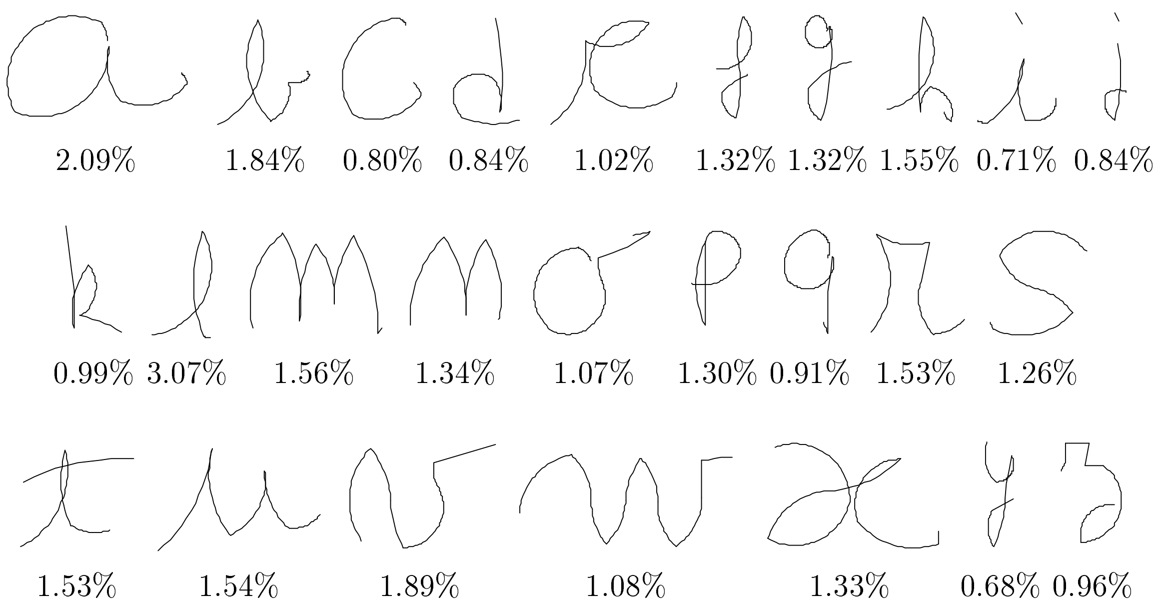


Figure 6: Individual character reconstruction

script is then reconstructed from its stroke sequences. Meantime, the root mean square reconstruction error and data compression rate are measured.

A segmentation and reconstruction experiment has been conducted upon four data sets, the first one contains 55 French words, the second one contains 2600 characters, the third one contains 6519 digits, while the fourth contains 28799 characters. With respect to these data sets, the average root mean square reconstruction error varied between 1.20 to 1.55 percent, while the data compression rate varied between 50.04 to 71.38 percent.

We have shown, through theoretical analysis and experimental results, that our procedure is accurate in locating extreme and inflection point and efficient in script segmentation and reconstruction in terms of data compression rate and root mean square reconstruction error. The limitation of our approach is that it is scale dependent and hence it requires scale normalization.

As one possible application, our procedure can be used for on-line handwriting data compression. The original handwritten script can be segmented and compressed, it then can be reconstructed without losing its shape. As another possible application, the stroke sequences resulting from the segmentation can form a basis for statistical or structural pattern recognition.

Acknowledgements

This research work was supported by NSERC Grant OGP0155389 to M. Parizeau, and NSERC Grant OGP00915 and FCAR Grant ER-1220 to R. Plamondon.

References

[1] L.R.B. Schomaker and H.L. Teulings. A handwriting recognition system based on the properties and architectures of the human motor systems. *Proceedings of the First International Workshop on Frontiers in Handwriting Recognition*, 195-209 (1989)

[2] C.C. Tappert, C.Y. Suen and T. Wakahara. The state of the art in on-line handwriting recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-12, No.8, 787–808 (1990)

[3] L.R.B. Schomaker and H.L. Teulings. Stroke versus character based recognition of on-line, connected cursive script. *Proceedings of the Second International Workshop on Frontiers in Handwriting Recognition*, 265-277 (1991)

[4] R. Plamondon. A Model-based segmentation framework for computer processing of handwriting. *Proceedings of the Eleventh IAPR International Conference on Pattern Recognition*, 303-307 (1992)

[5] M. Parizeau and R. Plamondon. A handwriting model for syntactic recognition of cursive script. *Proceedings of the Eleventh IAPR International Conference on Pattern Recognition*, 308-312 (1992)

[6] J.J. Brault and R. Plamondon. Segmenting handwritten signature at their perceptually important points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.15, No.9, 953-957 (1993)

[7] R. Plamondon, A.M. Alimi, P. Yergeau, and F. Leclerc. Modeling velocity profiles of rapid movements: a comparative study. *Biological Cybernetics*, Vol.69, 119-128 (1993)

[8] X. Li and N.S. Hall. Corner detection and shape classification of on-line handprinted Kanji strokes. *Pattern Recognition*, Vol.26, no.9, 1315-1334 (1993)

[9] R. Plamondon. Handwriting generation: the delta log-normal theory. *Proceedings of the Fourth International Workshop on Frontiers in Handwriting Recognition*, 1-10 (1994)

[10] R. Plamondon, S. Clergeau and C. Barrière. Handwritten sentence recognition: from signal to syntax. *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 117-122 (1994)

[11] D. Lopresti and A. Tomkins. On the searchability of electronic ink. *Proceedings of the Fourth International Workshop on Frontiers in Handwriting Recognition*, 156-165 (1994)

[12] W. Guerfali and R. Plamondon. The delta lognormal theory for the generation and modeling of cursive characters. *Proceedings of the Third ICDAR International Conference on Document Analysis and Recognition*, 495-498 (1995)

[13] M. Parizeau and R. Plamondon. A fuzzy-syntactic approach to allograph modeling for cursive script recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.17, No.7, 702-712 (1995)

[14] R. Plamondon. A kinematic theory of rapid human movements Part I. Movement representation and generation. *Biological Cybernetics*, No.72, 295-307 (1995)

[15] R. Plamondon and W. Guerfali. Why handwriting segmentation can be misleading? *Proceedings of the 13th International Conference on Pattern Recognition*, Vol.3, 396-400 (1996)

[16] X. Li and D.Y. Yeung. On-line handwritten alphanumeric character recognition using dominant points in strokes. *Pattern Recognition*, 30/1, 31-44 (1997)

[17] R. Plamondon and W. Guerfali. The Generation of handwriting with delta-lognormal synergies. To appear in *Biological Cybernetics.*