Off-Line Loop Investigation for Handwriting Analysis

T. STEINHERZ, D. DOERMANN, E. RIVLIN, and N. INTRATOR

***Abstract*-- Resolution of different types of loops in handwritten script presents a difficult task, and is an important step in many classic word recognition systems, writer modeling, and signature verification. When processing handwritten script, a great deal of ambiguity occurs when strokes overlap, merge or intersect. This paper presents a novel loop modeling and contour-based handwriting analysis that improves loop investigation. We show excellent results on various loop resolution scenarios including axial loop understanding and collapsed loop recovery. We demonstrate our approach for loop investigation on several realistic data-sets of static binary images and compare with the ground truth of the genuine online signal.**

***Index Terms*--Handwriting analysis, Shape, Contours.**

I. INTRODUCTION

SHARING many of the trajectory singularities, loops appear as one of the most dominant features available in cursive handwriting processing ([1]-[5]). In particular, loops are the key to successful off-line-to-online-based word recognition systems, i.e. those mapping a static (bitmap) image to an ordered list of pixel locations along a time axis ([6]-[10]). It is useful to investigate how improved loop detection and recognition can facilitate not only character recognition, but also writer modeling for identification and examination ([11]-[13]), and script or style identification ([14]-[16]). Similarly, many other applications in forensic science, such as signature verification, could benefit from loop analysis ([17]-[21]).

The dominance of loops in these tasks reflects in part on their frequent presence in handwritten cursive words and their parameterizable descriptive nature. J. C. Simon first elucidated the elementary nature of loops and provided an intuitive definition of the types of loops ([22]): “Displacing a pen from left to right in an oscillating movement, with loops, descendants (legs), and ascendants (poles).” Moreover, in the common case of pure cursive handwriting, its continuous nature constrains many ascending and descending strokes in a loop form. Therefore, we consider an extended definition of loops to contain all kinds of uninterrupted enclosures

Digital Object Indentifier 10.1109/TPAMI.2008.68

([23]), including those with invisible 'holes'. Thus, loops can be found in the usual letters like *a,d,e,g,o,p,q,* and in letters like *b,f,h,j,k,l,s,t,y,z*. In most cases, any stroke intersection, excluding delayed strokes, relates to some kind of a loop.

The significance of loops increases because of their parameterizable nature, which enables the transformation of a static loop image into a quantified feature vector. Thus, the loop provides information in a format usable in machine learning algorithms. Given the ground truth for genuine loops provided by the online signal, loop investigation essentially tries to understand the isomorphism between the off-line image and the online signal. Unfortunately, such a transformation is not straightforward ([2]).

Loop investigation has been considered in the context of enhancing off-line handwritten word representation and the reconstruction of the genuine ordered list of strokes. It has been done mostly by using temporal (dynamic) information recovery techniques, such as contour analysis ([24],[25]), gray scale examination ([26],[27]) and path minimization ([28]-[30]). Other methods include thinning / skeletonization ([31]-[39]) and morphological loop investigation ([40]-[42]). This paper improves aspects of former solutions; we detect and resolve the structure of most loops. Our method uses a sophisticated contour analysis we call *The Multipartite Matching Approach*. Our algorithm is beneficial in cases where a complete off-line to online transformation is desired.

This paper has four main sections: Section II introduces the theory of loops; in Sections III and IV, *The Multipartite Matching Approach* and its implementation are demonstrated; Section V provides experimental results. The concluding section provides a final discussion.

II. Loop Theory

A. Definition

A *loop* is a handwritten pattern, made of several strokes formed when the writing instrument returns to a previous location while touching the pad continuously, giving a closed outline with a 'hole' in the center. In this case, a *stroke* is a writing locus between every two consecutive local extremity points of the vertical dimension (*y* axis).

In practice, especially in low resolution images, we require additional information to distinguish a real authentic loop from a pair of two close, connected and partially overlapping strokes. For this purpose we define an authentic loop to be either *real* or *large*, where:

- In a *real loop*, like the one presented in Figure 1-(a), the area is not empty;
- In a *large loop*, like the one shown in Figure 1-(b), the area is empty but the perimeter exceeds a predetermined threshold.

where the *area* is the group of all background pixels inside the 'hole', and the *perimeter* is the collection of foreground pixels surrounding it.

However, Figure 1-(c) shows a shape that satisfies the definition of a loop, but is rejected because it is too small. Patterns like the one illustrated in Figure 1-(d) are considered loops only when sufficiently large because they are rarely actual loops.

B. Classification

Let the *axis* be the main shortest path from the left side of the word to its right side, and let the *tarsi* be the remaining parts – ascenders and descenders. Then, based on J. C. Simon's definition of a loop, we propose classification into two kinds:

Natural loops, like the ones presented in Figure 2, appear on tarsi and hence have a single anchor point where they hang on the axis.

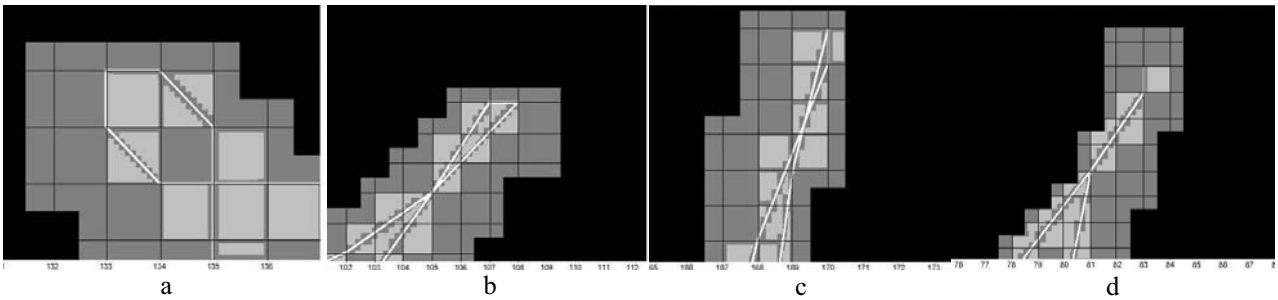


Figure 1: Zoom-in on various patterns of alleged loops: (a) a real loop, where the area is not empty, i.e. there are background pixels inside the 'hole'; (b) a large loop, where the area is empty but the number of pixels on the perimeter is more than a threshold; (c) a small loop, where the area is empty and the number of pixels on the perimeter is less than or equal to a threshold; (d) the "lower limit" of a loop - an extremely thin loop that might be considered a pole because the top-down stroke completely overrides the bottom-up stroke it follows, leaving no theoretic 'hole' in the middle. Light and dark gray squares illustrate the trajectory captured by a digitizer (online – determines the area and perimeter) and the image of scanning the associated inked page (off-line), respectively. Each square represents a single pixel. White lines represent the center mass of the pen tip movement.

Artificial loops, like the ones shown in Figure 3, partition the axis by presenting two interface points in diverse locations. The first interface point, on the left side, is the entrance while the second one is the exit, hence it appears on the right side.

C. Modeling

We have developed two distinctive models that describe the two loop classes.

A *natural loop* (Figure 2) consists of a continuous pair of consecutive strokes that surround an imaginary *natural 'hole'*. All pairs of successive pixels by the temporal order are also 8-neighbors in the image domain. A bottom-up top-down pair of adjacent strokes sketches an ascending loop in a counterclockwise manner. A descending loop is drawn the other way around.

An *artificial loop* (Figure 3) occurs when two sets of consecutive strokes introduce another contact point separate from their concatenation interface, sandwiching a blocked *artificial 'hole'* between them. Either enclosing sets of consecutive strokes could be a natural sub-loop itself. The other alternatives are short poles or simple lines. The contact point could appear at the top or bottom for upper or lower artificial loops, respectively.

An artificial loop is categorized according to its natural sub-loop ('hole') configuration: neither side, only on the left side, only on the right side, both on the left and right sides.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

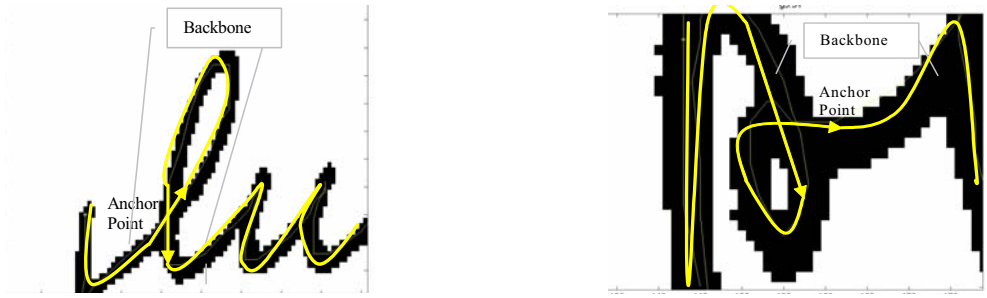


Figure 2: Two natural loops, each made of a single continuous chain of consecutive strokes surrounding a 'hole'. A bottom-up top-down pair of adjacent strokes sketched in a counterclockwise manner forms the ascending loop on the left. The descending loop on the right is drawn the other way around. Both loops have a single anchor point where they hang on the axis.

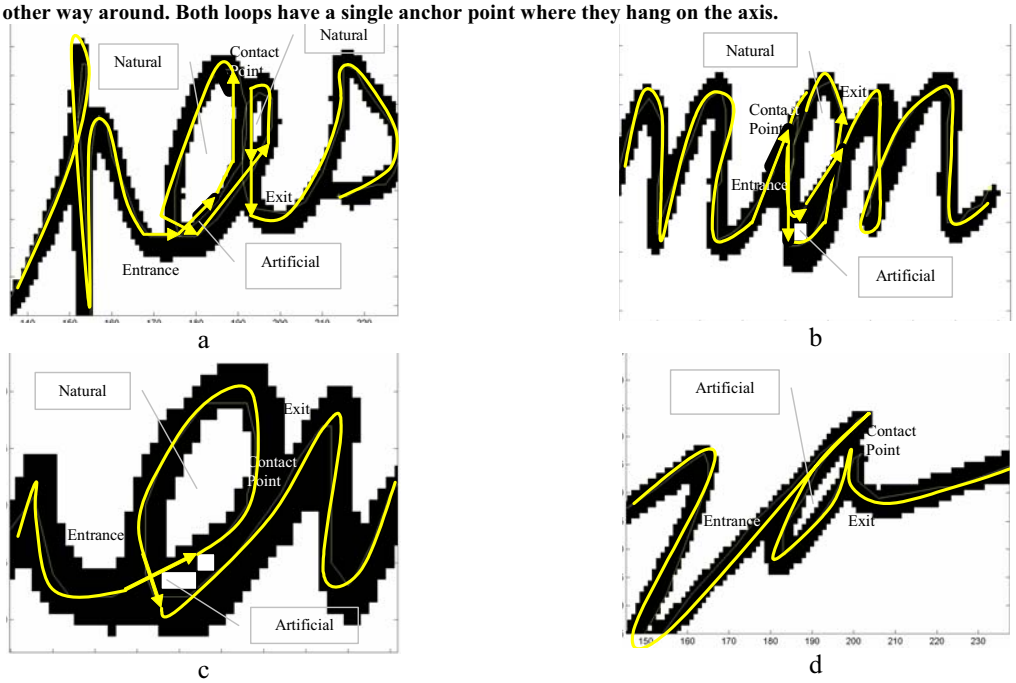


Figure 3: Four artificial loops, each created by two sets of consecutive strokes that introduce a separate contact point away from their concatenation interface, sandwiching a blocked artificial 'hole' between them. The different loops suggest various alternatives of sets of stroke combinations: (a) two natural sub-loops ('holes'); (b) a pole and a natural sub-loop ('hole'); (c) a natural sub-loop ('hole') and a simple line; (d) two poles. In addition, the expected blocked artificial 'hole' in sub-figure (a) is not visible in the off-line representation. All loops partition the axis by presenting two interface points in diverse locations – the entrance and exit, on the left and right sides, respectively.

D. Off-line vs. Online

In static (binary) images the temporal information is implicit and often ambiguous. When off-line processing occurs, visible 'holes' appear as the only evidence that support allegedly genuine loop identification. Nevertheless, the above-mentioned designation clearly displays that both types of loops present 'holes', so no isomorphism occurs between the collection of 'holes' and one of the classes. The designation of axial and on-tarsus loops helps moderately, but would not provide an indicative response for situations where multi-ambiguous 'holes' are encapsulated

within a single 'frame' of an artificial (axial) loop. This happens when a natural sub-loop formulates one of the 'walls' that encloses the complete loop's artificial 'hole'. Furthermore, *superfluous 'holes'* exist, like those in Figure 4-(a)+(b), resulting from noisy pixels or remainders of traced-over natural sub-loops. Therefore, general information on 'hole' inclusion or absence does not satisfactorily complete the categorization task, so the architecture of the presumed loop must be figured for each 'hole' individually.

In addition to uncertainties in the association of some observable 'holes', many others, both natural and artificial, collapse and become hidden in the transition to a static (binary) image format ([40]-[42]). In this case, a blob remains in the original position of the genuine 'hole'. This phenomenon results from blotting or blurring affects common to mechanical writing tools. Figure 4-(c)+(d) provides two examples of collapsed 'holes', a natural hidden loop and a natural hidden sub-loop on the left and right, respectively. Figure 3-(a) shows a collapsed artificial 'hole'.

Handwritten blobs also can be found in intersections of strokes, junctions, and zones where consecutive strokes touch or partially override each other. In fact, previous studies showed that blob width could not distinguish the derivatives of genuine loops from the other byproducts ([25],[41]-[43]). This task requires advanced shape analysis.

In the context of loops, then, one must be able to identify and classify all authentic 'holes' to bridge the gap between off-line and online and recover the topological structure of a loop. Complete identification requires recovery of collapsed 'holes'. Successful classification means distinguishing between natural, artificial and superfluous 'holes' a-posteriori.

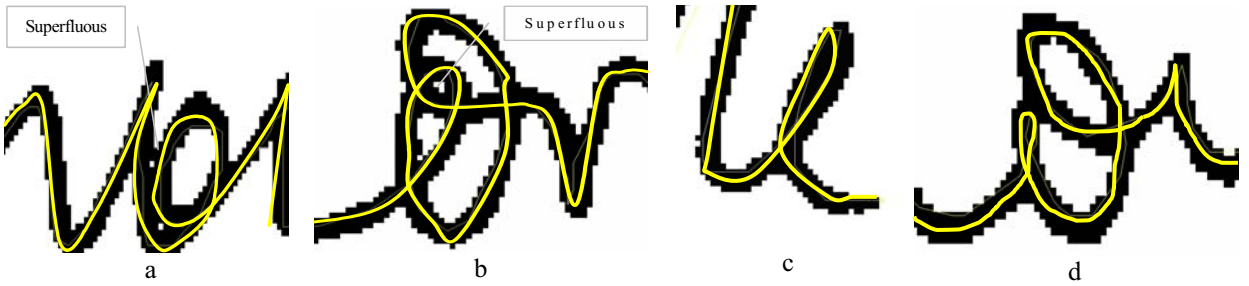


Figure 4: Two examples of superfluous 'holes': (a) the result of noisy pixel(s); (b) traced-over legitimate natural sub-loop(s). Two examples of collapsed 'holes': (c) natural ascending hidden loop; (d) natural ascending hidden sub-loop featuring a left enclosing stroke in an artificial axial loop.

III. The Multipartite Matching Approach

A. Overview

Originally, a single stroke in real time produces two contours on opposite sides, where a *contour part* is an ordered list of the minimum on-the-edge adjacent pixels. Usually a contour piece would be located approximately half a stroke-width away from the exact position of the pen tip. Naturally, some valuable information is lost in the transition to a static image representation. In this case, some contour pieces that cross have disappeared; others become difficult to sort because spatial connectivity is not isomorphic to the genuine temporal order; and the internal manner in which the pixels of a single piece are traversed (either forward or backward) is not properly defined either. Two chains of concatenated matching contour pieces on opposite sides, that were created with a continuous set of strokes and produced a single connectivity component, can be incorrectly represented by a static image. Specifically, the resulting static image presents a deceptively diverse picture of the contour. In this case, one element of the *external contour* exists, a single integrated portion of contour pieces that surrounds the whole body (one segment per connectivity component), and several elements of the *internal contours*, a collection of contour fragments that surround each 'hole'. Nevertheless, each and every visible contour piece derived from the static image is a genuine contour piece produced in real time.

Normally, contour pieces of consecutive strokes also follow one another on the external edge of the resulting static image. One chain of concatenated contour pieces appears on the upper side of the external edge while their complements appear on the lower side.

However, abnormalities/singularities arise around junctions where two strokes intersect and cross each other. In this case, some contour pieces are covered and permanently lost, leaving an inked gap between pairs of consecutive contour parts. The neighbor of a contour piece at the intersection point does not actually follow it. Instead, the consecutive contour part would be found elsewhere along an edge of the resulting static image. Given the continuous nature of cursive handwriting, an intersection implies a close outline, i.e., a loop, so the next pieces of the contour on one side would appear inside the resulting 'hole'. The contact between such allegedly neighboring contour pieces is referred to as a *discontinuity point*.

As a first step toward regaining the separation to two distinguished sides, it is advisable to divide the external contour into upper and lower. Without limiting generality, the external contour begins at the original starting point chosen by the writer (up to an exact location on the perimeter of a loop). Likewise one can presume the authentic final point. The original finishing point partitions the external contour into upper and lower: let the prefix up to the original finishing point constrain the *upper external contour* and let the suffix from this point on describe the *lower external contour*. Pieces surrounding ascenders (descenders) can be further partitioned to left and right around the piece's uppermost (lowermost) local maximum (minimum) point.

Figure 5 presents the external (upper + lower) and internal contour pixels of the word 'flat'.

a b c

Figure 5: The various contour sides of the word 'flat': (a) upper external; (b) lower external; (c) internal.

The Multipartite Matching Approach enables either the association of corresponding opposite-sided contour pieces or the validation of such a presumed matching hypothesis, to bridge/concatenate same-sided consecutive contour pieces across junctions, and to locate lost internal contours.

Both opposite-sided contour piece association and lost internal contour location rely on measurements of mutual distances and shape similarity between contour pieces. Same-sided consecutive contour piece bridging/concatenation across junctions use smoothness in slope/gradient changes and trend.

The Multipartite Matching Approach utilizes a dual representation of contour pieces – pixel-based and section-based. In this case, a *section* is a short straight line that represents the smooth representation for the group of consecutive pixels located in the interval between its starting and ending points. Above this, we develop a multi-layer theory about inner and inter-contour piece relations, derived from both representations in parallel – see the map in Figure 6. The first level presents the basic attributes of an atomic entity – a single pixel or section, respectively. The second and third levels describe local and remote relations between pairs of touching and distant atomic entities, respectively, among which are turning angle and direction, distance, and shape similarity. The fourth level brings several operators that denote existence of association between pairs of atomic entities, possibility of legal concatenation between pairs of atomic entities, and prediction of whether a lost contour fragment occurs between two chains of consecutive atomic

entities of contour pieces.

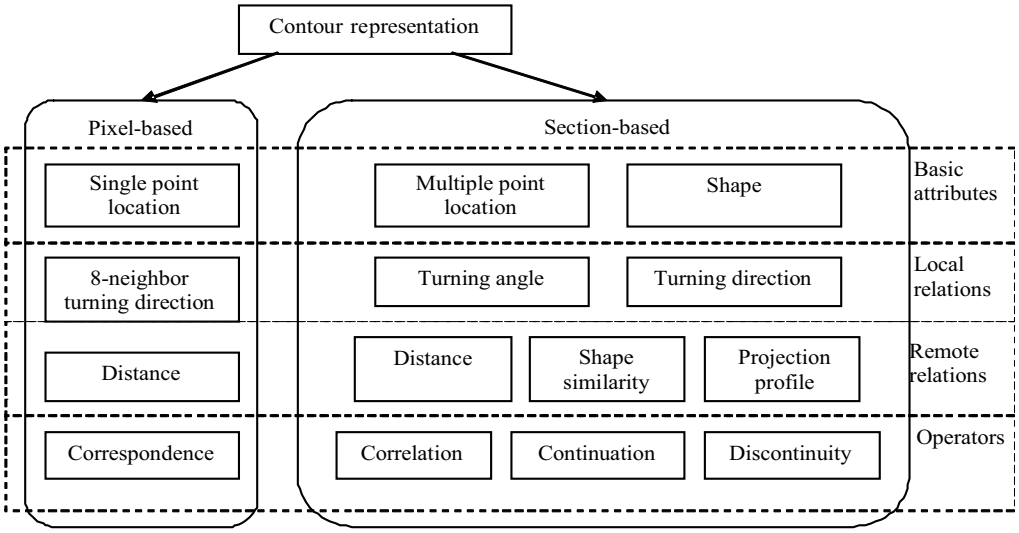


Figure 6: A diagram of the four layers of information derived from the dual representation of contour pieces, pixel-based and section-based, respectively, including basic attributes, local and remote relations, and operators.

In the following sub-sections, we elaborate on the labeling of the various contour parts – upper, lower, left and right, the isomorphism between pixels and sections, and the four layers of information derived from and computed based on contour representations.

One may also refer to other papers that have suggested contour-based methods for various tasks in the document and handwriting processing and recognition fields ([21],[25],[44]).

B. Representation

Given a static image, the external contour is derived by surrounding the word's body segment in a clockwise manner, keeping the neighboring background pixels to the left at all times, in a way that each on-the-edge pixel is visited at least once. In a similar way, each and every internal contour element is the collection of minimal ordered lists of on-the-edge adjacent pixels that surround a 'hole' in the word's image, given each on-the-edge pixel is visited at least once. Natural ascending (sub-)loops are surrounded in a counterclockwise manner. Similarly, natural descending (sub-)loops are surrounded in a clockwise manner. The method of surrounding

artificial 'holes' is not properly defined. Pixel adjacency occurs in accordance with the 8-neighbor rule. The white pixels in Figure 7-(a) represent the external and internal contours of the character 'a', derived from the word 'flat'.

In our view, a complementary functional representation for a piece of contour (other than an ordered list of adjacent pixels) is a set of concatenated sections, where each section is a (short) straight line that begins from and ends at a pixel of the genuine set. This representation appears as a smoothed version, based on the tradeoff of eliminating noise and significant but tiny fluctuations.

Figure 7-(b) provides the equivalent section-based representation to the external and internal pixel-based contours of the character 'a', derived from the word 'flat'.

C. Isomorphism

Section $c = [p_1,p_n]$ – a straight line that connects p_1 and p_n is the smooth representation of an ordered list of neighboring pixels $\{p_1,...,p_n\}$ if, and only if, each one of the replaced pixels is located not more than 1 pixel (or some other predefined constant) away:

$$\|p_i - c\| \le 1 \quad 1 \le i \le n \tag{1}$$

where the Euclidean distance is the metric that measures the distance between a pixel and the straight line representing the section.

The minimal set of concatenated sections $\{c_1,...,c_M\} = \{[p_{1,begin},p_{1,end}],...,[p_{M,begin},p_{M,end}]\}$ provides the isomorphic section-based representation for the pixel-based genuine format of a contour piece $\{p_1,...,p_N\}$, where $p_{1,begin} = p_1$ and $p_{M,end} = p_N$ if, and only if:

$$p_{i,begin} \in \{p_1,\ldots,p_N\},\ p_{i,end} \in \{p_1,\ldots,p_N\},\ p_{i,end} = p_{i+1,begin} \quad 1 \le i \le M \quad \text{and} \tag{2}$$
$$\forall p_j \in \{p_{i,begin},\ldots,p_{i,end}\} \ \|p_j - c_i\| \le 1 \quad 1 \le i \le M$$

See Figure 7-(b) for further illustration.

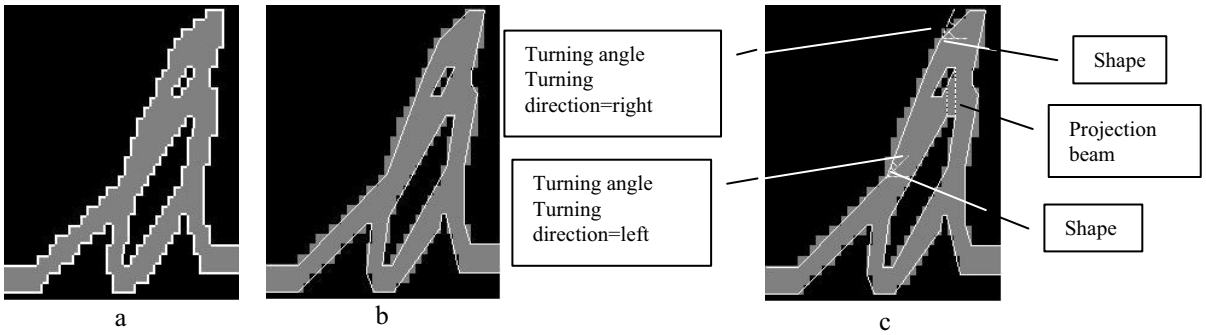


Figure 7: A zoom-in look on the external and internal contours of the character 'a', derived from the word 'flat', by pixel-based and section-based representations in (a) and (b), respectively. (c) An illustration of contour section's shape (its slant angle), turning angle and turning direction with the following one, and beam of the projection coverage.

D. Atomic Entities and Basic Attributes

The atomic entity of the pixel-based format of a contour piece is one point represented by a two-dimensional vector: $p = (x,y)$. Hence the description of a pixel provides a single basic attribute – its location in the image space.

The atomic entity of the section-based format of a contour piece is one short straight line represented by two enclosing pixels: $c = [p_{begin},p_{end}] = [(x_{begin},y_{begin}),(x_{end},y_{end})]$. In this case, the description of a section provides several basic attributes – the location of all of its constituent pixels; and its shape given by the slant angle. See Figure 7-(c) for an illustration of this. Table 1 summarizes the basic attributes of the atomic entities for both the pixel-based and section-based representations.

Property	Pixel-based	Section-based
Location	One point (x, y)	Straight line between two points $[(x_{begin},y_{begin}),(x_{end},y_{end})]$
Shape	-	Slant angle $\alpha = arctan((y_{end} - y_{begin}) / (x_{end} - x_{begin}))$

Table 1: Basic attributes of the atomic entities of contour representation

E. Relations between Atomic Entities

1) Local

A building block of the pixel-based model of a contour piece represents the local relation between two adjacent pixels. Hence the pair of pixels provides a joint attribute – one of eight

possible turning directions in which the preceding pixel points to the one that follows in the ordered list. A building block of the section-based model of a contour piece is the local relation between two concatenated sections (at their interface point). In this case, the pair of sections provides a joint attribute – the turning angle between the former and the subsequent in the set, given by subtraction of the current slant angle from the following one. See Figure 7-(c) for an illustration of this. The turning angle normalizes to the interval [0, Π]. The sign of the turning angle, also referred to as the *turning direction*, is designated positive for left turns and negative for right ones. Table 2 summarizes local relations between pairs of touching pixels and sections, respectively.

Property	Pixel-based	Section-based
Neighbors	Two adjacent pixels p_i, p_{i+1} | $\|x_{i+1}-x_i\| \leq 1$ and $\|y_{i+1}-y_i\| \leq 1$	Preceding and following sections $c_a = [p_{a,begin}, p_{a,end}]$, $c_{a+1} = [p_{a+1,begin}, p_{a+1,end}]$ | $p_{a+1,begin} = p_{a,end}$
Turning angle (absolute value)	-	$\|turning\_angle(c_a, c_{a+1})\| =$ $min(\|\alpha_{a+1} - \alpha_a\|, \|\Pi + \alpha_{a+1} - \alpha_a\|, \|\Pi + \alpha_a - \alpha_{a+1}\|)$
Turning direction (= sign of turning angle)	One of eight turning directions	$turning\_direction(c_a, c_{a+1}) =$ $sign(turning\_angle(c_a, c_{a+1})) =$ -1 or $+1$ for left or right, respectively

Table 2: Local relations between the atomic entities of contour representation

2) Remote

A building block of the pixel-based model of a contour piece is the remote relation between two pixels on opposite sides ({external vs. internal} or {upper vs. lower} or {left vs. right}). Hence the pair of pixels provides a joint attribute – their *mutual Geodesic distance*, defined as the minimum number of body pixels that separate the two body points. Figure 11**Error! Reference source not found.**-(b) illustrates a shortest Geodesic path that serves the distance calculation. Similarly, the distance between a pixel and a piece of contour is given by the minimum distance between the pixel and each one of the pixels on the other piece.

A Breadth First Search (BFS) algorithm calculates the distance matrix between pairs of pixels

on opposite sides. The search environment would be a graph isomorphic to the word's image – inked pixels associated with nodes and 8-neighboring relations represented by edges (see [45]).

A building block of the section-based model of a contour piece is the remote relation between two sections on opposite sides. In this case, the pair of sections provides several joint attributes – their *mutual distance*, which is the minimal distance between a pair of pixels one from each section; their *shape similarity* given by the absolute difference between their slant angles; and their *mutual projection coverage*, which tests the potential of one section to cross the projection beam perpendicular to the other one (Figure 7-(c)), and vice versa.

Table 3 summarizes remote relations between pairs of distant pixels and sections, respectively.

Property	Pixel-based	Section-based
Pairs	Two pixels on opposite sides	Two sections on opposite sides
Distance	$Pixel\_Distance(p_i,p_j) = \lvert min(Geodesic\ path(p_i,p_j))\rvert$ $Pixel\_Distance(p_i,\{q_1,...,q_n\}) = \min_{q_k \in \{q_1,...,q_n\}} (Pixel\_Distance(p_i,q_k))$	$Section\_Distance(c_a,c_b) = \min_{p_i \in \{p_{a,begin},...,p_{a,end}\},\ q_k \in \{q_{b,begin},...,q_{b,end}\}} (Pixel\_Distance(p_i,q_k))$
Similarity	-	$Section\_Similarity(c_a,c_b) = min(\lvert \alpha_b - \alpha_a\rvert, \lvert \Pi + \alpha_b - \alpha_a\rvert, \lvert \Pi + \alpha_a - \alpha_b\rvert)$
Projection Coverage	-	$Projection\_Coverage(c_a,c_b) = 1$ iff $(x'_{b,begin} \leq x'_{a,begin} \leq x'_{b,end}) \vee (x'_{b,begin} \leq x'_{a,end} \leq x'_{b,end})$ where $[x'_{a,begin}, 0] = M * [x_{a,begin}, y_{a,begin}]$ $[x'_{a,end}, 0] = M * [x_{a,end}, y_{a,end}]$ $[x'_{b,begin}, y'_{b,begin}] = M * [x_{b,begin}, y_{b,begin}]$ $[x'_{b,end}, y'_{b,end}] = M * [x_{b,end}, y_{b,end}]$

Table 3: Remote relations between the atomic entities of contour representation

F. Operators on Atomic Entities

1) Correspondence

In our view, two pixels from contour pieces on opposite sides, and that are near to one another, may belong heuristically to the same genuine stroke. So pixels p_i and p_j on opposite contour sides are presumed to be correspondence-based associated if, and only if, their mutual Geodesic

distance does not equal more than the stroke-width:

$$Correspondence\left(p_i, p_j\right) = 1 \text{ iff } Pixel\_Distance\left(p_i, p_j\right) \leq stroke\text{-}width \quad (3)$$

One can propagate the correspondence property to a pixel-piece level as follows:

$$Correspondence\left(p_i, \{q_1, \ldots, q_n\}\right) = 1 \text{ iff } \exists k\ 1 \leq k \leq n \mid Correspondence\left(p_i, q_k\right) = 1 \quad (4)$$

Figure 8-(a) shows all possible pairs of corresponding pieces within the external and internal contours and between the upper and lower sides of the external contour.

2) Correlation

Two sections may heuristically belong to the same genuine stroke if they originate from contour pieces on opposite sides, present a similar shape, are not too far apart, and have positive projection coverage potential. So sections c_a and c_b on opposite contour sides are presumed to be correlation-based associated if, and only if, the absolute difference between their angles is less than or equal to *pi* over four, their mutual distance is less than or equal to twice the stroke-width, and there is at least one pixel on one section that crosses the projection beam of the other:

$$\begin{aligned} &Correlation(c_a, c_b) = 1 \text{ iff } (Section\_Similarity(c_a, c_b) \leq \pi/4) \wedge \\ &(Section\_Distance(c_a, c_b) \leq 2 * stroke\text{-}width) \wedge \\ &((Projection\_Coverage(c_a, c_b) = 1) \vee (Projection\_Coverage(c_b, c_a) = 1)) \end{aligned} \quad (5)$$

One can propagate the correlation property to a section-piece level as follows:

$$Correlation(c_a, \{d_1, \ldots, d_m\}) = 1 \text{ iff } \exists l\ 1 \leq l \leq m \mid Correlation(c_a, d_l) = 1 \quad (6)$$

Figure 8-(b) shows all possible pairs of correlated sections within the external and internal contours and between the upper and lower sides of the external contour.

3) Continuation

The most common assumption about the oscillating hand movement when practicing cursive handwriting is that it acts under an objective to maintain smooth strokes as much as possible in order to lose as little energy as possible (Gestalt's assumptions and parameters in [50]). In this case, one will refrain from sharp turns and avoid switches of the general trend from convex walk

to concave, or vice versa, in the middle of a stroke. Instead, the same turning direction will appear all along the stroke. It seems that psychomotor factors relate to this behavior. This assumption was widely utilized in previous work (see for example [8],[25],[28]).

Based on this paradigm, we have heuristically determined that a newly created section that concatenates two pieces of contour into a single continuous stroke must preserve the same general trend of walk, either convex or concave, all the way from top to bottom, or vice versa, including the prefix of the originating piece of contour and the suffix of the destined one. From a practical standpoint, the turning direction at the interface points surrounding the newly created section must be preserved consistently.

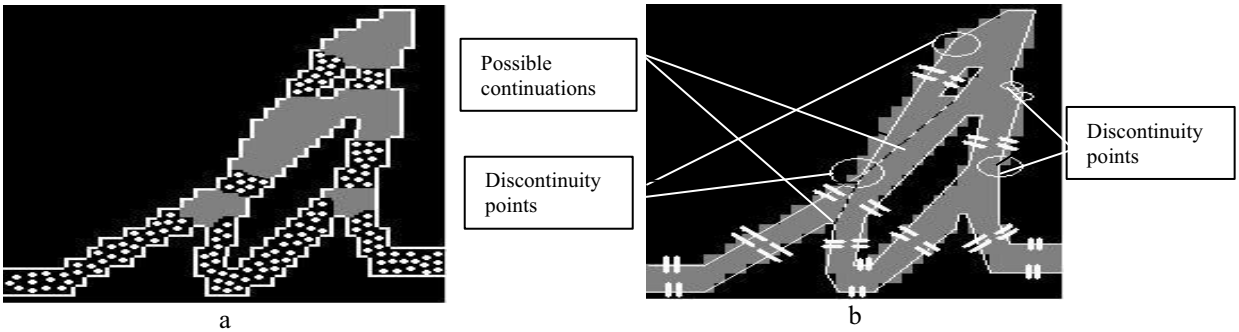


Figure 8: (a) An illustration of all possible pairs of corresponding pieces within the external and internal contours and between the upper and lower sides of the external contour. (b) An illustration of all possible pairs of correlated sections within the external and internal contours and between the upper and lower sides of the external contour (pairs of parallel bars); two possible continuations between a section on the upper external contour and a section on the internal contour, and between a section on the lower external contour and a section on the upper external contour (dashed lines); all possible discontinuity points on the external contour (surrounding circles).

Hence, a newly formed section c_{new} ($=[p_{a,end},p_{b,begin}]$) creates a legitimate continuation (bridge) $< \ldots,c_a,c_{new},c_b,\ldots >$ between two sections c_a ($=[p_{a,begin},p_{a,end}]$) and c_b ($=[p_{b,begin},p_{b,end}]$) on different sides of the contour around the same junction if, and only if, all the related turning directions (between the first and the newly formed sections, between the newly formed and second sections, in front of the first section, behind the second section) are the same, and the newly formed section does not cross or get too close to background pixels including 'holes':

$$Continuation(c_a,c_b) = 1 \text{ iff } (turning\_direction(c_a,c_{new}) = turning\_direction(c_{new},c_b) = turning\_direction(c_{a-1},c_a) = turning\_direction(c_b,c_{b+1})) \wedge (c_{new} \text{ is inside word's body}) \quad (7)$$

Figure 8-(b) presents two possible continuations by newly formed sections that concatenate a

section on the upper external contour and a section on the internal contour of an encapsulated 'hole', and a section on the lower external contour and a section on the upper external contour.

4) Discontinuity

Continuing with the smooth path paradigm, we have heuristically determined that a switch of the turning direction trend far from an extremal point, which means a transition from convex to concave walk, or vice versa, or a sharp turn between two neighboring sections, may indicate the existence of a discontinuity point. The latter is a possible position for a contour split.

The interface between two neighboring sections c_a and c_{a+1} is presumed to be a *discontinuity point* if, and only if, the turning direction is not the same as the turning direction between the first section and its predecessor, given neither the first nor the second section shares an extremal point, or the absolute turning angle is more than or equals to *pi* over two:

$$Discontinuity(c_a, c_{a+1}) = 1 \text{ iff } ((turning\_direction(c_a, c_{a+1}) \neq turning\_direction(c_{a-1}, c_a)) \vee (turning\_angle(c_a, c_{a+1}) > \pi/2)) \wedge (p_{a,begin} \neq extramum\ point) \wedge (p_{a+1,begin} \neq extramum\ point) \quad (8)$$

Figure 8-(b) shows all possible discontinuity points on the external contour.

G. Computed functions on pieces of contour

1) Internal contour recovery

Lacking a visible 'hole', a lost closed outline piece of an internal contour is characterized by the shape of a truncated ellipse with narrow waists - an aperture whose size approximates the stroke-width pixels at the origin, around the location of the genuine junction where the foregoing and backtracking strokes crossed each other. See, for example, the lost loop in Figure 9-(a)+(b).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Figure 9: An illustration of the separation of an external contour fragment surrounding a collapsed 'hole' to left and right sides in (a) and (b), respectively. The graphs of the distance functions between left and right, and vice versa, in (c) top and bottom, respectively.

Two pieces of contour on opposite sides of a blob that draw together an outline of an imaginary truncated ellipse, may heuristically indicate the existence of a lost piece of an internal contour surrounding a genuine 'hole'.

Let $<D_1,...,D_n>$ be the discrete function/vector of the pixel-based distances between every pixel on one piece of contour $\{p_1,...,p_n\}$ and a second piece of contour $\{q_1,...,q_m\}$ on the opposite side, where $\forall i, 1 \le i \le n\ D_i = Pixel\_Distance(p_i,\{q_1,...,q_m\})$. The two portions could enfold a natural 'hole' if, and only if, a pixel exists on the first piece, for which the attached distance function/vector presents a substantial local minimum in respect to the distance values of the surrounding pixels. In this case, a substantial local minimum, used to reduce the effect of natural gaps that emerge from normal quantization noise, means that either at least one pixel separates the local minimum and the local maximum that follows it or the difference between the values at these points is at least one. The distance values measured at the local minimum and maximum are required to be less than or equal to stroke-width and above stroke-width, respectively:

$$Recover(\{p_1,\ldots,p_n\},\{q_1,\ldots,q_n\}) = 1 \text{ iff } \exists j\ i<j<k \mid (D_i > D_j) \wedge (D_j < D_k) \wedge ((k-j \ge 2) \vee (D_k - D_j \ge 1)) \wedge (D_j \le stroke\text{-}width) \wedge (D_k > stroke\text{-}width) \quad (9)$$

In addition, one may require that the other discrete function/vector, that represents the pixel-based distances between every pixel on the second piece of contour $\{q_1,...,q_m\}$ and the first piece of contour $\{p_1,...,p_n\}$, will also present a substantial local minimum point around the same

region.

Figure 9-(c) presents the distance functions between the left and right sides of an external contour fragment, Figure 9-(a)+(b), respectively, around the joint local maximum (see [46] for elaborations).

IV. Modular Solution and System

A. Overview

Natural and artificial loops are always separated between tarsi and axis, respectively, so the preliminary module of a complete loop investigation solution/system partitions the external contour into these two types. Next, axial loops can be located and formed based on matches between pairs of ascenders and descenders. For each one of these artificial loops, any encapsulated 'hole' is classified, validated, and labeled. Thus, natural 'holes' and sub-loops are distinguished and identified. Then, both hidden natural loops and sub-loops are recovered on tarsi and within enclosing walls of axial loops, respectively. See the flowchart in Figure 10**Error! Reference source not found.**.

B. Separating between axis and tarsi

The external contour partitions into axis and tarsi areas by using the *Correspondence* operator between every pixel on the upper side and all of the pixels on the lower side, and vise versa. For every significant chain of consecutive external contour pixels, where neither corresponds to the other side of the external contour, it would be considered a piece of tarsus.

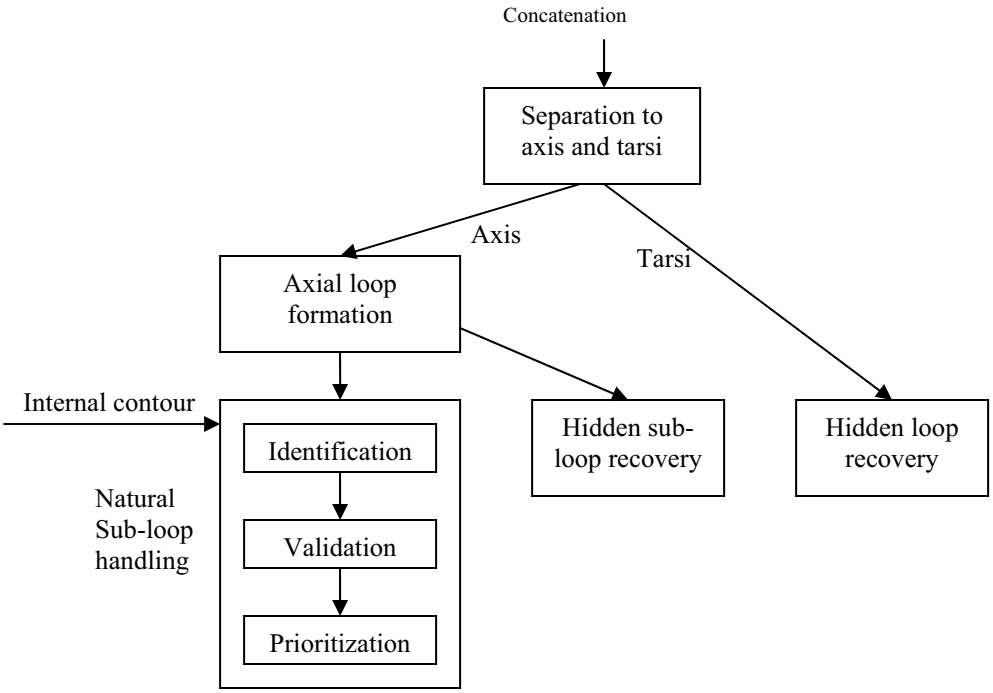


Figure 10: A flowchart of a complete loop investigation solution including axial and on-tarsus loop analysis in general and recovery of hidden 'holes' in particular.

Let $\{p_1,\ldots,p_n\}$ and $\{q_1,\ldots,q_m\}$ be the pixel-based representation of the upper and lower contours, respectively, then the subset $\{p_i,\ldots,p_j\}$ is a possible ascender if, and only if:

$$(Correspondence(p_k,\{q_1,\ldots,q_m\}) = 0) \wedge (j\text{-}i \geq stroke\text{-}width) \quad i \leq k \leq j \tag{10}$$

and the subset $\{q_r,\ldots,q_s\}$ is a possible descender if, and only if:

$$(Correspondence(q_t,\{p_1,\ldots,p_n\}) = 0) \wedge (s\text{-}r \geq stroke\text{-}width) \quad r \leq t \leq s \tag{11}$$

The remaining pixels on both sides of the external contour denote the axis parts.

See Figure 11**Error! Reference source not found.**-(a) for a full illustration of the extracted ascenders and descenders in white. In this case, the stroke-width proportional threshold was five pixels.

C. Forming axial loops

A pair of two matching opposite-sided tarsi on the upper and lower contours, an ascender and a descender, forms an axial loop if, and only if, no proven axial sub-stroke occurs between their roots on the axis. In this case, a proven axial sub-stroke requires correspondence between pixels

to the right of the ascender and others to the left of the descender, respectively, or vice versa. So

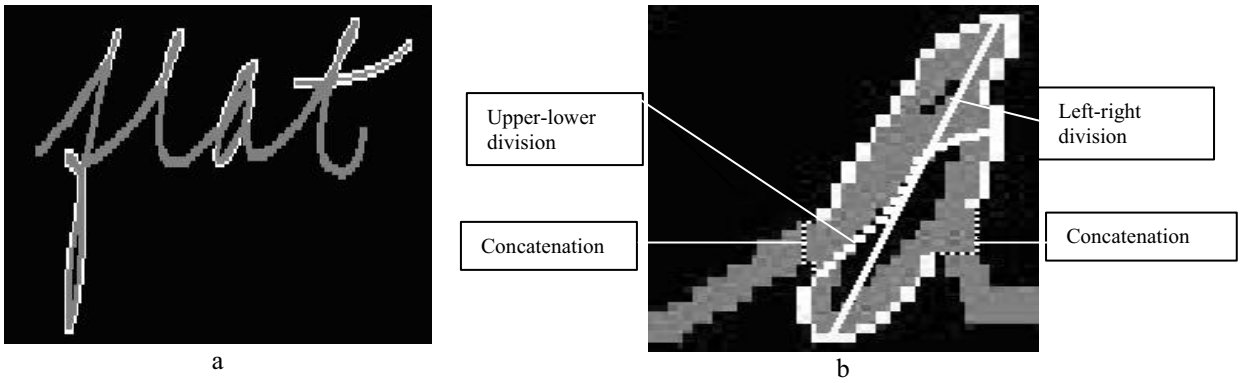


Figure 11: (a) An illustration of the ascenders and descenders in the word 'flat' following the separation of the axis parts. (b) A zoom-in illustration of an axial loop formation of the character 'a' by the concatenation (dashed lines) of a pair of matching ascender and descender (in white), and the suggested division mechanism of the area into quarters (full and semi-dashed white lines).

the ascender $\{p_i,\ldots,p_j\}$ matches the descender $\{q_r,\ldots,q_s\}$ for a common axial loop if, and only if:

$$\begin{aligned}
&\forall k\ k<i\ (Correspondence(p_k,\{q_1,\ldots,q_{r-1}\}) = 0) \wedge \\
&\forall k\ j<k\ (Correspondence(p_k,\{q_{s+1},\ldots,q_m\}) = 0) \wedge \\
&\forall t\ t<r\ (Correspondence(q_t,\{p_1,\ldots,p_{i-1}\}) = 0) \wedge \\
&\forall t\ s<t\ (Correspondence(q_t,\{p_{j+1},\ldots,p_n\}) = 0)
\end{aligned} \tag{12}$$

The complete perimeter of the axial loop is accepted by the concatenation of the left- and right-most pixels of the ascender and the descender, p_i with q_s and p_j with q_r, by the weighted shortest Geodesic path, biased in favor of the on-the-edge pixels.

Figure 11-(b) illustrates an axial loop formation by connecting the left and right ends of a pair of matching ascender and descender.

D. Distinguishing encapsulated natural 'holes' and identifying natural sub-loops

A natural 'hole' encapsulated within an axial loop is designated using the *Continuation* operator between the upper or lower external contour (separated to preceding and following sections) and the internal 'hole'-based contour. We suggest that a natural 'hole' refers to every segment of internal contour that presents a continuation potential with the relevant side of the external contour in the proper manner.

Let $\{c_1,\ldots,c_u\}$ and $\{e_1,\ldots,e_v\}$ be the section-based representations of the upper and lower

external contours, respectively, where the subsets $\{c_a,\ldots,c_b\}$ and $\{e_g,\ldots,e_h\}$ are a pair of matching ascender and descender that forms an axial loop, then the encapsulated segment of internal contour $\{f_1,\ldots,f_w\}$ is a possible natural 'hole' if, and only if:

$$\exists z \;\; 1 \le z \le w \mid \tag{13}$$
$$(\exists o \; 1 \le o \le a\text{-}1 \mid Continuation(c_o, f_z) = 1) \vee (\exists o \; b\text{+}1 \le o \le u \mid Continuation(f_z, c_o) = 1) \vee$$
$$(\exists o \; 1 \le o \le g\text{-}1 \mid Continuation(e_o, f_z) = 1) \vee (\exists o \;\; h\text{+}1 \le o \le v \mid Continuation(f_z, e_o) = 1)$$

When the first condition is satisfied, i.e. the natural 'hole' is a continuation of the current upper left part of the external contour, then the axial loop's left enclosing stroke is an upper left ascending natural sub-loop (see Figure 8-(b)). In a similar way, the second, third and fourth conditions refer to upper right ascending, lower left descending and lower right descending natural sub-loops at the right, left and right enclosing strokes, respectively.

E. Validating hypothesized natural sub-loops

Since an encapsulated natural 'hole' refers to a natural sub-loop, where the latter constructs one of the walls of the hosting axial loop, some significant relation must exist between matching parts of the internal and external contours at that environment. In particular, an upper left ascending natural sub-loop must associate with the left side of the ascender half of the axial loop, an upper right ascending natural sub-loop must associate with the right side of the ascender half of the axial loop, and descending natural sub-loops must associate with the originating side of the descender half of the axial loop. The meaning of association in this context denotes correspondence (Figure 8-(a)) and/or correlation (Figure 8-(b)) between the matching parts of the internal and external contours at the pixel and/or section level, respectively. The complete natural sub-loop validation is given in Appendix A.

F. Prioritizing hypothesized natural sub-loops

Given the artificial loop model, in which a natural sub-loop forms one, and only one, of the

loop's enclosing walls, two or more hypothesized natural sub-loops cannot co-exist when both associate with the same side of the hosting axial loop ({left, right}) or share the same encapsulated 'hole'. The complete natural sub-loop prioritization is given in Appendix B.

G. Recovering internal hidden sub-loops

A hidden sub-loop encapsulated within an axial loop is recovered using the *Recover* function between the internal contour of a proven artificial 'hole' and a related piece of the surrounding external contour on the relevant of the four sub-sides ({upper, lower} X {left, right}). A hidden sub-loop would refer to every blob enfolded between matching pieces of the internal and external contour that offer an internal contour recovery potential.

Let $\{\rho_{bottom},\ldots,\rho_{top}\}$ be the set of pixels on the left or right side of the internal contour and let $\{p_{matching\_bottom},\ldots,p_{matching\_top}\}$ be the set of pixels on the matching nearest piece of external contour, then a hidden sub-loop exists between the two if, and only if:

$$(Recover(\{p_{bottom},\ldots,p_{top}\},\{p_{matching\_bottom},\ldots,p_{matching\_top}\}) = 1) \vee (Recover(\{p_{matching\_bottom},\ldots,p_{matching\_top}\},\{p_{bottom},\ldots,p_{top}\}) = 1) \quad (14)$$

When investigating a vertical-oriented blob, it requires that $y_{matching\_bottom} = y_{bottom}$ and $y_{matching\_top} = y_{top}$, in case of a horizontal-oriented blob the x coordinates of the matching points prove identical.

H. Recovering on-tarsus hidden loops

An on-tarsus hidden loop is recovered using the *Recover* function between the left and right sides of the surrounding piece of external contour. A hidden sub-loop may refer to every tarsus with no visible 'hole' that presents recovery potential between its two sides.

Let $\{p_{preceding\_extremity},\ldots,p_{boundary}\}$ and $\{p_{boundary+1},\ldots,p_{following\_extremity}\}$ be the sets of pixels on the left and right sides of the external contour surrounding a tarsus, where both pieces touch at one

end (the boundary) and are trailed to the closest extremity points at the other end. A hidden natural loop exists between the two if, and only if:

$$(Recover(\{p_{preceding\_extremity},\ldots,p_{boundary}\},\{p_{boundary+1},\ldots,p_{followoing\_extremity}\}) = 1) \vee \\ (Recover(\{p_{boundary+1},\ldots,p_{followoing\_extremity}\},\{p_{preceding\_extremity},\ldots,p_{boundary}\}) = 1) \quad (15)$$

In contrast to the preferred boundary between the left and right parts of a tarsus constructing an axial loop, the first alternative is not a discontinuity point but rather the presumed location of the end-point acting as the symmetry axis. Usually this would be the top or bottom point for ascender and descender, respectively. See Figure 9-(a)+(b) with the on-tarsus hidden loop, plus the left and right trailed sides of the surrounding contour.

V. Experimental Results

A. Protocol

We will evaluate the proposed loop investigation algorithm on three tasks that demonstrate the robustness of our algorithm to find the isomorphism between static images of loops and the ground truth online trajectory. To continue our earlier analysis of this isomorphism, see Sections I and II, we have selected the following three key tasks:

- Classification of ‘holes’ encapsulated in axial loops;
- Identification (Recovery) of hidden natural loops located on tarsi; and
- Identification (Recovery) of hidden natural sub-loops encapsulated in axial loops.

The first task is equivalent to full axial loop resolution because collapsed artificial ‘holes’ can also be recovered in this way. The handwritten samples input for each experiment derived from images of pure cursive words, where loop investigation is highly crucial. The labeling process needed to create the ground truth for comparison purposes is labor intensive, so the amount of work that could be accomplished in a reasonable time has been limited. Under these

circumstances it is acceptable to test some aspects of robustness by sharing the three evaluated tasks with two databases. In this way, both cross-database and cross-task within the same database comparisons were achieved. For example, in each of the two databases the resolution of the images differed. As a result, the stroke-widths also differed. An extended session in which each experiment would be performed on all the available datasets falls beyond the scope of this work. Nonetheless, this paper reports all the experiments conducted to test the above-mentioned three tasks and does not filter inferior results.

All thresholds either remain constant or vary in accordance with a single parameter – the stroke-width. The latter parameter is neither writer-dependent nor word-related. Instead, it is shared by all the images of a database and should be estimated only once per database. For this purpose, we select the most popular value of a histogram measuring the distances between pixels on the external and internal sides of the contour, respectively. Without limiting generality, any stroke-width dependent parameter may be fine-tuned by shifts of few pixels - $\varepsilon < \textit{stroke-width}$. In a similar way, negligible shifts of several degrees may occur in angle-oriented parameters. Eventually a single fixed constant value will be established for each threshold per database. In this case, one may consider calculating the optimal value on a disjoint training set, and applying it to the remaining test set.

B. Data

1)Experiment 1: Encapsulated 'hole' classification

The experiment was conducted on 344 samples of the most frequent axial loop-oriented characters, *o-s* and *a-s*, that occurred in the middle of pure cursive words, and did not suffer from unrelated noisy phenomena typical to first and last letters. All samples originated from the same directory of the IRONOFF database ([47]), which includes a significant collection of cursive words. Forty-seven writers penned 1-20 words each. The IRONOFF database provides the online

signal and a gray scale image, taken simultaneously with the digitizing process, per sample. The off-line bitmap image representation was achieved by mapping all gray scale values below a selected threshold to 0, and all those above it to 1. The same threshold applied to all images in the database. This threshold was manually selected because it was an adaptation of the data to the required input format and not part of the tested algorithm. Strokes approximately five pixels wide were eventually achieved.

The 344 samples contained 540 'holes' according to the following distribution: 287 natural, 241 artificial, and 12 superfluous, see Figure 3 and Figure 4.

Table 4 provides the distribution of the 344 axial loops as a function of our natural sub-loop configuration mentioned in Section II, and in respect to the condition of the artificial 'hole' {hidden, visible} that shared a common encapsulation.

	No natural Sub-loops	Only left natural Sub-loop	Only right natural sub-loop	Two opposite natural sub-loops[1]	Total
Hidden artificial 'hole'	5	56	34	8	103
Visible artificial 'hole'	62	57	112	10	241
Total	67	113	146	18	344

Table 4: Axial loop distribution

2)Experiment 2: Hidden loop identification

The experiment was conducted on 1,273 pure cursive words taken from the Rumelhart's dataset ([48]), an extension of the *HP* dataset found in the UNIPEN collection ([49]). Six writers participated with 170-223 words each. In this case, the off-line bitmap image representation of a word was produced artificially by a linear concatenation of all neighboring pixels between every pair of pen-down, pen-up operations. Each line was three pixels thick on average, so for every pixel along its central mass approximately one pixel on each side appears along the gradient directions. Quantization aspects and intentional noise provided real conditions on the edges. See

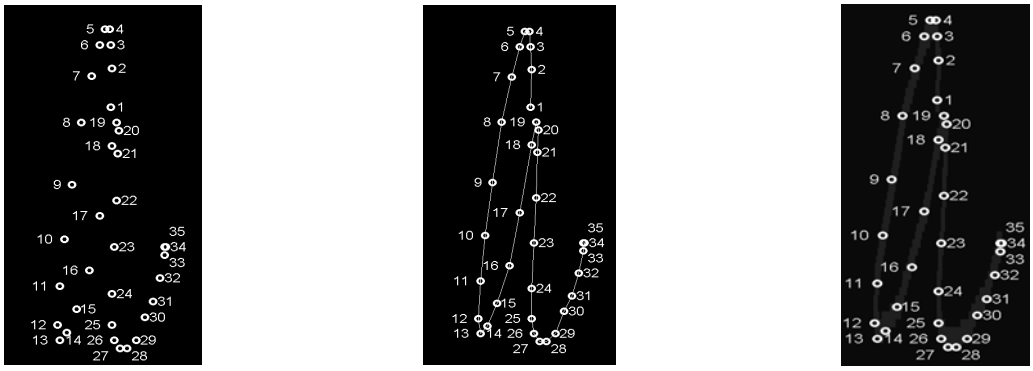
[1] Including a single 'hole' that share both sides, like an ascending\descending natural on-tarsus loop.

Figure 12 for illustration.

Given a total number of 10,131 tarsi, 1,211 (12 percent) were found irrelevant as being part of an axial loop, and 1,478 (14.6 percent) were disqualified because they had branches or were convex, i.e. twisted and ended away from the top or bottom point. Another 1,447 (14.3 percent) were uninteresting because they surrounded on-tarsus visible loops. Figure 13 illustrates several examples for each one of the above-mentioned filtered tarsus types. The remaining 5,995 tarsi were distributed as follows: 547 real hidden loops, 469 large hidden loops, and 4,979 poles (either small hidden loops or no loop), see Figure 1.

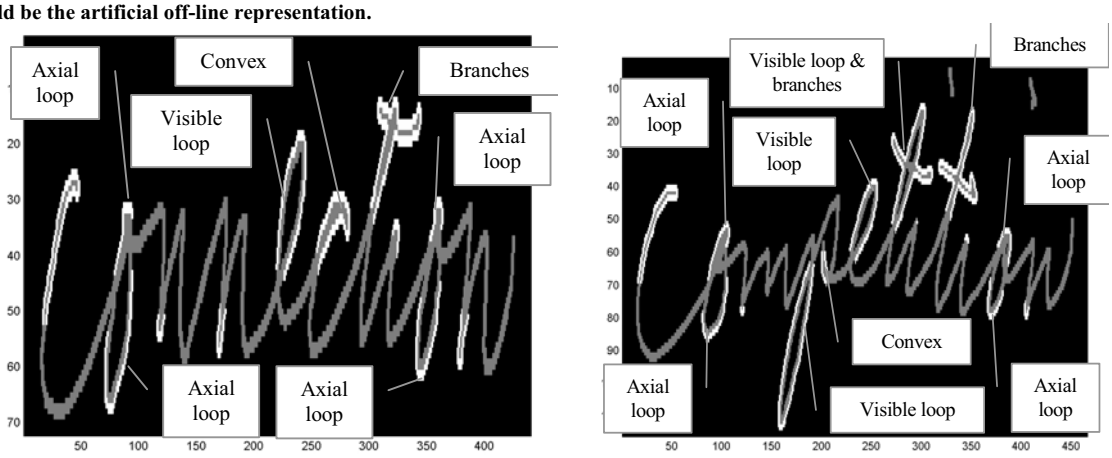
3)Experiment 3: Hidden natural sub-loop identification

A group of *a-s* acting as first letters was tested in the last experiment. The specific directory of the IRONOFF database ([47]) provided only a limited group of relevant words. Eventually, 95 words were collected from more than 40 writers, who produced one to four samples each. Nevertheless, eight of these words were manually disqualified because they were ambiguous. In addition, some of the remaining labels were overruled by our objective assessment of the true class that represents a given stroke (e.g. a very long hidden natural sub-loop that definitely stands for a pole). The remaining 87 letters were distributed between the two natural ‘hole’-less configurations: 51 hidden loops, and 36 poles, see Figure 14.



a b c

Figure 12: An illustration of the off-line bitmap image production: let the enumerated pixels in (a) be the discrete locations recorded by the digitizing tablet, then the thin trajectory in (b) would be the continuous online signal representation and the thick trajectory in (c) would be the artificial off-line representation.



a b

Figure 13: Four types of tarsi that are filtered-out and not tested for exclusive hidden loops: part of an axial loop, with branches, convex (twists and ends away from the top or bottom point) and surrounding of visible loops.

C. Results

1)Experiment 1: Encapsulated 'hole' classification

Given 259 of 287 authentic natural sub-loops (90.2 percent) were successfully detected, false alarms happened in 18 of 253 (7.1 percent) instances, where authentic artificial or superfluous 'holes' were mistakenly labeled as natural. This produces a total 'hole' identification rate of 91.5 percent (494/540).

The complete axial loop recognition rates appear in Table 5. In 80.2 percent of the axial loops, all encapsulated 'holes' were classified properly and associated with the correct side.

	No natural Sub-loops	Only left natural Sub-loop	Only right natural Sub-loop	Two opposite natural sub-loops	Total
Hidden artificial 'hole'	5/5 (100%)	44/56 (78.6%)	25/34 (73.5%)	7/8 (87.5%)	81/103 (78.6%)
Visible artificial 'hole'	54/62 (87.1%)	43/57 (75.4%)	91/112 (81.3%)	7/10 (70%)	195/241 (80.9%)
Total	59/67 (88.1%)	87/113 (77%)	116/146 (79.5%)	14/18 (77.8%)	276/344 (80.2%)

Table 5: Complete axial loop recognition rates

a b

Figure 14: The two natural 'hole'-less configurations of axial loops acting as first letter *a-s*. A hidden natural sub-loop forms the right enclosing stroke of the artificial loop on the left. The loop on the right utilizes a pole type of enclosing right stroke.

2)Experiment 2: Hidden loop identification

Given 517 of 547 authentic real hidden loops (94.5 percent) and 341 of 469 authentic large hidden loops (72.7 percent) were successfully detected, false alarms happened in 563 of 4,979 (11.3 percent) instances, where authentic poles were mistakenly labeled as hidden loops. This produces a total concave tarsus classification rate of 88 percent (5,274/5,995).

3)Experiment 3: Hidden natural sub-loop identification

Given 36 of 51 authentic hidden natural sub-loops (70.6 percent) were successfully detected, false alarms occurred in 10 of 36 (27.8 percent) instances, where authentic poles where mistakenly labeled as hidden sub-loops. This produces a total non-questionable stroke interpretation rate of 71.3 percent (62/87).

D. Discussion

1)Experiment 1: Encapsulated 'hole' classification

Approximately one third of the mis-recognized axial loops, 6.4 percent, could not have been resolved by the proposed algorithm, because they do not behave according to the artificial loop paradigm and present twists in place of continuous smooth strokes - see Figure 15-(a)+(b). One

particular writer produced most of these troublesome instances.

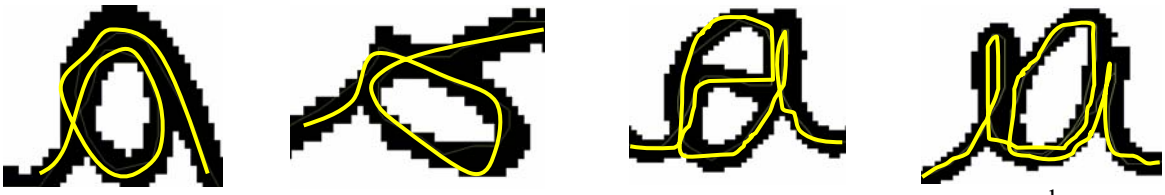
In an additional 2.9 percent of the cases, the existing natural sub-loop could not have been designated by the proposed algorithm because the presumed continuation does not satisfy the smooth inclination condition.

This extraordinary situation, illustrated in Figure 15-(c), occurs when an unexpected sharp turn is made in the middle of the stroke. In this case, the assumption that one maintains smooth strokes to lose as little energy as possible (sub-section III.F.3) was violated.

Noise offers explanation for the remaining portion of failure cases. In 1.2 percent, the neighboring character attaches to the substantial side of the axial loop, shadowing the origination of the required continuation (concatenation). A similar phenomenon was observed in another 1.7 percent of the investigated loops, where extra ascenders exist between the natural sub-loop and the axis. See an exemplar in Figure 15-(d).

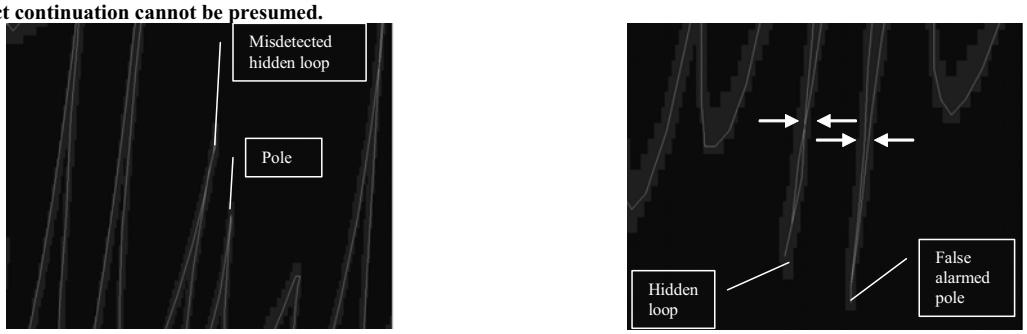
2)Experiment 2: Hidden loop identification

A vast majority of the misdetections / false alarms relates to quantization difficulties that affect the visualization of large hidden loops in contrast with thin poles. Sometimes, similar circumstances cause confusion between a real hidden loop and a thick pole that has a trapeze shape. The transformation of a pair of true diagonal lines, denoting the left and right sides of the (external) contour of a tarsus, into two sets of discrete pixels is not well defined. As one can see in Figure 16, two parallel lines, even those representing a cusp shape, could produce fluctuations in the mutual distance functions. As a result, hidden loops and poles can be substituted.



a b c d

Figure 15: Two misrecognized axial loops that do not behave according to the artificial loop paradigm: (a)+(b) the left stroke twists to the right and completes a natural sub-loop in a clockwise manner. Additional two groups of unresolved axial loops: (c) natural sub-loops with extraordinary deformations caused by unexpected sharp turns in the middle of the constructing strokes; (d) natural sub-loops blocked between pairs of sub-strokes, where one of each couple acts as an extra ascender that separates the 'hole' from the axis. In both cases a direct continuation cannot be presumed.



a b

Figure 16: (a) A misdetected large hidden loop in comparison with a similar-looking pole. Both tarsi present matching staircases, with only one pixel difference in the mutual distances. Thus, both resemble a pole. (b) A false alarmed pole, mistakenly considered as hidden loop, in comparison with a similar-looking large hidden loop. Both tarsi present narrow waists that widen a few pixels away, giving the impression of an aperture behind a collapsed loop.

In some cases, the scanning and digitizing resolution fails to distinguish the actual hidden loops. In other words, the identified large hidden loops may have been perceived as real, i.e. with a visible 'hole', if the resolution was higher. The mis-recognized hidden loops, however, are probably not actual. When the resolution increases, the signal-to-noise ratio of the *Recover* function can also be improved by demanding higher differences between the local minimum and maximum points. From a different point of view, 3.5 percent of the mistakenly recovered blobs contained small hidden loops which may be considered ambiguous.

3)Experiment 3: Hidden natural sub-loop identification

In two of five misinterpreted loops, the reasons for failure match those of Experiment 2. However, most of the confused strokes, approximately 16 percent of the error rate, concern bad handling of collisions between origin of the left stroke and the body of the right one.

Figure 17: Characteristic misinterpreted axial loops in the letter *a* at the beginning of a word: (a) a misdetection caused by the other stroke overwriting the aperture; (b) a misdetection that occurred when the attempt to detach the left and right strokes (see the crossing line) failed to preserve the original shape of the latter - as a result, no hints of a hidden natural sub-loop were found in the remaining blob; (c) an example of a misleading intersection between the left and right strokes, resulting in a false alarm - a cusp-shaped pole creates an illusion of a hidden natural sub-loop because the attached stroke forces narrow waists on the internal contour; (d) another false alarm that happened when the tail of the left stroke was not eliminated properly, leaving behind a false local minimum point in the distance functions between the internal and external contours.

Figure 17-(a)+(b) presents two examples of misdetections, where the critical parts of a genuine hidden natural sub-loop were concealed. Figure 17-(c)+(d) illustrates the potential of false alarms caused by interference of the other stroke.

VI. Discussion

We have proposed a novel contour-based handwritten analysis approach. The discussed method showed excellent results on various loop resolution scenarios including axial loop understanding and collapsed loop recovery. Although we did not present experimental results that demonstrate the direct impact of our algorithm on word recognition and writer modeling systems, the theoretic analysis and supporting references we described offer ample evidence of the importance of loop interpretation as a preprocessing step in such applications. In this case, when utilizing the suggested method, one may be provided with additional valuable information that may distinguish among loop-based handwritten patterns that appear similar in their topological and geometrical structure in advance.

Furthermore, we found strong evidence that loop understanding supports character recognition. In this case, we learned that the frequency of some common structures changes dramatically between letters. Writer identification/verification is supported similarly by the proposed algorithm, given that the correspondence between writers and styles of loops is high. The

robustness of the proposed algorithm was demonstrated using two databases that provided samples in different resolutions and stroke-widths. In particular, the method demonstrated its practicality for use with low-resolution images.

Future work would generalize the suggested method and expand the framework of events that can be treated. Improved recovery capabilities can be achieved by utilizing the preliminary dehooking technique that was partially practiced and discussed in the context of the hidden natural sub-loop identification on the first letter *a*-s. In such way, sub-branches, like the two sides of a *t-bar*, might be filtered. For the same purpose, it would be beneficial to have the ability of tracing genuine end-points that are neither the uppermost nor the lowermost pixels in order to handle convex tarsi such as *c-strokes* and open *s*-s. Complete implementation of this approach to the beginning and ending letters of a word also requires adaptation of the axial loop model in addition to the above-mentioned dehooking preprocess.

The presented algorithm is highly relevant to other tasks and applications in the field of handwriting analysis. We strongly believe the advanced tools developed here, along with the contour-based concept and guidelines, may prove useful in other fields of logical image understanding, where one can define a set of constraining rules on the visible edges and their interconnections. Examples could include medical visualizations, urban landscape interpretation, and automatic target recognition.

REFERENCES

[1] J.C. Simon and O. Baret. Regularities and singularities in line pictures. International Journal of Pattern Recognition and Artificial Intelligence, 5(1-2):57-77, 1991.

[2] R. Plamondon and S.N. Srihari. On-line and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):63-84, 2000.

[3] A. Vinciarelli, A Survey on Off-Line Cursive Word Recognition, Pattern Recognition, vol. 35, no. 7, pp. 1433-1446, 2002.

[4] T. Steinherz, E. Rivlin, and N. Intrator. Off-line cursive script word recognition – a survey. International Journal on Document Analysis and Recognition, 2(2-3):90-110, 1999.

[5] L. Schomaker and E. Segers. Finding features used in the human reading of cursive handwriting. International Journal on Document Analysis and Recognition, 2(2-3):13-18, 1999.
[6] V. Govindaraju and R.K. Krishnamurthy. Holistic handwritten word recognition using temporal features derived from off-line images. Pattern Recognition Letters, 17(5):537-540, 1996.
[7] V. Govindaraju, D. Wang, and S.N. Srihari. Using temporal information in off-line word recognition. In Proceedings of U.S. Postal Service 5th Advanced Technology Conference, pages 529-543, 1992.
[8] P.M. Lallican, C. Viard-Gaudin, and S. Knerr. From off-line to on-line handwriting recognition. In Proceedings International Workshop on Frontiers in Handwriting Recognition, pages 303-312, 2000.
[9] H. Bunke, M. Roth, and E.G. Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden Markov models. Pattern Recognition, 28(9):1399-1413, 1995.
[10] H. Bunke, M. Roth, and E.G. Schukat-Talamazzini. Off-line recognition of cursive script produced by a cooperative writer. In Proceedings International Conference on Pattern Recognition, pages 383-386, 1994.
[11] S.N. Srihari and Z. Shi. Forensic handwritten document retrieval system. In Proceedings International Workshop on Document Image Analysis for Libraries, pages 188-194, 2004.
[12] K. Frank, L.R.B. Schomaker, C. Veenhuis, L.G. Vuurpijl, M. Van Erp, and I. Guyon. WANDA: A common ground for forensic handwriting examination and writer identification. ENFHEX news - Bulletin of the European Network of Forensic Handwriting Experts, 1:23-47, 2004.
[13] C. Hertel and H. Bunke. A Set of novel features for writer identification. In Proceedings International Conference on Audio- and Video-Based Biometric Person Authentication, pages 679-687, 2003.
[14] J. Hochberg, K. Bowers, M. Cannon, and P. Kelly. Script and language identification for handwritten document images. International Journal on Document Analysis and Recognition, 2(2-3):45-52, 1999.
[15] N.E. Ben Amara and F. Bouslama. Classification of Arabic script using multiple sources of information: State of the art and perspectives. International Journal on Document Analysis and Recognition, 5(4):195-212, 2003.
[16] L. Vuurpijl, and L.R.B. Schomaker. Coarse writing-style clustering based on simple stroke-related features. In Proceedings International Workshop Frontiers in Handwriting Recognition, pages 29-32, 1996.
[17] K. Han and I.K. Sethi. Handwritten signature retrieval and identification. Pattern Recognition Letters, 7(1):83-90, 1996.
[18] V.S. Nalwa. Automatic on-line signature verification. IEEE Proceedings, 85(2):215-239, 1997.
[19] R. Sabourin, J.P. Drouhard, and E.S. Wah. Shape matrices as a mixed shape factor for off-line signature verification. In Proceedings International Conference on Document Analysis and Recognition, pages 661-665, 1997.
[20] R. Sabourin, Off-line signature verification: recent advances and perspectives. In Proceedings Brazilian Symposium on Document Image Analysis, pages 84-98, 1997.
[21] S. Chen and S.N. Srihari. Use of exterior contours and word shape in off-line signature verification. In proceedings International Conference on Document Analysis and Recognition, pages 1280-1284, 2005.
[22] J.C. Simon. Off-line cursive word recognition. IEEE Proceedings, 80(7):1150-1161, 1992.
[23] R.G.J. Meulenbroek, A.J.W.M. Thomassen, P.H.H.M. van Lieshout and S.P. Swinnen. The stability of pen-joint and interjoint coordination in loop writing. Acta Psychologica, 100(1-2):55-70, 1998.
[24] G. Boccignone, A. Chianese, L.P. Cordella and A. Marcelli. Recovering dynamic information from static handwriting. Pattern Recognition, 26(3):409-418, 1993.
[25] R. Plamondon and C.M. Privitera. The segmentation of cursive handwriting: An approach based on off-line recovery of the motor-temporal information. IEEE Transactions on Image Processing, 8(1):80-91, 1999.
[26] D.S. Doermann and A. Rosenfeld. Recovery of temporal information from static images of handwriting. International Journal of Computer Vision, 15(1-2):143-164, 1995.
[27] D.S. Doermann and A. Rosenfeld. The interpretation and reconstruction of inferring strokes. In Proceedings International Workshop on Frontiers in Handwriting Recognition, pages 41-50, 1993.
[28] S. Jaeger. On the complexity of cognition. In Proceedings International Workshop on Frontiers in Handwriting Recognition, pages 291-302, 2000.
[29] Y. Kato and M. Yasuhara. Recovery of drawing order from single-stroke handwriting images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(9):938-949, 2000.
[30] Y. Qiao, M. Nishiara, and M. Yasuhara. A novel approach to recover writing order from single stroke offline handwritten images. In Proceedings International Conference on Document Analysis and Recognition, pages 227-231, 2005.
[31] T.S. Cheng, C. Chiang, S.M. Roan, and H.C. Fu. Feature-preserving thinning algorithm for optical character recognition. In Proceedings of the SPIE – Character Recognition Technologies, pages 279-290, 1993.

[32] L. Lam, S.W. Lee, and C.Y. Suen. Thinning methodologies: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(9):869-885, 1992.
[33] B.K. Jang and R.T. Chin. Analysis of thinning algorithms using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(6):541-551, 1990.
[34] T. Caesar, J. Gloger, A. Kaitenmeier, and E. Mandler. Recognition of handwriting word images by statistical methods. In Proceedings International Workshop on Frontiers in Handwriting Recognition, pages 409-416, 1993.
[35] L. Lam and C.Y. Suen. An evaluation of parallel thinning algorithms for character recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(9):914-919, 1995.
[36] R. Plamondon, M. Bourdeau, C. Chouinard, and C.Y. Suen. Validation of preprocessing algorithms: a methodology and its application to the design of a thinning algorithm for handwriting characters. In Proceedings International Conference on Document Analysis and Recognition, pages 262-269, 1993.
[37] R. Plamondon, C.Y. Suen, M. Bourdeau, and C. Barriere. Methodologies for evaluating thinning algorithms for character recognition. International Journal of Pattern Recognition and Artificial Intelligence, 7(5):1247-1270, 1993.
[38] T. Steinherz, N. Intrator, and E. Rivlin. A special skeletonization algorithm for cursive words. In Proceedings International Workshop on Frontiers in Handwriting Recognition, pages 529-534, 2000.
[39] J.C. Pettier and J. Camillerapp. Script representation by a generalized skeleton. In Proceedings International Conference on Document Analysis and Recognition, pages 850-853, 1993.
[40] V. Pervouchine, G. Leedham, and K. Melikhov. Three-stage handwriting stroke extraction method with hidden loop recovery. In Proceedings International Conference on Document Analysis and Recognition, pages 307-311, 2005.
[41] I.S.I. Abuhaiba, M.J.J. Holt, and S. Datta. Processing of binary images of handwritten text documents. Pattern Recognition, 29(7):1161-1177, 1996.
[42] I.S.I. Abuhaiba, S. Datta, and M.J.J. Holt. Processing of text documents: straight line approximation and lost loop recovery. In Proceedings International Conference on Document Analysis and Recognition, pages 1157-1160, 1995.
[43] D.S. Doermann. Document image understanding: integrating recovery and interpretation. PhD thesis. University of Maryland, 1993.
[44] S. Madhvanath, G. Kim, and V. Govindaraju. Chaincode contour processing for handwritten word recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(9):928-932, 1999.
[45] T. Steinherz, E. Rivlin, N. Intrator, and P. Neskovic. An integration of online and pseudo-online information for cursive word recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5):669-683, 2005.
[46] D.S. Doermann, N. Intrator, E. Rivlin, and T. Steinherz. Hidden loop recovery for handwriting recognition. In Proceedings International Workshop on Frontiers in Handwriting Recognition, pages 375-380, 2002.
[47] C. Viard-Gaudin, P.M. Lallican, S. Knerr, and P. Binter. The IRESTE on/off (IRONOFF) dual handwriting database. In Proceedings International Conference on Document Analysis and Recognition, pages 455-458, 1999.
[48] D.E. Rumelhart. Theory to practice: a case study - recognizing cursive handwriting. Computational Learning and Cognition: Proceedings of the Third NEC Research Symposium, pages 177-196, 1993.
[49] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmarks. In Proceedings International Conference on Pattern Recognition, pages 29-33, 1994.
[50] D.K. W. Walters. Selection of image primitives for general-purpose visual processing. Computer Vision, Graphics, and Image Processing, 37(2):261–298, 1987.