Pattern Recognition, Vol. 26, No. 3, pp. 419–431, 1993
Printed in Great Britain

0031-3203/93 \$6.00 + .00
Pergamon Press Ltd

NORMALIZING AND RESTORING ON-LINE HANDWRITING

WACEF GUERFALI and RÉJEAN PLAMONDON†
Laboratoire Scribens, Département de génie électrique et de génie informatique, École Polytechnique de Montréal, Campus de l'université de Montréal, 2900 Edouard-Montpetit, Case postale 6079, Succursale A, Montréal (Québec), Canada H3C 3A7

Abstract—Preprocessing and normalization techniques for on-line handwriting analysis are crucial steps that usually compromise the success of recognition algorithms. These steps are often neglected and presented as solved problems, but this is far from the truth. An overview is presented of the principal on-line techniques for handwriting preprocessing and word normalization, covering the major difficulties encountered and the various approaches usually used to resolve these problems. Some measurable definitions for handwriting characteristics are proposed, such as baseline orientation, character slant and handwriting zones. These definitions are used to measure and quantify the performance of the normalization algorithms. An approach to enhancing and restoring handwriting text is also presented, and an objective evaluation of all the processing results.

Feature extraction   Normalization   On-line   Handwriting   Preprocessing evaluation
Restoration

1. INTRODUCTION

With the introduction of electronic tablets (during the 1960s), interest in handwriting recognition focused on research and development to produce interactive technologies like computer keyboard emulators, form filling applications and computer-aided design systems.

A wide range of techniques are used in handwriting recognition, however all the systems developed generally share a common processing sequence, which includes data acquisition, preprocessing, recognition and postprocessing. The nature of data acquisition determines the class of the system (on-line or off-line). It is important, at this stage, to distinguish between on-line and off-line systems. On-line (or dynamic) systems record sampled information about the state of the pentip. This information enables the construction of the discrete functions $X_s(t)$ and $Y_s(t)$, which are the coordinates of the pentip movement sampled at a fixed time (t). Off-line (or static) systems use only the component[1–5] images; information is not available on either the pen movement or on the order of the components.

The preprocessing step used in either on-line or off-line systems presents the first difficulty in handwriting recognition. Many characteristics used in this first stage of processing are quite subjective. Measurable definitions of characteristics such as slant, baseline, zones, etc. are simply not available. This problem makes the results of preprocessing algorithms neither quantifiable nor comparable.

This paper covers three major aspects of the preprocessing and normalizing of *on-line* handwriting. First, an overview of the preprocessing techniques used in on-line systems is presented. Techniques to reduce the amount of data, eliminate imperfections and normalize handwriting are described. Second, normalizing and restoring techniques of handwritten words are treated. Measurable definitions are provided, and new algorithms that detect and correct orientation, slant and handwriting zones are described. Third, an objective evaluation scheme for handwriting preprocessing is proposed. All the algorithms developed are also evaluated and results are presented.

† Author to whom all correspondence should be addressed.

2. PREPROCESSING

On-line systems use preprocessing techniques as a step to simplify the tasks of shape recognition algorithms. This step has a great influence on subsequent processing, and a real impact on the recognition rate.[1,2] Preprocessing techniques for on-line handwriting can be divided into three groups. They are used first of all to reduce the amount of information (filtering and dot reduction); second, to eliminate imperfections (smoothing, wild point correction, hook removal and component connection); and third, to normalize handwriting (deskewing, baseline drift correction, size and component-length normalization).

2.1. *Reduction of information*

Electronic tablets enable the sampling of information about pentip position and status at a fixed sampling frequency. The amount of data transmitted by electronic tablets is usually reduced by eliminating duplicate points and redundant information. This processing step is aimed principally at minimizing the amount of data and reducing recognition time.

Filtering. Filtering, as described by many authors,[3–8] consists in eliminating consecutive points spaced by an interval under a certain threshold. Interpolation can also be used if equally spaced points are needed. Some authors[4,6,9,10] use supplementary conditions to detect regions of greater curvature in cases where they can avoid eliminating significant points and causing confusion (e.g. between pairs of characters like "U" and "V").

Dot reduction. Many authors[3,5–8] use specific filters to reduce dots (as found on "i", "j", etc.) into single points. Dots are located by their small size, their large angular variations (between points) and their relative positions. This operation can be very delicate, especially in noncursive writing, where the dot sizes are important and the distinction between some characters (!, i, :, ;, „ =) is far from obvious.

It is also important to be aware that all these spatial filtering operations result in a loss of the kinematic information of writing (curvilinear and angular velocities).

2.2. *Elimination of imperfections*

Many imperfections can occur with the use of electronic tablets. Imperfections are mostly attributed to hardware problems, limited accuracy of the tablets, the digitizing process, erratic handmotion and the inaccuracies of pen-down indications.[1,3,5,7–17] Several techniques are used to detect these kinds of imperfection and attempt to correct the data points involved.

Smoothing. Smoothing is one of the most popular preprocessing techniques used in on-line systems.[3–5,7,8] It is used to eliminate (or reduce) the effect of either hardware problems or erratic handmotion. Smoothing usually consists in averaging point positions $P_i(x_i, y_i)$ with respect to their neighbours $(P_{i-n} \cdots P_{i+m})$. The coefficients $(C_{i-n} \cdots C_{i+m})$ and the number of neighbours used to calculate averages determine the filter type, its order and its frequency band. This kind of processing can be very fast, and computation is usually performed during the acquisition step. It is important to realize, however, that smoothing reduces real angular discontinuities as well as imperfections, and a drastic smoothing can easily transform a "V" into a "U".

Wild point reduction. Wild points are occasional spurious points detected by the electronic tablet and are mainly due to hardware problems. Major improvements to electronic tablets have reduced this kind of imperfection, but software processing is still used to completely eliminate this problem. Wild points are detected by high-velocity variations. Thresholds, based on the limits of handmotions,[14,15] can be used to detect and correct this kind of imperfection.

Hook removal. Hooks are imperfections at the beginning and at the end of components, a problem primarily due to erratic handmotion and the inaccuracies in pen–paper contact detection that occur during the pen-up and pen-down movements.[7,8] Hooks can be detected by their location (at the beginning and at the end of components), their smallness and their great angular variations. This preprocessing technique has been used by many authors.[9,13,16] Components are generally processed at their extremities where thresholds on the length and the angular variation between points are used to eliminate the hook portions of the components.

Component connection. Two kinds of imperfection related to component connection are observed. The first problem, described by Mandler[13] and Ward and Kuklinski,[17] involves "straight lines" at both ends of a component, usually retracing themselves in a significant way. These imperfections can be detected by high-velocity conditions and small angular variations between successive extremities of components.

The second problem, described by Brown and Ganapathy,[1] is usually the result of inaccuracies in pen-down detection. Pen-up detection occurs in a normally connected component. This imperfection can be detected by the angular continuity and the shortness of the distance between successive extremities.

2.3. *Normalization*

The normalization process is used to reduce the effect of handwriting variations and to simplify shape recognition algorithms. Basically, this processing is used to normalize the slant, the baseline orientation and the size of words, characters and components.

Baseline drift correction. As described by many authors,[1,7,8,18,19] baseline correction is aimed at bringing the orientation of writing to the horizontal level. Baseline correction is an important normalizing transformation which affects the efficiency of subsequent processing, such as segmentation, shape recognition and even other preprocessing techniques such as size normalization and deskewing.

Normalization of size and component length. Normalizing writing size or component length is used to simplify the signal comparison process. Authors such as Brown and Ganapathy[1] and Burr[2] have described the size normalization of handwritten words: Doster and Oed,[11] Mandler[13] and Plamondon and Nouboud[5] use the normalization of isolated characters for subsequent comparison. Component length normalization has also been used by Guerfali[20] and Plamondon and Nouboud[5] to constrain components to a specified number of points (filtering or interpolation are used to reduce or increase, respectively, the number of points involved).

Deskewing. Deskewing as described by many authors[1,2,12,21,22] is a measure and a correction of the slant of a word or character. This normalizing transformation constrains the projection of characters, *on the x-axis*, to be spatially separable (t-bars, dots, accents, etc. are not considered), which helps the character segmentation process of cursive words.[12] Different techniques have been used to detect and correct word slants. Bozinovic and Shriari[21] measured the local slant in regions that present low pixel density projected on the Y-axis. The global slant is the average of local slants. Brown and Ganapathy[1] measured the local slant of central regions of the word. Burr[2] used a very different technique, based on kinematics. A relation between the y and x velocities was used to measure the global slant of writing.

3. NORMALIZING AND RESTORING

As we have seen, three features are generally extracted from handwriting: baseline orientation, writing zones and word slant. The major difficulty in the extraction of this information arises from the fact that all the features commonly used are subjective. Many authors describe different algorithms that detect and correct those characteristics, but no measurable results have been presented, which make the comparison of these techniques impossible. To avoid this problem, we are proposing some operational definitions in this paper to measure these characteristics and also to evaluate the detection algorithms.

New algorithms for baseline, slant and zone detection are proposed, together with an approach for writing enhancement and restoration. The results of these detection, normalization and restoration steps are objectively evaluated. The evaluation step is usually neglected in the literature because of the major difficulties involved in finding objective, measurable and universal criteria. The criterion usually used to evaluate normalization processes is the recognition rate. This criterion is quite inadequate, however, due to the fact that the sensitivity of the recognition algorithm to the parameters modified by the normalizing transformation is ignored, which means that the real effect of the normalizing transformation is simply unknown.

A database of 55 French words, each written by five writers (for a total of 275 words), was used for evaluation. The database contains supplementary information, introduced and revised by two readers. Supplementary information indicates the kind of each component (text, point, accent, cross, etc.), the zone boundaries of each word and the type of each extremum (maximum or minimum, and the zone to which it belongs). This information was kept as a reference and was very helpful for automatic measurement and comparison of algorithm performances.

3.1. *Baseline detection and correction*

Writing orientation is defined as the deviation between the baseline and the horizontal axis. In spite of the fact that the problem of measuring this deviation seems trivial, no robust algorithms are available to solve it in general. The major difficulty arises from the presence of baseline distortions due to zone variations. These zone variations can be very large, which makes baseline location ambiguous. Baseline detection is normally the first normalizing process in handwriting preprocessing, however, the algorithms developed for baseline, slant and zone detection take advantage of an interaction between them.

Many authors have described algorithms for baseline correction,[1,7,8,18,19] but no measurable definitions of baseline have been made. The evaluation of these algorithms is completely subjective and thus no comparison is possible.

To solve this fundamental problem, a baseline definition is proposed here: *the writing baseline is defined as the best fitting, straight line passing through the minima of the letter bodies.* The line is determined by the least squares method (LSM). This definition enables precise measurement of word orientation, and the evaluation and comparison of different algorithms.

Based on this definition, a two step algorithm that locates and corrects writing orientation has been developed.[20] The first step in evaluation and correction is still very helpful for detecting major deviations (more than 20°) that could compromise the precise location of extrema. The first evaluation consists in spatially dividing the word into eight equal and successive regions. In each region, we evaluate the centre of mass of all the points crossing this part of the word. These successive mass centres (a total of eight points) permit the calculation of an approximation of the real baseline. This evaluation is aimed *only* at detecting major deviations, which can cause numerous errors in locating the real minima of the letter bodies. The second step, based on minima analysis, locates the baseline as defined above more accurately.

The minima analysis method uses a retroactive process consisting of successive estimations and corrections until satisfactory results are achieved. The baseline is estimated from a prediction of zone boundaries, which allows the subsequent elimination of superfluous points. Only minima within the median zone are considered in the baseline calculation. The least squares method is used to determine the best fit. The word is then rotated by an angle equal to the baseline slant. This process is repeated until an estimated baseline angle of less than 2° is reached, which is considered an acceptable limit.

The baseline detection algorithm has been evaluated relative to the baseline reference estimated from the supplementary information available in the database discussed previously. The baseline orientation is measured by the least squares method, as expressed by the following formula:

$$\text{Orientation} = \text{Arctan}\left(\frac{n*\sum_1^n (t_i*y_i) - \sum_1^n t_i*\sum_1^n y_i}{n*\sum_1^n t_i^2 - \left(\sum_1^n t_i\right)^2}\right) \tag{1}$$

where n is the number of minima, (x_i, y_i) the coordinate of a minimum point i, and t_i the sampling instant.

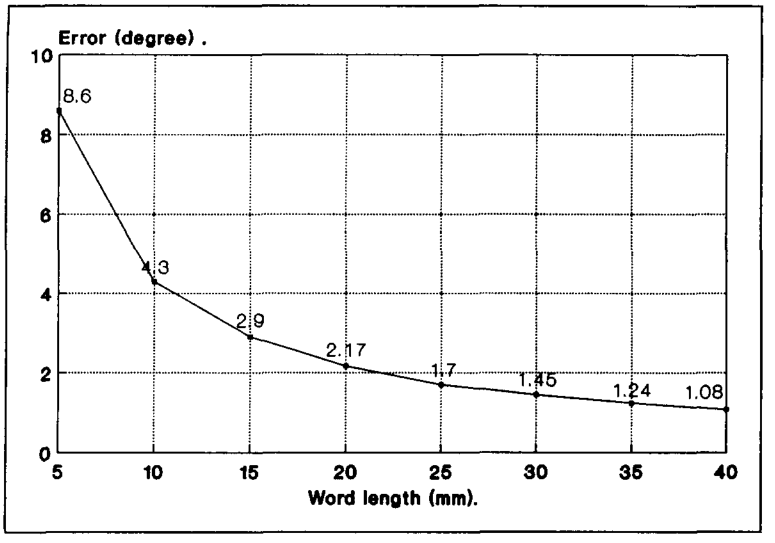

Fig. 1. Relation between word length and estimated error.

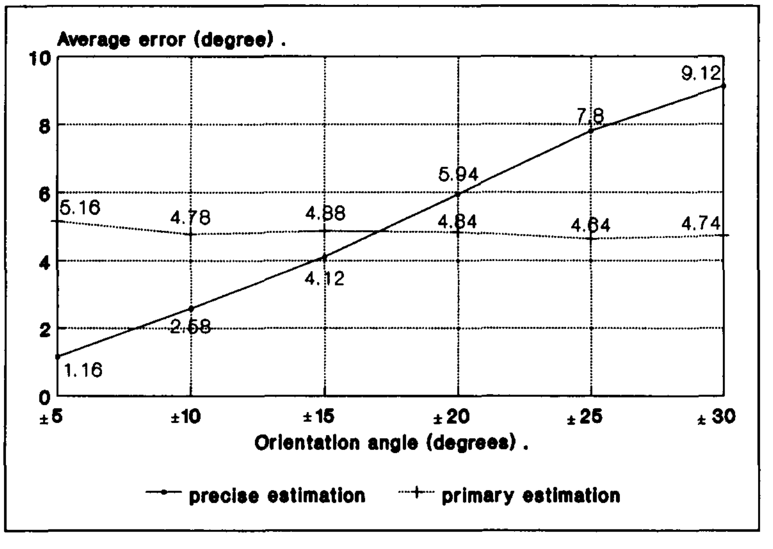

Fig. 2. Average errors for the primary and precise locations of word baseline.

To compare the results of the detection algorithm to the reference, it is important to know the accuracy (of the baseline measure) of the reference. With an evaluation of the errors made on this estimation, we can fix the accuracy of the references. In fact, the error made on the estimate depends on three factors:

- First, the accuracy of the electronic tablet (PENPAD 300), which is about ($\pm 5 \times 10^{-3}$ in. $\approx \pm 0.127$ mm).
- Second, the error on the extremum location, which depends on the sampling rate (about 100 Hz) and the average velocity around the extrema (evaluated to about 1.4 cm s^{-1}). As we can see, at a sampling rate of 100 Hz, the average displacement around the extremum is around 0.14 mm, and, if we consider the worst case, where the sampled points are furthest from the real extremum, the absolute error on the extremum location cannot go beyond 0.07 mm (0.14/2 mm).
- Third, the average quadratic error on the estimations (evaluated to about 0.032 mm^2).

The maximum total error is thus the sum of the three errors $\pm(0.127 + 0.07 + \sqrt{(0.032)}) \approx \pm 0.38$ mm. This error fixes the uncertainty bandwidth for the extrema locations.

The error on the reference orientation depends *only* on the bandwidth estimated above, and the word length. This error is estimated by the absolute value of the angle difference between the two diagonal lines of the box formed by the bandwidth (as a box width) and the word length (as a box length). Figure 1 shows the

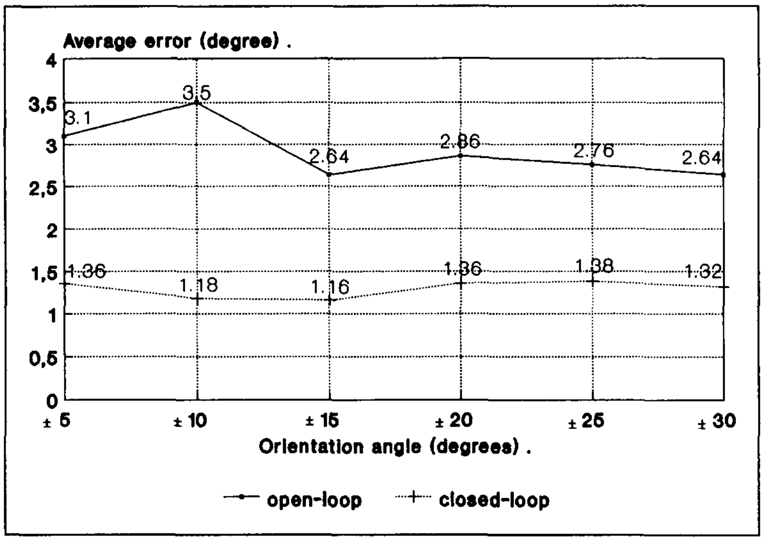

Fig. 3. Average errors of the closed-/open-loop algorithm for baseline detection and correction.

relation between the word length and the estimation error (for an uncertainty bandwidth of ± 0.38 mm).

The average length of the words used in the evaluation process is about 25 mm, which means that the average error on the references is around 1.7°.

The algorithm of baseline detection and correction presented earlier involves two principal steps. The first correction trends to eliminate large deviations, while the second correction locates the real baseline with greater precision. A separate evaluation of these two methods, where we measure the resulting error of each step, enables us to show the pertinence of their combination. Figure 2 shows the results of the separate evaluation of these two steps, and it can clearly be seen that the use of the primary evaluation guarantees a constant error (around 4.8°) for all the ranges of orientation ($\pm 5°$ to $\pm 30°$), while the precise location is more adequate for absolute angles less than about $\pm 17°$. This evaluation shows the importance of joining the two methods of baseline detection to optimize the location results for a wide range of angles.

The effect of the retroactive process (closed loop) was also evaluated, as shown in Fig. 3. The average error of the closed loop algorithm is around 1.3° for a wide range of angles ($\pm 5°$ to $\pm 30°$), while the error on the references is about 1.7°. These results confirm that the results of the baseline detection algorithm are very satisfactory.

3.2. *Slant detection and correction*

The handwriting slant is known as the deviation between the principal axis of characters and the vertical axis. Slant correction is often used to simplify the segmentation procedure.[12]

The slant of handwriting is, in fact, a subjective characteristic, which is difficult to measure (accurately), and very unclear in many cases (example: c, e, o, x, etc.). Slant can be identified much better by a range of angles, but the normalizing process calls for the angle value that best estimates the whole word slant.

In our approach, the combination of the two static methods used by Bozinovic and Shriari[21] and Brown and Ganapathy[1] has been considered. The approach based on kinematic measures used by Burr[2] was eliminated because of a fundamental problem: the measure of the slant was based on a ratio of horizontal and vertical velocities. The measure of the slant is then affected by the relative importance of regions that do not contain slant information (x displacements). This problem becomes more apparent in long words where x displacements are large.

The algorithm developed combines the approach used in reference (21) and zone detection. Figure 4 shows the major steps of the process. After zone location, as described in Section 3.3, three regions are considered: the upper zone, the lower zone and the central region of the middle zone (Fig. 4(a)). Within the three regions extracted, observation windows are isolated (Fig. 4(b)). Each window is divided into an upper and a lower part, where a centre of mass of the points passing through this part is measured. The local slant of the window is then estimated by the line joining the mass centres of the superior and inferior half-windows (Fig. 4(c)). The global slant of the word is fixed to the average of local slants.

The slant correction, also called the deskewing process, consists of a word transformation relative to the baseline. The slant of the characters is estimated relative to the vertical axis with the assumption that the baseline is horizontal. The transformation is illustrated as follows:

$$\begin{bmatrix} 1 & -\tan(\theta) \\ \tan(\theta/2) & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}. \quad (2)$$

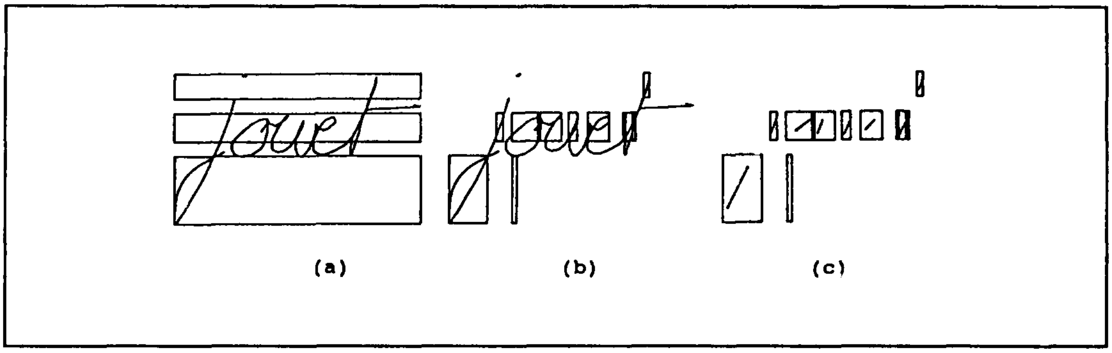

Fig. 4. Slant measure.

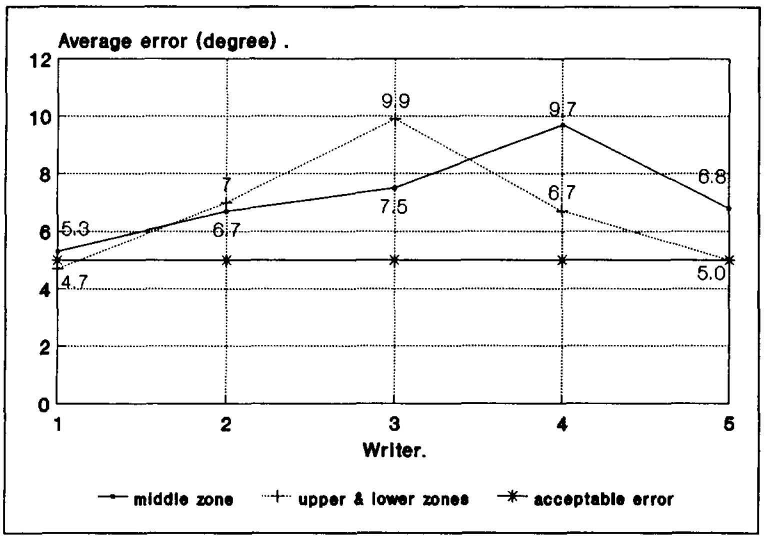

Fig. 5. Average error of the slant estimation algorithm: comparison between zones.

The evaluation of the slant detection algorithm is quite a difficult process. The problem of evaluation mainly stems from the fact that automatic quantification of a unique reference is not possible, even with human intervention. To fix a reference for our algorithm evaluation, we asked ten subjects to determine the slant of each one of the 275 words in the database (discussed above). The averages of the selections were considered as references, while standard deviations were treated as an indication of acceptable error. Compilation of these results has shown that the standard deviation is about $\pm 5°$.[20]

The first goal of the evaluation process was to determine if there is any zone predominance in slant detection. This evaluation is aimed at determining if ascender and descender portions are more important in slant detection than the character bodies. A separate measure of the slant, in each of the three zones, is compared to the references. The result of this experiment shows (Fig. 5) that there is no obvious predominance of one zone relative to another. We can therefore say that there is an equivalent amount of slant information in each of the three zones. Because no zone is predominant in the slant evaluation, the global slant of the word is measured by the average of the three local zone slants.

The second goal of the evaluation process was to measure the effect of the closed-loop process, where retroactive processing (detection–correction) is used, vs. open-loop processing, where only the first evaluation of the slant is considered. The errors committed by those estimations are shown in Fig. 6, and, as we can see, closed-loop processing and open-loop processing are essentially equivalent.

The results of this evaluation of the slant detection algorithm have shown that estimation errors (absolute values) vary between 4.6° and 8.0° (average 6°) for the five scriptors (open-loop processing, Fig. 6). This error is considered as acceptable, knowing that the accuracy of the subjects slant was about 5° (absolute value). The second interesting result, shown by the slant evaluation (Fig. 5), was that slant information is available in the three writing zones and that no predominance was detectable. Thus we can say that slant information can

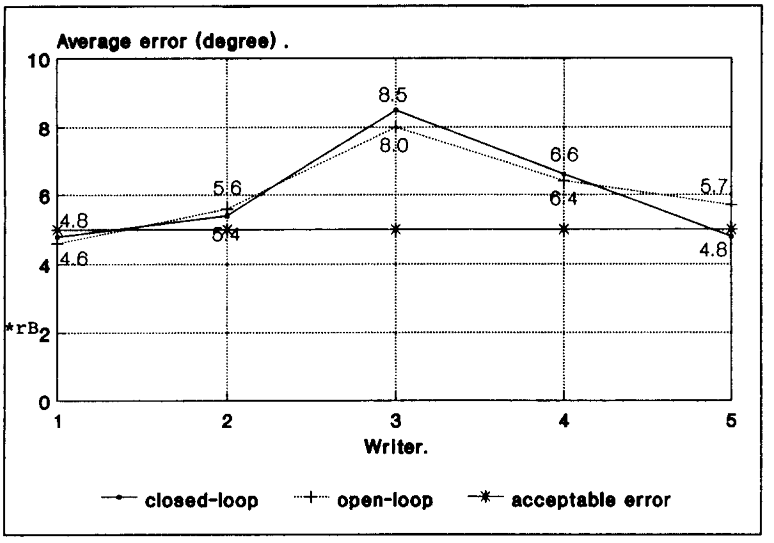

Fig. 6. Average error of the slant estimation algorithm: open loop vs. closed loop.

be measured in any zone in the body of the letter (middle zone) as well as in its descenders/ascenders.

3.3. *Zone detection*

Cursive writing is principally composed of lower case characters, which occupy three vertical zones. The middle zone, where the bodies of the letters reside, and in the upper and lower zones, where the ascenders and descenders are headed.

The extraction of zone information is important in the normalizing and the recognition processes. Normalizing allows each letter to be indirectly assigned to one of the three character groups, which are the central characters (a, c, e, i, m, n, o, r, s, u, v, w, x), the ascenders (b, d, f, h, k, l, t) and the descenders (f, g, j, p, q, y, z).

Zone location, as described by many authors,[2,21] is the detection of the boundaries between the three vertical regions described above. The problems related to this detection are due to two main factors. First, handwriting does not always respect typographic norm of the alphabet. Second, zone variations can be very large, which makes the location of a unique boundary unclear.

Two classes of algorithms have been proposed in the literature for zone detection. The first, known as the histogram method, is used for both off-line[21] and on-line systems[2] and consists of the analysis of the histogram of the horizontal projection of points. A threshold is used to locate the boundaries between the middle, the upper and lower zones.

The second method, known as the extrema method, is used mainly for on-line systems[20] and consists in the analysis of the writing extrema, an attempt to find horizontal lines, which can be considered as zone boundaries.

A comparative study of these two methods shows that the histogram method is more efficient in detecting the existence of the various zones. However, due to the use of thresholds, there is difficulty with this method in locating the real boundaries with any accuracy. Contrary to the histogram method, the extrema method has more difficulty in detecting distinct zones, especially when zone variations are large, but can locate optimal boundaries with greater precision when the number of zones is known. This suggests, in fact, that the two methods can be complementary. The histogram method can be used to determine information like the number of zones (word with ascenders, with descenders, etc.), and the extrema method, knowing the class of word involved, can easily and accurately detect the real boundaries. Based on this idea, we have developed a zone detection algorithm that combines the histogram and extrema methods, making the most of the advantages of each and avoiding their respective disadvantages.

The first stage of the algorithm using the histogram method is an attempt to determine the class of words involved (by "class", we mean either words with, or words without, ascenders or descenders). A threshold is used to detect the central zone, and an analysis of the histogram determines the class of the word.[20]

Once the class of word is determined, the extrema method is used to detect the optimal boundaries between the different zones. The boundaries are defined as *the upper maximum and the lower minimum of the middle zone*. This definition includes the maximum of the zone variations of the middle zone. (Typographic deformations are excluded from this definition.) An example of the different steps of the zone detection algorithm is shown in Fig. 7. Histogram H (Fig. 7(a)) is compared to the four references (H_1, H_{2s}, H_{2i} and H_3). After the histogram classification, the word is associated with one of the four classes. This association enables application of the second method, which is to classify the extrema into one of the three zones

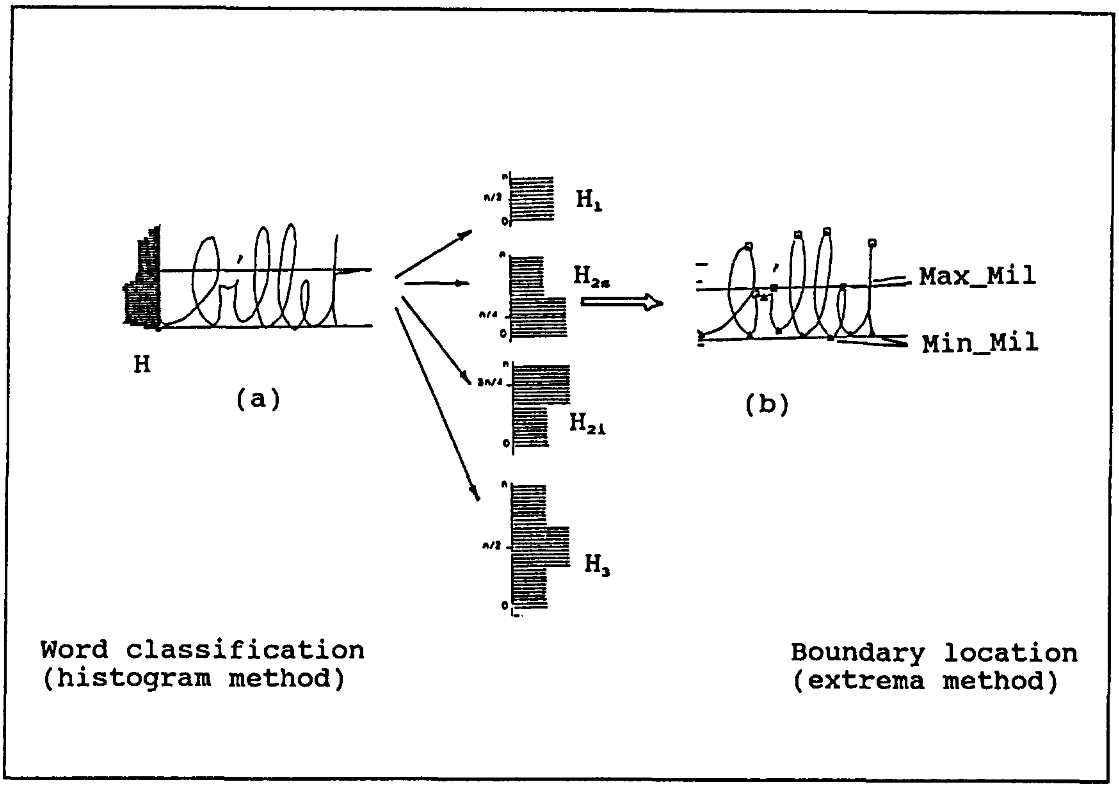

Fig. 7. Example of zone detection.

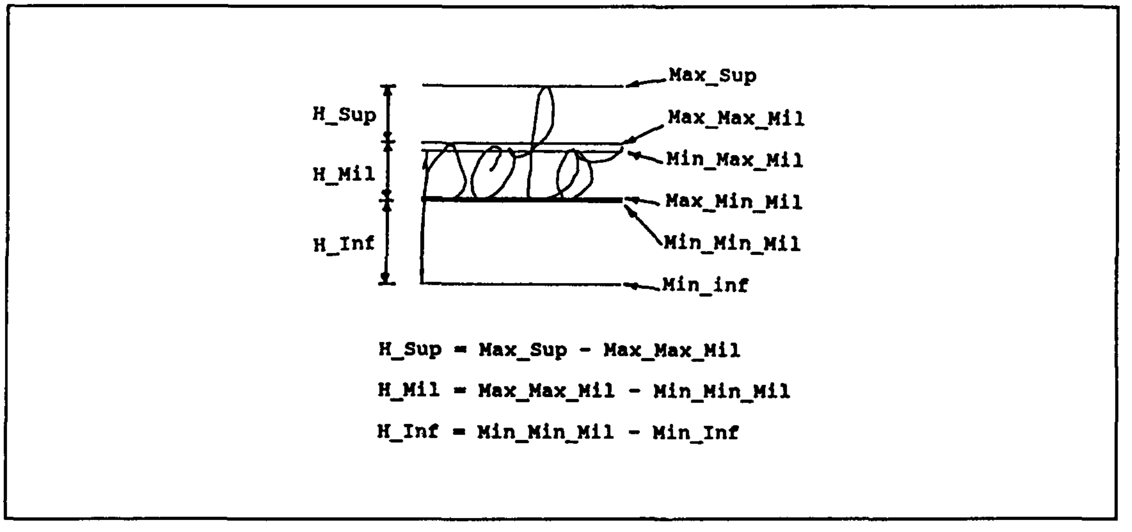

Fig. 8. Zone variation measures.

(Fig. 7(b)). The boundaries of the middle zone are then fixed to the upper maxima and the lower minima classified in this zone. Two kinds of problems are also treated by the algorithm: the case where no maxima or minima belong to the middle zone, so that no boundary can be selected, and the case where one zone is too small, relatively, to be considered as distinct. In these two cases, the solution consists in joining the problematic zone to the adjacent one (upper or lower zone to middle one), and in restarting the extrema analysis process.

The evaluation of the zone detection algorithm is accomplished relative to the references available in the database used. The reference boundaries used in the evaluation of the proposed algorithm are chosen by locating the upper maxima and the lower minima that belong to the median zone. All the information needed to evaluate this algorithm is available in the database used.

The estimation error depends on two principal factors; first, the accuracy of the electronic tablet (± 0.127 mm), and, second, errors in the extrema location (estimated above at 0.07 mm). The cumulative error is thus around ± 0.16 mm ($\pm 0.127 + \pm 0.07/2$).

To evaluate the performance of the zone location algorithm, a new entity that reflects the relative importance of the error committed has to be defined. To define this entity we must first consider the quality of the writing, which is measured by the variations of the median zone, shown in Fig. 8, and expressed by formula (3). Secondly, the relative influence of the error on the zone detection, which is the average ratio of the absolute error and the height of the zones involved, expressed by formula (4).

as shown in Fig. 10, errors committed by the estimations are 2–5 times lower than the writer's zone variations.

3.4. *The restoration process*

The restoration process is one step in handwriting enhancement, using a nonlinear transformation, and is designed to normalize writing zones and proportions. This enhancement can be very helpful for subsequent recognition algorithms, and can also be used in applications where we want to produce a restored handwritten document. A normalizing transformation has also been presented by Burr,[2] but no evaluation of this technique is available. The difficulties arising in evaluation have two principal causes: first, this transformation is very critical and can result in the introduction of a great deal of distortion; and second, there is no way to measure this distortion automatically.

$$V(\%) = \frac{1}{2}\left(\frac{\text{Max\_Max\_Mil} - \text{Min\_Max\_Mil}}{\text{H\_Sup} + \text{H\_Mil}} + \frac{\text{Max\_Min\_Mil} - \text{Min\_Min\_Mil}}{\text{H\_Mil} + \text{H\_Inf}}\right) * 100 \qquad (3)$$

where $V(\%)$ is the variation of the median zone.

The performance of the zone detection algorithm was measured in three steps in which word classification and boundary location, and their combination with the retroactive corrections, are evaluated separately.

Evaluation of the classification step, which proceeds with the histogram method, has shown that the major part of the errors can be traced to a bad classification. In fact, 22% of the words were wrongly classified. Most of the errors (63%) occur with one-zone words, while three-zone words are usually well detected (3% of the errors). Errors occur mostly when one- and two-zone words are considered in the same way as three-zone ones (68% of the errors). This evaluation clearly shows the weakness of the histogram method.

Boundary location causes far fewer problems. A measure of the errors is expressed by formula (4), as can be clearly seen in Fig. 9. In all cases, the average error for the five writers is four to five times lower than the average zone variations measured for each writer. This figure shows the performance of the extrema method in zone location, when the word type is known.

The restoration procedure consists of four principal steps:

(1) *The extraction of critical points.* Two classes of information were considered. The first class of points is composed of extrema of the horizontal (X) and vertical (Y) axis. The X extrema enable the collection of information on character width, while the Y extrema collect information on character height and zone sizes. The second class is composed of points that contain information on angular variations. A four-direction encoding is used, and only the centres of each segment obtained are saved. Only these two classes of points are considered in the restoration and reconstruction processes.

(2) *Extrema classification.* The classification of extrema consists in assigning each Y extremum to one of the three writing zones (upper, middle and lower zones). The Y extrema used in the restoration process are the upper's maxima, the median's maxima, the median's minima and the lower's minima.

(3) *Correction.* The extrema assigned to one of the

$$\text{Error}\ (\%) = \frac{1}{2}\left(\frac{|\text{Sup\_Boundary} - \text{Max\_Max\_Mil}|}{\text{H\_Sup} + \text{H\_Mil}} + \frac{|\text{Inf\_Boundary} - \text{Min\_Min\_Mil}|}{\text{H\_Mil} + \text{H\_Inf}}\right) * 100. \qquad (4)$$

The results of combining the detection and the location steps are shown in Fig. 10. The effect of the retroactive correction is appreciable, the total classification error drops from 22 to 15%. The final results of the zone location algorithms remain very satisfactory,

three zones are relocated at optimal positions. Zone sizes, proportions and zone variations can be controlled by Y extrema positions. The other critical points are relocated as intermediate points between successive Y extrema and component ends.

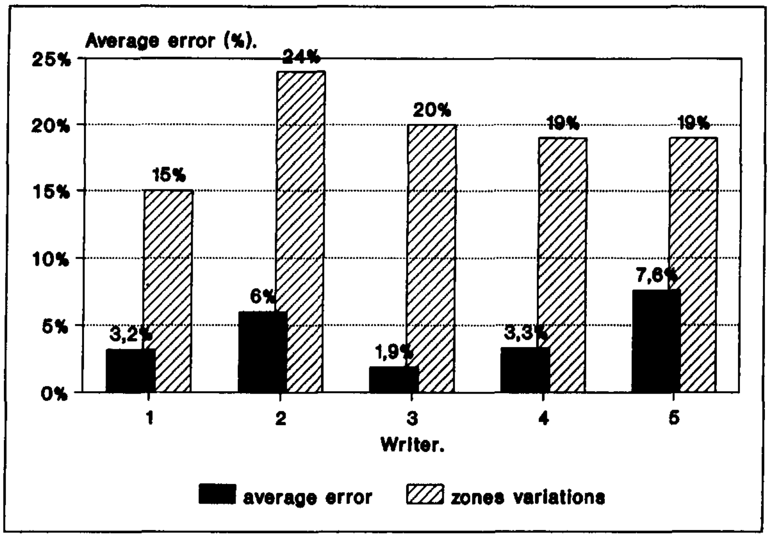

Fig. 9. Average zone location errors: extrema method.

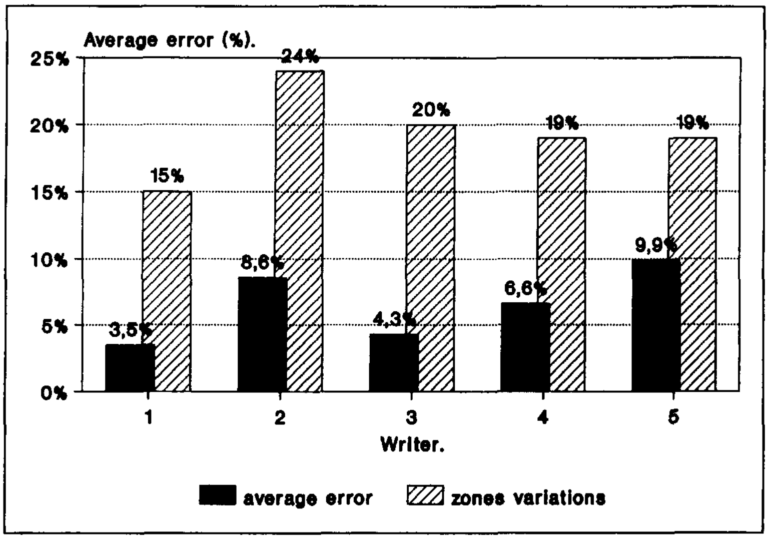

Fig. 10. Average zone location errors: final algorithm.

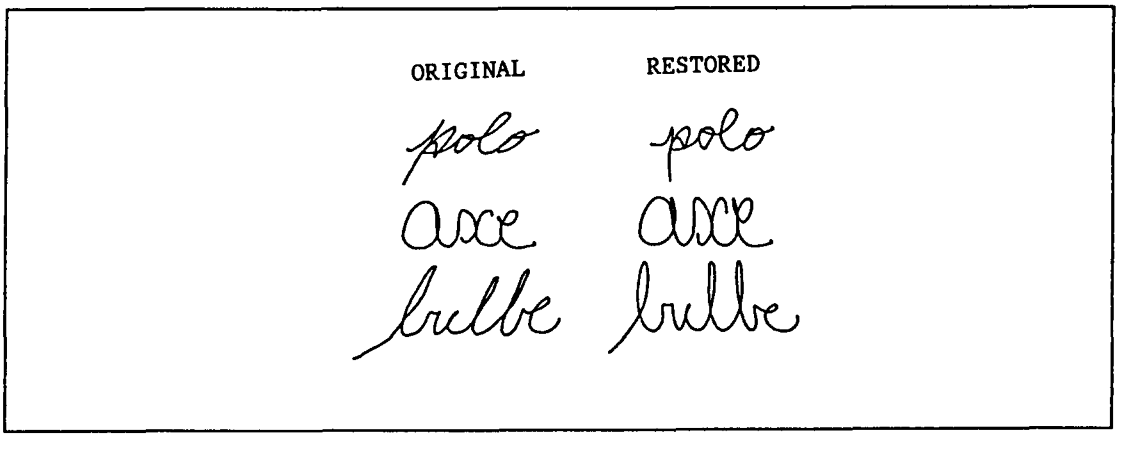

Fig. 11. Word restoration.

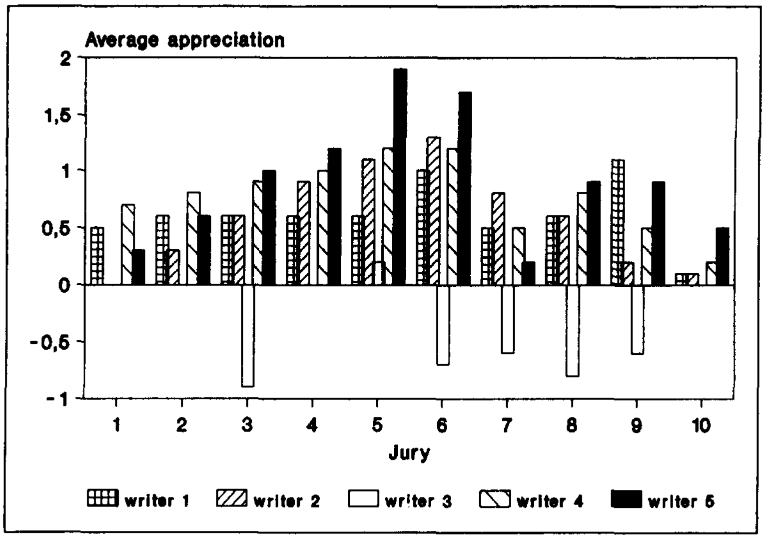

Fig. 12. Jury's average appreciations.

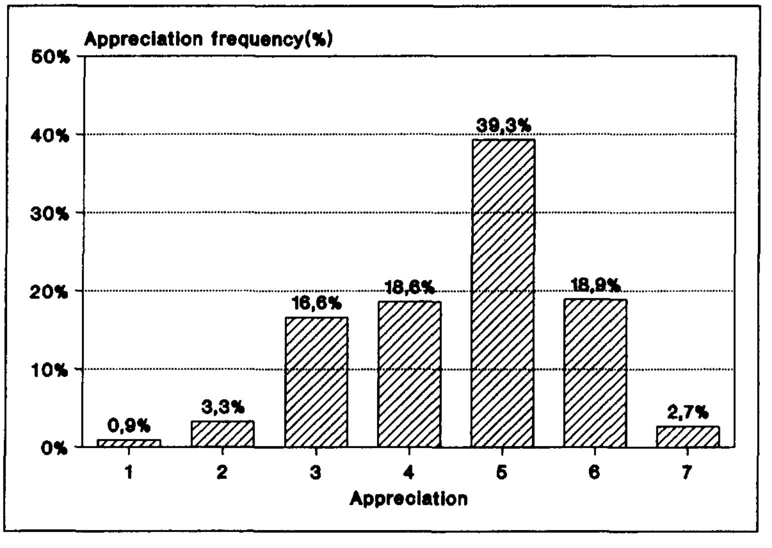

Fig. 13. Distribution of jury appreciations.

(4) *Reconstruction.* Component reconstruction is achieved by a cubic spline interpolation between successive critical points. The first and second derivatives are maintained constant to preserve curvature smoothness.

Figure 11 shows some results of word restoration, where zone variations and proportions are controlled.

Evaluating restoration techniques is a very subjective process. A jury of ten people was formed to evaluate the results of the restoration algorithm on the database of 55 words written by five writers. Each juror had to fill in appreciation forms where he or she had to compare the original word to the restored one (appearing in random order). The appreciation scale ranged from 1 to 7 as follows:

1. (a) is much better than (b).
2. (a) is better than (b).
3. (a) is a little bit better than (b).
4. (a) is equivalent to (b).
5. (a) is a little bit worse than (b).
6. (a) is worse than (b).
7. (a) is much worse than (b).

The evaluation results allow us to quantify and evaluate the performance of the proposed restoration algorithm. The results of this evaluation are presented in Fig. 12, scaled from 3 to -3, where 3 is a clear improvement and -3 is a clear deterioration.

The jury's appreciation corresponds to their relative evaluation of the original and restored words. The appreciations vary with the quality of the original writing, which means that the better the original writing the more indifferent the jury was to the restoration. The evaluation distribution, shown in

Fig. 13, indicates that the results of the restoration process were positive, in general: 61% of the responses show that there is a real improvement, 18% show that the words are of an equivalent quality and only 21% of responses reveal a preference for the original handwritten words. The major comment of the jury was that the normalizing and restoring of slant and zones is, in general, of benefit to the recognition process, but its "synthetic" form might be quite disturbing for some readers.

4. CONCLUSION

This paper covers three major aspects related to the preprocessing of on-line handwriting. We first presented an overview of the principal techniques of on-line preprocessing. We covered the various aspects of data reduction, imperfection problems and handwriting normalization. In the second part of the paper were revealed the principal problems of feature extraction and handwriting normalization. New algorithms were proposed for slant, zones and baseline detection. There was a major emphasis on the evaluation of these algorithms to quantify their performances. We have shown that the principal evaluation difficulties are first, the lack of objective, measurable and universal definitions of writing characteristics, and second, the absence of "standard" databases that would enable algorithm comparison. This paper proposes a new scheme for the objective evaluation of normalization and restoration algorithms, and some measurable definitions were proposed for the baseline and the zones which allow us to quantify the errors of detection algorithms. We have also presented some approaches for evaluating subjective characteristics such as handwriting slant and writing quality. All the algorithms proposed were evaluated, and the results shown were found very satisfactory in all cases.

REFERENCES

1. M. K. Brown and S. Ganapathy, Preprocessing techniques for cursive script word recognition, *Pattern Recognition* **16**, 447–458 (1983).
2. D. J. Burr, A normalizing transform for cursive script recognition, *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, Vol. 2, pp. 1027–1030, October (1982).
3. F. Nouboud and R. Plamondon, On-line recognition of handprinted characters: survey and beta tests, *Pattern Recognition* **23**, 1031–1044 (1990).
4. R. Oed and W. Doster, On-line script recognition—a user friendly man–machine interface, *COMPINT 85*, pp. 741–743 (1985).
5. R. Plamondon and F. Nouboud, On-line character recognition system using a string comparison processor, *IEEE 10th Int. Conf. on Pattern Recognition*, Atlantic City, Vol. 1, pp. 460–463, June (1990).
6. C. C. Tappert, Adaptive on-line handwriting recognition, *IEEE 7th Int. Conf. on Pattern Recognition*, pp.1004–1007, August (1984).
7. C. C. Tappert, C. Y. Suen and T. Wakahara, On-line handwriting recognition—a survey, *9th Int. Conf. on Pattern Recognition*, Vol. 2, pp. 1123–1132 (1988).
8. C. C. Tappert, C. Y. Suen and T. Wakahara, The state of the art in on-line handwriting recognition, *IEEE Pattern Analysis Mach. Intell.* **PAMI-12**(8), 787–808 (August 1990).
9. W. W. Loy and I. D. Landau, An on-line procedure for recognition of handprinted alphanumeric characters, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-4**(4), 422–427 (July 1982).
10. F. Nouboud and R. Plamondon, A structural approach to on-line character recognition: system design and applications, *Int. J. Pattern Recognition Artif. Intell.* **5**(1–2), 311–335 (1991).
11. W. Doster and R. Oed, Word processing with on-line script recognition, *IEEE Micro* **4**, 36–43 (October 1984).
12. J. M. Kurtzberg and C. C. Tappert, Handwriting recognition by substroke segmentation and composition, *IBM Tech. Disclosure Bull.* **27**(9), 5167–5170 (February 1985).
13. E. Mandler, Advanced preprocessing technique for on-line script recognition of nonconnected symbols, *Proc. 3rd Int. Symp. on Handwriting and Computer Applications*, pp. 64–66, July (1987).
14. R. Plamondon and B. Clement, Analysis of handwriting asymmetries for the context of an intrinsic generation model, *Human Movement Sci.* **10**, 193–221 (1991).
15. R. Plamondon and F. J. Maarse, An evaluation of motor models of handwriting, *IEEE Trans. Syst. Man Cybern.* **19**(5), 1060–1072 (September 1989).
16. C. C. Tappert, Dehooking procedure for handwriting on a tablet, *IBM Tech. Disclosure Bull.* **27**(5), 2995–2998 (October 1984).
17. J. R. Ward and T. Kuklinski, A model for variability in handprinting with implications for the design of handwriting character recognition systems, *IEEE Trans. Syst. Man Cybern.* **18**(3), 438–451 (May/June 1988).
18. J. Kim, Baseline drift correction of handwritten text, *IBM Tech. Disclosure Bull.* **25**(10), 5111–5114 (March 1983).
19. J. D. Wilson, The recognition of handwritten words using a line-follower, *Proc. Inst. Radio Electron. Engrs Australia* **31**(10), 339–344 (October 1970).
20. W. Guerfali, Editeur de Texte Manuscrit, Mémoire de Maîtrise, Ecole Polytechnique de Montréal, December (1990).
21. R. M. Bozinovic and S. N. Shriari, Off-line cursive script word recognition, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-11**(1), 68–83 (January 1989).
22. F. J. Maarse and A. J. W. M. Thomassen, Produced and perceived writing slant: difference between up and down strokes, *ACTA Psychologica: Motor Aspect of Handwriting* 131–147 (1983).

About the Author—WACEF GUERFALI received B.Eng. and M.Sc.A. degrees in electrical engineering from the École Polytechnique de Montréal in 1988 and 1990, respectively. He joined the Scribens Laboratory staff at the École Polytechnique de Montréal in 1988, where he is doing a Ph.D. degree in handwriting recognition.

About the Author—RÉJEAN PLAMONDON received a B.Sc. degree in physics, and M.Sc.A. and Ph.D. degrees in electrical engineering from University Laval, Québec, Canada, in 1973, 1975 and 1978, respectively. In 1978, he joined the staff of the École Polytechnique, University of Montréal, where he is currently a full professor. His research interests are concerned with the computer applications of handwriting: biomechanical models, neural and motor aspects of handwriting, character recognition, signature verification, signal analysis and processing, computer-aided design via handwriting, forensic sciences, software engineering and artificial intelligence. He is the founder and director of the Laboratoire Scribens at École Polytechnique de Montréal, a research group dedicated exclusively to study of these topics. An active member of several professional societies and a senior member of the IEEE, Professor Plamondon is a member of the board of the international society, chairman of the technical committee, TC-11, of the International Association for Pattern Recognition (IAPR), on text processing applications. He is president of the Canadian Information Processing and Pattern Recognition Society and a governor of the IAPR. In 1989–90, he was a fellow of The Netherlands Institute for Advanced Study in Wassenaar. He is the author or coauthor of numerous publications and technical reports. He has coedited three books: *Computer Recognition and Human Production of Handwriting* (1989), *Computer Processing of Handwriting* (1990) and *Pattern Recognition: Architectures, Algorithms and Applications* (1991). He has also published a children's book, a novel and a book of poetry.