The Segmentation of Cursive Handwriting: An Approach Based on Off-Line Recovery of the Motor-Temporal Information

 Réjean Plamondon, Senior Member, IEEE, and Claudio M. Privitera

Abstract—This paper presents a segmentation method that partly mimics the cognitive-behavioral process used by human subjects to recover motor-temporal information from the image of a handwritten word. The approach does not exploit any thinning or skeletonization procedure, but rather a different type of information is manipulated concerning the curvature function of the word contour.

In this way, it is possible to detect the parts of the image where the original odometric information is lost or ambiguous (such as, for example, at an intersection of the handwritten lines) and interpret them to finally recover a part of the original temporal information.

The algorithm scans the word, following the natural course of the line, and attempts to reproduce the same movement as executed by the writer during the generation of the word. It segments the cursive trace where the contour shows the slowdown of the original movement (corresponding to the maximum curvature points of the curve). At the end of the scanning process, a temporal sequence of motor strokes is obtained which plausibly composed the original intended movement.

Index Terms—Curvature, handwriting segmentation, off-line motor analysis, stroke recovery.

I. INTRODUCTION

The generation of a graphic shape on a writing surface during handwriting or drawing is the result of a complex motor planning process starting from an input allographic representation of the shape and then producing a partially overlapped sequence of primitive movements (called motor strokes) of the hand-pentip system that traces that shape [1], [2]. The sequence of motor strokes can be described by a vectorial sum of velocity vectors, each one having its module described by a delta lognormal law [3]. This results in a correspondence between the minimum of the curvilinear velocity and the maximum of the angular velocity [4], [5].

Manuscript received January 28, 1996; revised September 23, 1997. This work was supported in part by NSERC Canada under Grant OGP00915, and FCAR Québec under Grant ER-1220F. The work of R. Plamondon was supported by NSERC Canada under Grant OGP00915 and by the a grant from IRIS Project IC-4, Center of Excellence Network. The work of C. M. Privitera was supported by MURST-Italy Piano Nazionale di Ricerca Tecnologie per la Bioelettronica, and by the International Computer Science Institute–Berkeley. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jeffrey J. Rodriguez.

R. Plamondon is with the Laboratoire Scribens, Département de Génie Électrique et de Génie Informatique, École Polytechnique de Montréal, Succursale Centre-Ville, Montréal, P.Q., Canada H3C 3A7 (e-mail: rejean.plamondon@polymtl.ca).

C. M. Privitera is with the Neurology and Telerobotics Units, University of California, Berkeley, CA 94720 USA (e-mail: claudio@milo.berkeley.edu).

Publisher Item Identifier S 1057-7149(99)00221-3.

Fig. 1. Correspondence between maxima in the angular velocity and minima in the curvilinear velocity. (a) Scanned image of a handwritten word. (b) Cartesian coordinates $x(t)$, $y(t)$ as collected with a digitizer. (c) magnitude of the pentip velocity as a function of time. (d) Trajectory curvature as a function of time. Curvature peaks generally correspond to velocity minima.

as reported by many (see, for example, [6] and [7]). Fig. 1 shows this characteristic: Fig. 1(a) is a scanned image of a handwritten word and Fig. 1(b) represents the corresponding Cartesian coordinates $x(t)$, $y(t)$ as sampled with a digitizer. The plots in Fig. 1(c) and (d) correspond respectively to the magnitude of the pentip velocity and the trajectory curvature as a function of time: curvature peaks generally correspond to velocity minima.

This general motor aspect of handwriting also seems to play a key role in the complementary behavior of the generation, that is, the perception of an image representing a handmade message. In fact, the human visual system seems to exploit, in a reverse operation, the same mechanism taking place during generation: the image is analyzed in order to detect the corresponding sequence of primitive motor components,
which have then to be interpreted on the basis of existing
general knowledge about motor control [8], [9]. Finally, the
syntactical class of the word (or drawing) is determined.

On-line handwriting recognition methods represent this
mechanism very well: these algorithms deal with the analysis of
an $x$-$y$ coordinate sequence of the stylus position as
a function of time [Fig. 1(b)], where parameters such as
pressure, velocity [Fig. 1(c)] and acceleration might be readily
available as well. At the end of the acquisition process, the
course of the handwriting line is accurately defined by a
time function which can be segmented into primitive line
pieces or more or less accurately representing the motor strokes
[10]. The corresponding temporal order reconstruction is
then directly obtained by means of the acquired kinematic
information.

The off-line case represents a larger set of problems because
the image of a word does not allow any direct access to
temporal and kinematic information [Fig. 1(a)]. Here, the
preprocessing level, such as the segmentation of a word into
the corresponding writing units, becomes a complex task to
be defined and implemented.

Many methods have been proposed so far that try to segment
a word into basic principal units taking into consideration
general characteristics of the word such as, for example, the
density of pattern pixels along a horizontal projection: in this
case, the areas with low density would correspond to less
meaningful parts of the word (such as the ligature between
two different numerals or characters) where the segmentation
can finally take place (see, for instance, [11], [12] and [13]). A
similar slant-independent approach has been proposed in [14]
where the word is segmented on the basis of some general
characteristics of local extremas along the $y$-coordinate of the
upper contour of the word. These methods seem to be very
efficient but they are all characterized by a common drawback:
the trace can be segmented at points where no original motor
segmentation took place during handwriting generation, so
that part of a word belonging to the same motor unit (for
example, a single stroke or the middle of an allograph) can
be split (and consequently not further recovered) into different
over-dimensioned and nonrecognizable units.

These problems could be overcome by segmenting the
word into elements belonging to a predefined set of heuristic
templates which could identify a set of complex shapes, and
even the set of the letters (a sort of divide and conquer method;
see, for instance, [15] and [16]): in this case the problem of
getting meaningless units is avoided but the segmentation
itself becomes a sort of wholistic interpretation where some
hazardous inferences about the original pentip movement are
necessarily made.

On the other hand, if what we want is to recover all the
original kinematic information of the word, nothing during the
segmentation process has to be approximated, so that, finally,
the recognition process will operate only when the odometric
development of the word has been understood. This approach
might be very promising not only in the light of the cognitive
processes involved, but also because it provides a natural
possibility for filling the gap that exists between on-line and
off-line methods.

In the literature, a few models have already been proposed
in this direction, most of them involving a skeletonization or
a thinning [17] of the original image: in [18], for example,
the original image is transformed into a chain graph by means
of MAT transformation, and finally the temporal course of the
word is recovered using writing rules. In [19], a similar mecha-
nism has been proposed to separate text from overlapping
nontextual contours. In both cases, however, the skeletoniza-
tion of the word can obstruct the natural interpretation of the
visual information because very often it can destroy the word
original shape information [20]. Generally speaking, thinning
processes are classically considered ambiguous and unclear,
since key features such as branch points could be distorted and
because the algorithms are very sensitive to noise [22].

In [23], an interesting approach has been defined where any
thinning process takes place: a sort of taxonomy is described of
the local, regional and global temporal clues about the ink trace
left by the writing device that are capable of driving a correct
off-line recognition recovery of temporal information. The
efficiency of the algorithm comes with a high computational
cost which could dramatically slow down the whole image
processing and consequently the recognition of the word.

The general idea of the original method presented in this
paper is to define a segmentation system based on a contour
tracing of the image which partly mimics the cognitive-
behavioral process used by human subjects to recover motor-
temporal information from the image of the handwritten word.
The word segments are located on the maximum curvature
points of the contour, just as in some on-line methods based
on the detection of the minima on the curvilinear velocity
[see, for instance, [24], and also Fig. 1(c) and (d)] corre-
ponding to a maximum on the angular velocity. As such,
the segmentation reflects the original allographic motor image
of the characters (see, for example, [2] and [25]), introducing
a sort of invariance about subjective writing modalities such
as slant orientation and size of the word, finally leading to an
interesting motor plausibility of the resulting stroke sequence.

Analyzing the contour curvature function, it is also possible
to detect and interpret the ambiguity points of the word (for
example, strokes accidentally touching each other or matching
with nontextual lines) at low computational cost: in this way,
a word can be scanned by following the natural course of the
contour and attempting to repeat the same movement as
executed by the writer during the generation of the word.

At each maximum curvature point of the contour, the line
is segmented and reconstructed by a specific interpolation
procedure.

In the following section, we describe the basic elements of
the system: the image adjustment process, shape analysis and
the ambiguity zone detection and interpretation of the word
image (based on the evaluation of the contour curvature). In
Section III the general reconstruction process of the word is
presented, and, finally, in Section IV the performance of the
system is discussed.

1 Shape analysis algorithms are classically gathered into two different
groups: thin line methods based on thinning and full set methods based on
contour tracing of the image (see also [21]).
II. THE ALGORITHM IN DETAIL

The algorithm is principally structured in three different processing levels which manipulate the image of a handwritten word acquired by means of a scanner (with 300 dpi of resolution) and finally produce the list of reconstructed strokes in the right sequence. These reconstructed strokes represent the smooth, continuous interpolation of the ink trace left on the writing surface.

The first level of the system, image adjustment (IA), binaries the image of a word into a digital picture by means of a threshold process and produces information about the distance of each pixel of the picture from the uniform background. At the second level, shape analysis (SA), a global analysis of the picture is performed using cognitive information to detect the regions of the word characterized by a maximum value of curvature. The SA level also singles out the points that presumably correspond to the beginning of a component (the point where the writing instrument touches the surface in order to begin the writing process [28]) and finally locates and interprets the ambiguous parts of the word. This interpretation is necessary for the last level, the segmenting and reconstruction (SR) level, where the word is scanned from left to right following the continuous course of the handwritten line and consequently segmenting it into the corresponding sequence of motor strokes.

A. Image Adjustment

Although document scanning devices usually already have a built-in filter level, the gray-scale image produced at the end of the acquisition process can still be characterized by the presence of noise, like, for example, stains spread across the surface of the background or imperfections of the ink traces. In order to remove the occasional noise of the image and, consequently, to obtain a good and smooth resolution of the handwritten line, a Gaussian convolution is applied. After the convolution, a thresholding process transforms the filtered gray-scale image into a binary picture.

The binary image coming from the filtering and thresholding stage is now ready for the so-called image adjustment. At this level, the image needs to be transformed into a form that is easier for the subsequent computational levels to manipulate: this transformation is often undertaken by a thinning process, that is, the extraction of the skeleton from the course of the cursive line. But such procedure can be very complex and can involve the partial destruction of the intrinsic motor-temporal content of the word which has to be kept and subsequently interpreted [20].

In an attempt to preserve all the necessary information, our approach is based on a distance transformation of the original picture which leaves the original spatial distribution of word pixels intact. The transformation simply associates each black word pixel \((i, j)\) of the initial picture with a nonzero number \(\chi_{i,j}\) representing the distance between that pixel and the nearest white background pixel. In this manner, the contour of the handwritten shape is defined by the sequence of the black pixels with a distance equal to 1 from the background (see Fig. 2; hereafter, we refer to these pixels as 1-pixels). Moreover, the information about the skeleton is intrinsically present in the result of the transformation since it corresponds roughly to the continuous sequence of the interior-most pixels (the sequence of pixels with maximum distance are referred to as crest); any decision about the effective course of the skeleton is not inferred at this level, postponing the problem to the last level, SR.

Several techniques do exist in the literature for distance transformation (DT), in digital images (see, for example, [29], [30]): the general effort is directed at finding algorithms that provide the best approximation of the Euclidean metric at the least computational cost. These algorithms are classically based on an iterative propagation of local approximations between neighboring pixels where different neighborhood conformations result in different levels of approximation [29].

Since a very accurate Euclidean approximation is not important in our model (since we are simply interested in detecting the sequence of 1-pixels defining the handwritten word border and in having only a rough representation of the crest; details will be provided in the following sections), we favored faster and simpler algorithms. A neighborhood defined by the cityblock distance seems to be an appropriate solution in our context: the corresponding distance transformation algorithm is the fastest to execute, [29], and the resulting distance approximation meets all the system requirements. The cityblock distance is a regular metric: for two given points \(P = (P_1, Q_1)\), \(Q = (P_2, Q_2)\), the distance is defined by the following expression: \(d_c(P, Q) = |P_1 - P_2| + |Q_1 - Q_2|\). The conversion process can be defined by using the parallel algorithm presented in [31] which transforms each binary pixel of the input bitmap into the element of a new array, which in turn represents the corresponding distance from the background.

The procedure is described as follows. At the beginning, each element of the array \(\chi_{i,j}\) is equal to the corresponding binary value of the pixel on the initial bitmap (that is, zero for the nonfeature background pixels of the image and one for the feature foreground pixels representing the word). For
every iteration, the value of $\chi_{i,j}$ is transformed according to:
\begin{equation}
\chi_{i,j}(t+1) = \chi_{i,j}(0) + \min_A \chi_{x,y}(t)
\end{equation}

where $A = \{(x,y): d_c((x,y), (i,j)) \leq 1\}$ and $(x,y)$ is a generic pixel on the digital binary map. In this way, the nonfeature pixels keep their initial zero value for the duration of the operation since the minimum is zero in every iteration.

A contour pixel also keeps its initial unity value for the same reason: the corresponding neighborhood intuitively includes at least one background zero pixel. All the interior pixel distances become two at the first iteration and the ones close to at least one contour pixel will never change again. The more interior pixels become three at the second iteration, and the same process is repeated until no pixel value changes, i.e., the number of iterations is proportional to the largest distance in the image which corresponds to the width of the strokes. The final state necessarily represents the city block distance array.

However, the process of filtering, and consequently of distance computation, can generate some ambiguous possibilities along the course of a stroke. Especially for the extreme points of a stroke, when, for example, the writing device is taken off the writing surface, or at the beginning of a word, where the width could be characterized by an unclear grouping of boundary pixels (\(\chi_{i,j} = 1\)) [Fig. 2(b)] and consequently the natural interpretation of the course boundary can be ambiguous.

Another possibility is shown in Fig. 2(c): in this case there are two boundaries touching each other (two 1-pixels belonging to two different strokes are in contact).

In the preliminary version of the algorithm [32], this problem was resolved by means an additional filtering of the picture which deleted and corrected all trivial boundary pixels. This operation not only involved an increase in the computational cost, but also generated some contour alterations which could produce some errors. In the current version, these drawbacks are directly managed at the SR level by means of a new mechanism of the contour scanning process which simply tries to follow the pixels of the contour taking into account the location of the background; in this way, pixels belonging to different contours will be processed in a different manner. The mechanism is explained in Section III.

B. Shape Analysis

The array produced by the distance computation level is a sort of isoaltitude map having the writing surface as a zero reference value. This map implicitly provides many cue data concerning the word, like, for example, the boundary conformation (the points at altitude 1) and the rough course of the crest. At the shape analysis level, the map is analyzed in order to evaluate the contour curvature function, and consequently the parts of the handwritten line with maximum curvature are deduced.

The curvature of the stroke contour plays a key role for two fundamental reasons: first, as already mentioned in the introductory section, the maximum curvature corresponds to the overlap between two consecutive motor strokes so that segmentation based on this criterion reflects the original intention of the writer without oversegmenting the whole word [3]; second, the curvature function provides decisive factors for the detection and interpretation of the ambiguous parts of the ink trace.

1) Contour Curvature Evaluation and Maximum Curvature Points Extraction: In the real Euclidean plane, the expression of the curvature function $C(x,y)$ of a two-dimensional (2-D) contour is defined as the rate of change of slope $\partial \varphi$ as a function of the arc length $\partial s$, $C(x,y) = \frac{\partial \varphi}{\partial s}$, (as shown in Fig. 3) and can be expressed in terms of derivatives of the $x$ and $y$ coordinates by means of the following well known equation:
\begin{equation}
C(x,y) = \frac{\frac{\partial^2 y}{\partial x^2}}{\left(1 + \left(\frac{\partial y}{\partial x}\right)^2\right)^{3/2}}.
\end{equation}

Dealing with digital curves, such as handwritten traces, where the contour of the strokes is represented by a finite and discrete path of 1-pixels arranged like the black squares on a chessboard, the discrete interpretation of the above curvature equation is not straightforward. In fact, simply replacing the derivatives by differences, and taking into account that the discrete approximation of the Euclidean distance between two consecutive pixels is zero or one, leads to a computed digital curvature in multiples of 45°. Moreover, (2) represents only a local definition of curvature, whereas, in order to detect more general properties of the handwritten line such as good continuation, a more global evaluation of the curvature is needed. In this way the general trend of original pentip movement can be captured.

A possible solution might be to evaluate the curvature function of the digital curve by simply smoothing (and then differentiating) the curve with a 2-D filter such as a Gaussian filter [33], [34]. Another possibility might be to approximate the local curvature taking into consideration a specific and sufficiently large region of support. The magnitude of the region of support obviously plays a key role in the balance between noise sensitivity and resolution of the curvature: too large a region will hide small changes in line slope, whereas a small region will generate very noisy curvatures. A general solution to the above-mentioned tradeoff is to take into account a set of curvature estimates regarding all the points belonging to region of support and then rescale the size of this region on the basis of local information about the curve [35], [36].
In Fig. 4, for example, a generic piece of a digital curve is shown where $p_i$ represents the point where the curvature has to be evaluated and where the sequence of points $p_{i-m} \ldots p_{i-2} \ldots p_i \ldots p_{i+m}$ represents a prefixed range of region of support. For each contour pixel, a first curvature estimation $S_{ik}$ (or measure of significance [37]) is represented by the distance of point $p_i$ to the cord joining a pair of opposite points $p_{i-k}$ and $p_{i+k}$ within the region of support (see Fig. 4). The region of support is then adaptively rescaled on the basis of this measure of significance by an iterative process starting with $k = 1$ and the first point $p_{m^*}$ is determined such that $S_{m^*} > S_{k=1}$; in this manner, the ascending part of the curve (see Fig. 4, from point $p_{m^*}$ to point $p_i$) is discarded in the final rescaled region of support since it represents a different change in line slope. In other words, two consecutive curvature maxima will be considered separately by the adaptive rescaling process.

Finally, the curvature function of the point $p_i$ can be represented by the following average computed over all the curvature estimations belonging to the rescaled region (see also [38]):

$$C(p_i) = \frac{1}{m^* - k_0 + 1} \sum_{k=k_0}^{m^*} S_{ik}$$  (3)

where $k_0$ can be more than 1 in order to discard the points closest to $p_i$ which represent the worse curvature approximation.

The result of the above mentioned process as applied to an image displaying a handwritten word is shown in Fig. 5: in particular, only the 1-pixels belonging to the contour are outlined in the resulting image and the $X$ signs along the contour represent the local maxima of the evaluated curvature function. The maxima of the curvature function (or dominant points) are determined on the basis of local information: in particular, a point $p_i$ is a local maximum if, $\forall j$ belonging to the rescaled region of support ($k_0 \leq j \leq m^*$), the condition $C(p_i) \geq C(p_j)$ holds. Small local maxima are simply discarded with a threshold process.

It is worth noting that the concentration of maximum curvature points is higher in those parts of the word where two or more strokes interfere with each other. This high concentration indicates the presence of ambiguity parts in the word, and will prove strategic for a fast and efficient procedure for detection of the interference zones of the word.

2) Ambiguity Zone Detection and Interpretation: The maximum curvature points of the handwritten word contour extracted during the previous process represent important key references for detecting those parts of the handwritten word where the odometric information is ambiguous and any trivial inference about the original pentip movement cannot be made.

In fact, as shown in Fig. 5(b), some of the points selected by the previous level do not correspond to maximum curvature regions of the underlying strokes. A taxonomy of detectable points is presented in Fig. 6: apart from the points located in a high curvature trajectory of the ink trace [see Fig. 6(a)], we can label points probably referring to the first or last contact between the writing device and the surface of the paper or to very narrow traces of the generating movement [the End points; see Fig. 6(b)]; points corresponding to the regions of high concavity taking shape when two strokes cross each other [Fig. 6(c)] or when two strokes merge into a single stroke, for example, when the second turns back, partially...
covering the previous one [Fig. 6(d)]; and, finally, points due
to accidental contact between strokes belonging to different
characters [Fig. 6(e)].

Though the taxonomy shown in Fig. 6 defines five different
handwriting events, it can be structurally restricted to only two
different classes, which are parts of words where segmentation
should be applied (end points and maximum curvature points)
and parts of words where the ambiguity in the evolving
strokes (crossing strokes, merging strokes and discontinuity
strokes) should be resolved using an intelligent interpretation
procedure. In this sense, all the ambiguity cases, Fig. 6(c),
(d), and (e), are analyzed using the same general procedure
and discrimination among them is not necessary: first the
ambiguity zone is detected (the type of ambiguity is not
relevant in this context) and then, on the basis of general
criteria about handwriting movement (Gestalt’s assumptions,
for example), all the strokes involved are analyzed in order to
finally recover the original motion and to associate strokes (if
any) corresponding to the same original handwriting stroke.

In order to detect interference zones on the digital image,
the algorithm scans the image with a moving window of a
pre-fixed initial size from the top left side of the image (see
Fig. 7) and, each time that the border window encloses a zone
of dominant point concentration, a series of checking processes
take place.

The first operation involves evaluating the connectivity of
the part of the image defined by the window border, and the
second is to compute the number of handwritten lines crossing
through the window border. If the space representing the
handwritten line inside the surface of the window is connected
(see Fig. 8), and the strokes cross the border of the window
more than twice (for example, in Fig. 8(a)), the strokes cross
the window contour four times), this means that the window
is effectively covering an interference zone of the handwritten
word. Finally, the window dimensions are scaled in order to
improve the representation of the interference zones detected
by that window without missing any information about the
conformation of the handwritten lines in that area.

Let us represent the perimeter of a window \( \square \) with a
function \( P(\square) \) defined by a minimum \( (P_{\text{min}}) \) and a maximum
\( (P_{\text{max}}) \) value. In general, the maximum dimension (which is
the only input parameter needed for this process, since the
minimum value could correspond just to the dimension of the
single pixel) cannot be too large because some false crossing
stroke would be detected. By contrast, if the maximum dimen-
sion is less than the section of the handwritten line, the window
would not be able to circumscribe the interference zone. The
exploitation of a general heuristic function of the height of
the word has proved sufficient for a good performance of the
whole algorithm: specifically, the maximum side \( P_{\text{max}} \) of the
window is set to \( \frac{\text{Height}}{6} \) where Height represents the height
of the handwritten shape easily determinable by a vertical
projection of the density of pattern pixels.

Let \( R \) be a Boolean function representing the above men-
tioned checking process so that \( R(\square) \) is true if the window
\( \square \) circumscribes an interference area (the internal area is con-
ected and the window border is crossed three times or more
by the handwritten line) and let \( M(\square) = M_{\text{max}}1 \times M_{\text{max}2} \times \cdots \)
be the set of dominant points circumscribed by the window.
Finally, let \( D = \{ \square_1, \square_2, \cdots \} \) be the set of all the win-
dows representing the interference zones detected during the
scanning process at a certain point.

The following steps represent the general procedure em-
ployed for interference zone detection and window size rescal-
ging:

1) move the window \( \square \) (with the maximum
dimension) as shown in Fig. 7;

\[
\begin{align*}
&\text{if } R(\square) \text{ then } \\
&\text{2) keeping the top-left vertex of the} \\
&\text{window constant (see Fig. 6(b)):} \\
&\text{for } 1 \leq P(\square) = P_{\text{min}} \text{ to } P_{\text{max}} \\
&\text{if } 2 \text{ } R(\square) \\
&\text{3) } \exists \square_i \in D : M(\square) \cap M(\square_i) \neq \{ \} \\
&\text{then add } \square \text{ to } D \\
&\text{else} \\
&\text{4) } \exists \square_i \in D : M(\square) \supseteq M(\square_i) \\
&\text{then } \\
&\text{5) } \#M(\square) > \#M(\square_i) \text{ or } P(\square) < P(\square_i) \\
&\text{then discard } \square \text{ and keep } \square \\
&\text{else discard } \square \\
&\text{endif} \\
&\text{else discard } \square \\
&\text{endif} \\
&\text{endif} \\
&\text{endfor}
\end{align*}
\]
In other words, the process of window rescaling is performed by an iterative procedure that first detects the interference area (Step 1) and then tries to locate the smallest window in that area (Step 2). In step $f_2$, a new interference area is detected; in step $f_3$, two different windows have some dominant points in common and the selection of the window is determined in terms of both the perimeter and the surrounded dominant points in order to favor the smallest window with the most surrounded dominant points.

An example of the final result of this interference detection algorithm is shown in Fig. 9: as can be noted, all the parts of the word containing an overlap between different strokes are detected. Each square represents the final dimension of the windows at the end of the algorithmic process.

It is worth noting that this general approach to handwritten image analysis is consistent with the cognitive behavior of the visual system: in particular, human visual system seems to have a strong specificity to the points of a picture where the curvature values of the boundary are high, so that a complex input shape is initially analyzed on the basis of these maximum concavity regions [39].

In the following section, each stroke arriving at or departing from an interference window is characterized in terms of its direction, bend value and section in order to finally pair strokes belonging to the same movement and to recover the original pentip trajectory.

3) Recovery of Odometric Information in the Interference Zones: This part of the system is based on Gestalt parameters [40] that seem to be involved in the human visual processing of handmade graphic forms. These parameters are 1) the section of the stroke (strokes belonging to the same primitive movement have to have sections of approximately the same width); 2) the good continuation rule (the handwriting process is a smooth movement where the curvature cannot abruptly change, but rather, the writer tends to keep the same value of curvature for each primitive movement); 3) the closure law (the tendency of a handwriting generation system to alternate tracts of opposite curvature during the generation of a sequence of primitive strokes). In the off-line analysis of the ambiguity points of a word, these perceptivo-motor constraints of handwriting generation have to be taken into account in order to arrive at a plausible characterization of the original movement.

For each window representing an interference zone, a specific series of measurements has to be made: the most important among these is a curvature evaluation of the departing and arriving strokes, because, as previously discussed, the curvature represents a key parameter for the correct pairing of strokes belonging to the same movement. In other words, we attempt to evaluate the level of mechanical inertia associated with each possible pair of strokes present in the interference zone.

A set of pairing hypotheses is generated by linking each stroke contour $\gamma_i$ entering the window with the corresponding contour of another stroke $\gamma_j$ of the interference zone (see Fig. 10) as if the first were the continuation of the second. For all the possible different combinations of pairing hypotheses the corresponding curvature function is then computed (see Fig. 10). In this way, the curvature function can provide an important clue as to how the curvature would change if the corresponding pair of stroke contours were to be effectively associated. The curvature function for three pairing hypotheses is shown in Fig. 10: the stroke numbered 0 with the remaining strokes of the window. In particular, each curve displays the mean curvature function as evaluated by taking into consideration the two quasiparallel contours characterizing each single stroke. The peak of the curvature in two of these cases underlines quite an abrupt change which could represent an unnatural movement.

The evaluation of this abrupt change in curvature can be represented by the difference between the maximum and minimum values of the curvature function in the definition interval. In particular, for each pair of points $x_h, x_q$ of the pairing hypotheses curvature function, the following absolute discrete derivative is computed:

$$|\gamma_{i,j}(x_h, x_q)| = \frac{|C_{i,j}(x_h) - C_{i,j}(x_q)|}{|x_h - x_q|}.$$  \hspace{1cm} (4)

The maximum value $\max |\gamma_{i,j}| = \max |\gamma_{i,j}^{\infty}(x_h, x_q)|$ (i.e., the Lipschitz parameter of the function) represents the smoothness level of the pairing hypotheses between the strokes $\gamma_i$ and $\gamma_j$: the smaller this value is the smoother is the function. In other words, curvature functions with abrupt changes (for example a peak such as in a Gaussian with a small variance) or a large oscillation (even if the function is continuous) will be represented by a value of $\max |\gamma_{i,j}|$ larger than the one obtained for a smooth (even if non monotonic) curvature function. In this sense, smoothness is intended to be the opposite property of abrupt change.

The value $\max |\gamma_{i,j}|$ associated with each curvature function is consequently an efficient parameter for the representation of the good continuation paradigm; in the case of the example shown in Fig. 10, the values of $\max |\gamma_1|$ are: 0.41 for Curvevature$(0,1)$, 0.11 for Curvevature$(0,2)$ and finally 0.33 for Curvevature$(0,3)$.

By contrast, while the parameter $\max |\gamma_{i,j}|$ represents a good continuation criterion for the pairing hypotheses, a measure is also needed for making a hypotheses about the level of closure. The bend direction of the two corresponding strokes of a pair hypotheses is used for this purpose. The direction of the curvature $\Sigma_2$ (clockwise or counterclockwise) associated with each stroke $\gamma_i$ of the interference zone is represented by the sign of the $z$ projection (the axis normal to the image plane) of the following vectorial product of the two vector $\bar{B}_i$ (connecting the center of the window and a point along the stroke) and $\bar{M_i}$ (connecting the previous point with another
Finally, the third parameter that has to take part in the interpretation of the interfering stroke is the relationship $S_{i,j}$ between the section of the strokes for each pairing hypotheses; the section of the two strokes $d_i$ and $d_j$ is easily evaluated along the border of the window and the evaluation method is a Gaussian $S_{i,j} = \exp\left(\frac{-(d_i - d_j)^2}{\sigma^2}\right)$, with a fixed variance.

With respect to the cognitive issues raised in this paper, the four parameters, $\max\phi_{i,j}$, $\gamma_i$, $\gamma_j$ and $S_{i,j}$, represent the basic information that is needed for a correct and plausible interpretation of the interference zone. All the above listed parameters are condensed in the following global interpretation function $\Omega_{i,j}$ which practically assigns a score for each pairing hypotheses of the two strokes $i$ and $j$:

$$\Omega_{i,j} = \begin{cases} \max\phi_{i,j}(\alpha - S_{i,j})\gamma_i, & \text{if } \pm(\gamma_i) \neq \pm(\gamma_j) \\ \max\phi_{i,j}(\alpha - S_{i,j}), & \text{otherwise} \end{cases}$$

where $\alpha \simeq 1$ and $0 < \eta < 1$ is an attenuation factor that improves the evaluation of that hypotheses when the corresponding bend directions are opposite ($\pm(\gamma_i) \neq \pm(\gamma_j)$).

The global interpretation function tends to follow the same indication as the curvature function: small values have to be interpreted such that there is a good association probability between the two corresponding strokes. The final interpretation of the interference zone takes place by associating every other stroke of the interference zone so that the sum of the score is optimum over all the possible combinations.

Some typical results of the algorithm are shown in Fig. 12. The double arrows represent the paired stroke, whereas the single arrow identifies a stroke that ends in the interference zone: this happens, for instance, in the first “c” of the word shown in Fig. 12(a), where the interference zone is composed of three strokes and of necessity one of these has to finish there. Similarly in the case of the last “c” [Fig. 12(b)] the model cuts one of the pairing hypotheses (stroke numbers 0 and 2). This is because the minimum score of $\Omega_{1,2} + \Omega_{0,3} = 0.63$ surpasses a pre-fixed maximum threshold (the results discussed in the next section use a threshold equal to 0.5) and consequently only the best hypotheses pair score is taken into consideration ($\Omega_{1,3} = 0.12$) cutting the remaining strokes. It should be noted that this cutting does not lead to an incorrect interpretation of the interference zone since the final hypotheses produced by the process is consistent and plausible from a handwriting point of view. It is worth underscoring the fact that the part of the stroke that has been cut is labeled as an End point so that, for instance, the previous cut would produce two possible points where the handwritten tract could be started or ended. The scribble in Fig. 12(c) represents another key
Fig. 12. Different examples of ambiguity parts interpretation performed by the algorithm: the double arrows represent a pairing between two different parts of the same stroke, whereas the single arrows identify strokes that start or end in the interfering zone.

case for testing the model where the crossing (in the middle of the picture) occurs over an elongated area: in this particular example, the crossing is interpreted by giving priority to the closure law. The example in Fig. 12(d) shows the performance of the algorithm when several retraced strokes are involved.

III. SEGMENTING AND RECONSTRUCTION FOLLOWING THE NATURAL COURSE OF STROKES

At this point, the previous processing levels have generated all the necessary information for the correct scanning of the word so that it is possible at this level to follow the whole course of the word tracing, beginning from the starting point, following the correct direction at discontinuity points or when two strokes overlap.

Detecting the exact points where the writer started generating the word is quite a complex task, taking into consideration the huge range of variations in writing styles. A good estimate can be obtained if a set of possible starting points is first collected and, after that, some cognitive knowledge about the writing process is applied. A good candidate for a starting point can be obtained by detecting the maximum values from the dominant points of the contour (we have labeled these candidates as End points): in our implementation, all the dominant points with a curvature function value greater than 5.0 are all considered as potential End points.

In general, we can determine general properties about the preferred direction exploited by a Latin right-handed writer during writing generation: usually writers tend to begin at the extreme top left of the writing surface continuing downwards, to the right in a counterclockwise fashion [41]. The scanning level SR attempts to detect the real starting point, exploiting the same strategy: initially it starts at the top-left corner of the image and continues down, scanning the picture by means of a series of 45° sloping trajectories. Whenever the scanning procedure detects an End point the algorithm starts from this point, following the stroke course of the corresponding section of the word. Trace following is performed as shown in Fig. 13: at each starting point (the black square in Fig. 13), two different, opposite contour paths, clockwise and counterclockwise (named * and ꞌ) come to be instantiated. For each 1-pixel of the two paths, the next 1-pixel of the contour sequence is selected according to a two-step procedure: the first background pixel (a pixel labeled with a 0; the pixel signed with a grey small circle in Fig. 13 just close to the starting point for example) belonging to the neighborhood of the current point is selected; then, following the direction (ꞌ or ꞌ) associated with that border, the first 1-pixel encountered is finally selected.

In this way, the algorithm follows the course of the boundary of the stroke, pixel by pixel, and stops when a point of maximum curvature is met. At each interruption, the algorithm reconstructs the piece of stroke just covered by interpolating the points with greater distance (the crest) by means of cubic splines (see Fig. 13). Whenever an interference zone is met, the scanning process follows the interpretation calculated in the previous level.

IV. TEST AND DISCUSSIONS

The algorithm has been tuned on several data base and finally tested on a data bases of handwritten word images representing names of international cities written by six different writers where each writer was asked to write six different city names on pieces of A4 paper. Each writer could use several sheets of paper and different cities. The final data base was composed of 200 words.

Each picture was first preprocessed in order to extract the contours and localize the maximum curvature points. Consequently, on the basis of the detected maximum curvature points, the procedure for interference zone detection was applied. Then, considering each interference zone, the interpretation rules were applied in order to detect the best combination from the pairing hypotheses. Finally, each word was segmented following the procedure just mentioned.
right and could serve as a reference [20]. Globally, the system succeeded in interpreting correctly 94% of the ambiguity zones of the word and the percentage of images with whole, original pentip movements correctly recovered was 89% (each word image contained several ambiguity zones where the odometric information was missing). These results are quite satisfactory, especially considering the fact that, in the context of a cursive off-line recognition system, our approach would represent a basic preprocessing stage as well as a knowledge data base for the subsequent prediction-verification steps. In this context, 100% accuracy is not necessary at the preprocessing level since hypotheses raised at higher levels in the recognition process could be used to check for missing or misinterpreted interference zones [42].

Some problems might arise if the handwriting tended to shrink into a restricted area of the writing surface, thus forming an unclear course of the ink line: in this case, the system can move outside its characteristic working capabilities, partly losing the capacity to deal correctly with special cases such as crossing and interfering strokes. Fig. 15 shows one example of this phenomenon (extracted from another data base; see Acknowledgment): after the IA level [Fig. 15(a)], some cues of the word are already lost (for example, the loops of the “c”') and finally the temporal segmentation generates incomplete information about the original word.

In this specific case, other general information could play an important role in the correct interpretation of original movements. The width of the stroke, for example, could indicate hidden loops in some cases. In Fig. 15, the detection of the crossing strokes in the two “c’s could be sufficient to effectively recognize the word seven. Otherwise, this ambiguity could be resolved at a later processing phase, at the recognition level using a fixed dictionary, for example.

The computational cost of the whole system is rather low. Indeed, apart from the initial filter convolution (which can be optimized with the discrete implementation previously mentioned for example), the time processing for each word is a few seconds, depending on the size of the word (on a SPARC 20 system). The whole procedure is also very easy to implement.

Since the final operation taking place is a cubic splines smooth interpolation of the points corresponding roughly to the crest, the results are not very sensitive to boundary noise. We remind readers that the main goal of this preprocessing system is to recover the basic odometric information of the word, such as the sequence of the motor strokes involved and their corresponding general characteristics (such as their tilt, their length and their curvature, which partly mimic the human vision during the reading process). In the light of this consideration, boundary noise was not a critical factor and did not alter the general reconstruction of each motor stroke.

In terms of the curvature function, the proposed algorithm, which is based on a large initial region of support that is then locally rescaled for all the pixels of the contour, according to the procedure defined in Section II-B1, seems to be very robust. In fact, in the experimental tests, none of the errors made by the algorithm was due to missing curvature estimation.
V. CONCLUSION

In the field of off-line handwriting recognition, the possibility of analyzing the word image by means of a temporal unfolding of the handwritten word into a sequence of primitive strokes representing the original movement sequence is a goal which, if achieved, could decidedly improve the performance of the existing reading system. The question does not concern merely cognitive argumentation, as mentioned in a previous section; rather, the key point is that when the analysis and recognition of handwriting are based on a structural transformation of the word (thinning processes, wholistic approaches), then the original information about the word could be irreversibly destroyed. The temporal recovery of the handwritten line, following the pen-up and pen-down movements in the direction of the original handwriting movement, appears to be an interesting possibility for accurately representing the word, which is at the same time easy for the recognition system to manipulate.

In this paper, we have shown how, using some general information regarding the handwriting process, it is possible to correctly segment the image of the word while preserving the original motor-temporal information. The segmentation takes place at the points of maximum curvature of the handwritten line so that it does not depend on subjective writing modalities such as slant orientation and size, but, on the contrary, on the original allographic representation. A handwritten word is thus segmented into basic essential units without any premature structural transformation or interpretation of the whole image. This not only leads to a realistic segmentation scheme (we mentioned some classical drawbacks of existing methods in the introductory section), but it is expected that it will facilitate the recognition phase of the word since this latter process could be based on a reliable recording of the original movement, as is the case for on-line recognition.

This approach is also in line with some basic neurophysiological aspects of perception: for instance, in the speech perception field, the syntactical recognition of speech seems to be linked to the knowledge of motor competencies [43].

The computational complexity of the algorithm is acceptable, since only the contour tracing of the image is taken into account, so that the system assumes an interesting versatility with many aspects of the off-line handwriting analysis. The results of our work prove that it is realistic to think that the combination of a cognitive general processing of a word based on appropriate reference points (the point of maximum curvature) followed by an intelligent scanning of the word can closely approach human handwriting analysis capability.

ACKNOWLEDGMENT

A large part of this work was done when C. M. Privitera visited the Laboratoire Scribens, École Polytechnique de Montréal. The authors thank CENPARMI, Concordia University, for providing one of the databases used for tuning the algorithm.

REFERENCES

PLAMONDON AND PRIVITERA: SEGMENTATION OF CURSIVE HANDWRITING


