Regular Paper

Recovery of Information on the Drawing Order of Single-Stroke Cursive Handwritten Characters from Their 2D Images

Tong Huang † and Makoto Yasuhara †

Handwriting is easier to recognize when it is done on-line than when it is done off-line, because for on-line handwriting one can use the time information obtained during the writing process, whereas for off-line handwriting one must depend solely on the static 2 D image, in which the time information has already been lost. The time information is essentially the order in which the stroke segments constituting the handwritten characters are drawn. We propose in this paper a novel approach to recovering the time information from a static 2 D image of single-stroke cursive script, based on a good continuity criterion implemented by applying the SLALOM approximation method. Departing from convention, we will evaluate smoothness not in the local area of crossing points but in the global area of the handwriting image, thus overcoming difficulties resulting from ambiguities or possible noise at the crossing points. To cope with the increased computational burden resulting from the complexity of the script, we divide the image into small components and rely on the principle of "divide and conquer."

1. Introduction

Handwriting recognition can be divided into two types, on-line and off-line. Although on-line recognition techniques have reached the level of practical applications, off-line techniques, except the OCR technique for printed or hand-printed characters, have not yet done so, because the information used for off-line recognition is much less powerful than that used for on-line recognition.

On-line recognition is characterized by the availability of time information, which makes it easier to recognize handwriting.[1),2)] By time information, we mean the time trace of the (x, y)-coordinates of pen-point movement, and possibly related elements such as writing pressure, velocity, and acceleration. In the conventional pattern-matching techniques, recognition is based on a one-dimensional vector signal, namely, a sequence of consecutive coordinate pairs recorded by a writing tablet.

On the other hand, off-line recognition is based on the static image obtained by scanning handwriting on paper, which contains no time information at all. The problem is analogous to the classic one of image understanding, which requires image enhancement, texture analysis, segmentation, perceptual reasoning, and so forth.[3)] Because of the tightly constrained feature space and the reduced need for segmentation, on-line recognition has many advantages over off-line recognition. The time information is essentially the order in which the stroke segments constituting handwritten characters are drawn. Thus, off-line character recognition would be greatly facilitated by recovering information on the drawing order of cursive handwritten character from their 2 D images because this would allow the use of on-line techniques.

However, the problem of recovering information on the drawing order, which is a kind of inverse problem, is very difficult because of its ill-posedness. That is, many possible solutions (drawing orders) may exist for a single handwriting image. Naturally, people tend to do things in such a way as to save energy; thus, like other human movements, the drawing movements are controlled by an attempt to minimize the energy used to produce them. This energy-saving results in minimal changes of curvature, and smooth continuation at junctions and crossings. This will be our fundamental principle in solving the ill-posed recovery problem.

Only a few studies have been done on recover-

† Department of Information Management Science, Graduate School of Information Systems, The University of Electro-Communications

ing time information from static handwritten 2 D images. Doermann and Rosenfeld[4] provided a taxonomy of local, regional, and global clues for recovering temporal properties from static images, but these clues depend on a knowledge of the mechanics of writing instruments. Boccignone et al.[5] adopted a method for reconstructing one of the most likely trajectories followed by the writer, based on a good continuity criterion. The authors of that paper noted that it may not be reliable to apply the good continuity criterion in the local area of each junction pixel on a digital plane, because of ambiguities or possible noise. They therefore presented a polygonal approximation for representing static handwritten images. The improvement is still within the local framework, however, and the more general problem still remains, especially at crossing points where the angle between two crossing stroke segments is very small.

In this paper, considering a situation in which ordinary handwritten numerals or characters or words in Latin alphabets are drawn in a single stroke and in which the pentip follows an almost fixed order of drawing movements, we propose a novel approach for recovering information on the drawing order from a given 2 D image of cursive single-stroke handwriting. This approach is based on a good continuity criterion implemented by applying the SLALOM approximation method,[6),7] which was originally developed for inverse quantization of digital signals. The SLALOM approximation method is the only one that satisfies the following two conditions: (1) the function form of the approximated curve is not predetermined, and (2) the curve does not always pass exactly through each of the given points, but sometimes passes only near them. Departing from convention, we evaluate the smoothness not in the local area of each crossing point of the strokes, but in the global area of the handwriting image, thus overcoming difficulties resulting from ambiguities or possible noise at the crossing point. To cope with the increased computational burden resulting from the complexity of the script, we divide the handwriting image into a collection of sub-images, called components, and apply the principle of "divide and conquer."

The paper is organized as follows. In Section 2, the SLALOM approximation method is reviewed. The mathematical basis for recovering information on the drawing order from a 2 D image of handwriting is discussed in Section 3. Experimental studies are presented in Section 4. Section 5 consists of some discussion and comments. Finally, we offer our conclusions in Section 6.

2. SLALOM Method

Evaluating the global smoothness of handwriting is the main problem in recovering time information from their 2 D images. In this section, we introduce a new good continuity criterion based on the SLALOM approximation method, which was originally developed for inverse quantization of digital signals. The SLALOM method is a way of obtaining a smooth curve when a sequence of sample points is given.

For a given sequence of sample points $f_1, f_2, \cdots, f_M$ on the digital image plane obtained by a scanner, we want to obtain a smooth function $g(x)$ having an arbitrarily small deviation error between $f_i (i=1, 2, \cdots, M)$, and $g(x)$ at $x_1, x_2, \cdots, x_M$, where $x_1 < x_2 < \cdots < x_M$.

We define smoothness as follows: To begin with, introducing a functional

$$J[g]=\int\left(\frac{d^2}{dx^2}g(x)\right)^2 dx, \tag{2.1}$$

we minimize $J[g]$ under the condition that

$$|g(x_i)-f_i|<\delta,\ i=1, 2, \cdots, M, \tag{2.2}$$

where δ is a small error tolerance. This is a conditional minimization problem in which

$$J[g]=\int\left(\frac{d^2}{dx^2}g(x)\right)^2 dx+\alpha\sum_{i=1}^{M}(g(x_i)-f_i)^2, \tag{2.3}$$

is minimized, where α is a parameter for controling the contribution of the conditional error to $J[g]$.

Rewriting Eq. (2.3) as a difference equation, we have

$$J[g]=\sum_{j=2}^{N-1}(g_{j-1}-2g_j+g_{j+1})^2+\alpha\sum_{i=1}^{M}(g_{j_i}-f_i)^2, \tag{2.4}$$

where N and M are the number of discrete points at which $g(x)$ is calculated and the number of sample points given, respectively. The term j_i denotes the discrete point corresponding to the i-th sample point, and satisfying

$$1\leq j_i\leq N,\quad j_i<j_{i+1}. \tag{2.5}$$

It follows, therefore, that $J[g]$ is minimized by solving the following linear system:

$$\begin{pmatrix} 1 & -2 & 1 & & & & & \\ -2 & 5 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & 1 & -4 & 6 & -4 & 1 \\ & & & & & 1 & -4 & 5 & -2 \\ & & & & & & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ \cdot \\ \cdot \\ \cdot \\ g_{N-2} \\ g_{N-1} \\ g_N \end{pmatrix} = \begin{pmatrix} \alpha\delta_{1,\Omega}(f_1-g_1) \\ \alpha\delta_{2,\Omega}(f_2-g_2) \\ \alpha\delta_{3,\Omega}(f_3-g_3) \\ \cdot \\ \cdot \\ \cdot \\ \alpha\delta_{N-2,\Omega}(f_{N-2}-g_{N-2}) \\ \alpha\delta_{N-1,\Omega}(f_{N-1}-g_{N-1}) \\ \alpha\delta_{N,\Omega}(f_N-g_N) \end{pmatrix}, \quad (2.6)$$

where Ω is the set of sample points, that is, $\Omega=(j_1, j_2, \cdots, j_M)$ and $\delta_{j,\Omega}$ is defined as

$$\delta_{j,\Omega}=\begin{cases} 0 & j\notin\Omega, \\ 1 & j\in\Omega. \end{cases} \quad (2.7)$$

This can be solved fast[7], since the matrix of the left side is of the band form. Thus we have the approximated curve presented by $g_1, g_2, g_3, \cdots g_N$. The maximum value $-J^*$ is defined as the good continuity criterion of the curve approximated.

So far we have discussed the SLALOM approximation method when the x coordinate of the sample point increase monotonously. However, the curves in a 2 D plane do not usually correspond to this, since the x, y coordinates of the sample point do not always increase monotonously. We therefore introduce a new monotonously increasing parameter u representing the distance from a fixed point along the curve. Thus, we have

$$J_x[g_x]=\int\left(\frac{d^2}{du^2}g_x(u)\right)^2du + \alpha\sum_{i=1}^{M}\left(g_x(u_i)-f_{xi}\right)^2, \quad (2.8)$$

$$J_y[g_y]=\int\left(\frac{d^2}{du^2}g_y(u)\right)^2du + \alpha\sum_{i=1}^{M}\left(g_y(u_i)-f_{yi}\right)^2, \quad (2.9)$$

in which f_{x_i} and f_{y_i} are the x and y coordinate functions, respectively, at the sample point u_i.

We can solve Eq. (2.8) and Eq. (2.9) in the same way as before, obtaining the approximation curve (g_{x_j}, g_{y_j}), $j=1, 2, \cdots, N$ for the given sequence of samples at u_i, $i=1, 2, \cdots, M$, $u_1 < u_2 < \cdots < u_M$.

We now define

$$S=-(J_x^*+J_y^*). \quad (2.10)$$

This S is the definition of our good continuity criterion, which we will use to evaluate the smoothness of the recovered sequence of sample points.

It is particularly significant to apply the SLALOM method to the present problem, because the curve approximated by the method does not always pass exactly through each of the sample points but sometimes passes only near them.

3. Recovery of Drawing Order of Cursive Script

In this paper, for simplicity we limit our discussion to the problem of recovering information on the drawing order of the cursive handwriting from its 2 D image on following assumptions:

1. The handwriting is done in a single *stroke*. By a stroke we mean a mark made by writing, beginning at the position where pen-point touched down on the writing surface, and ending at the position where it was lifted up.
2. The starting and the ending points are different points.
3. There exists no junction at which three or more stroke segments cross.
4. There exists no stroke segment that the pen-point traces more than once.

3.1 Notations and Preparations

Through the thinning process for computing the medial axis transform (MAT)[8] of a digital handwriting image composed entirely of an elongated stroke, an 8-connected digital line and/or curve called a *skeleton* is obtained with a unit width. An example is shown in **Fig. 1**.

Each of the pixels of the skeleton is labeled as a *terminal* pixel (TP), a *line* pixel (LP) or a *branch* pixel (BP), depending on whether its eight neighbors include one, two, or more than two skeleton pixels, respectively. TPs are re-labeled as STPs (*starting* terminal pixels), ETPs (*ending* terminal pixels) or HTPs (*hypothetical* terminal pixels). The resulting image is referred to as a *labeled pixel skeleton* (L_p-skeleton) as shown in **Fig. 2**.

A *line segment* consists entirely of an LP or a set of LPs linked to form a line or a curve. Further, for convenience, the line segment is classified into one of three kinds:

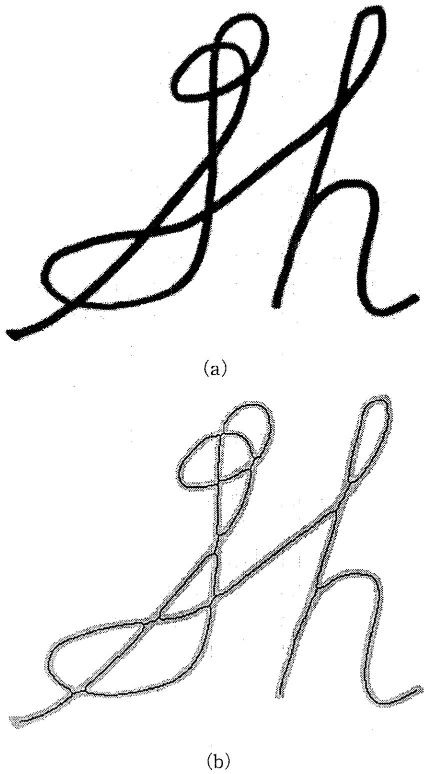


Fig. 1 An example of *skeleton.*

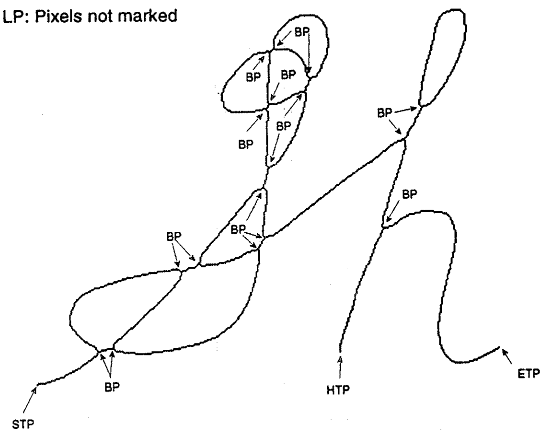


Fig. 2 Labeled pixel skeleton.

LINE : a line segment between a TP and a BP, or between two different TPs or BPs.

LOOP : a closed LINE having exactly one BP connected to it.

D-LOOP : a degenerated loop ; that is, a thin loop degenerated into a LINE as a result of the thinning process. Formally, a line segment between a BP and a HTP.

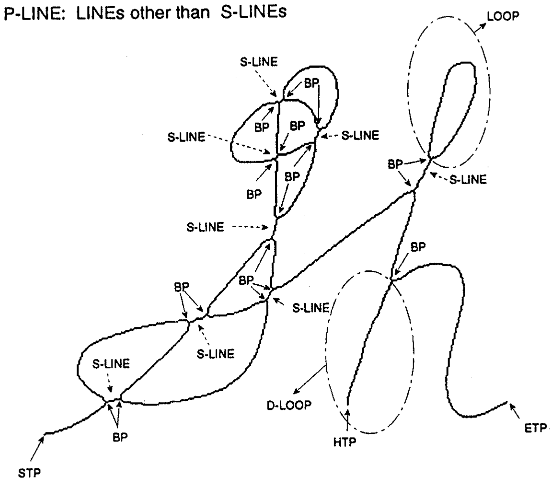


Fig. 3 Examples of P-LINEs, S-LINEs, LOOPs and D-LOOPs.

The thinning process often gives LINEs that do not exist in the original handwriting, because of noise created by quantizing analog handwriting images into digital ones. We refer to these as *spurious* LINEs (S-LINEs) ; they are usually short LINEs having BPs connected to their ends. We refer to the others as *proper* LINEs (P-LINEs) ; these form the complete instance of handwriting drawn in a single stroke. Examples of P-LINEs, S-LINEs, LOOPs, and D-LOOPs are shown in **Fig. 3**, which is referred to as an *labeled line skeleton* (L_l-skeleton).

Thus an L_l-skeleton can be seen as consisting of a set of BPs and TPs (STP, ETP, and HTP) and a union of LINEs (P-LINEs or S-LINEs). The problem is to give a formal algorithm for identifying all of the possible S-LINEs. This problem will be discussed later in section 3.4.

By assuming that BPs and TPs are vertices and that P-LINEs are edges, an L_l-skeleton can be viewed as a connected graph that gives a logical description of the handwriting, and in this case it is referred to as a *script graph* (S-graph). Note that each of the S-LINEs identified by the algorithm described later is assumed to be a vertex of an S-graph. See **Fig. 4** (a). A single-stroke of handwriting constructs in an S-graph an Eulerian path from the starting terminal vertex (STV) to the ending terminal vertex (ETV), that is, a path that passes along every edge of the graph exactly once. This means that every vertex except STV and ETV must be of even degree, where

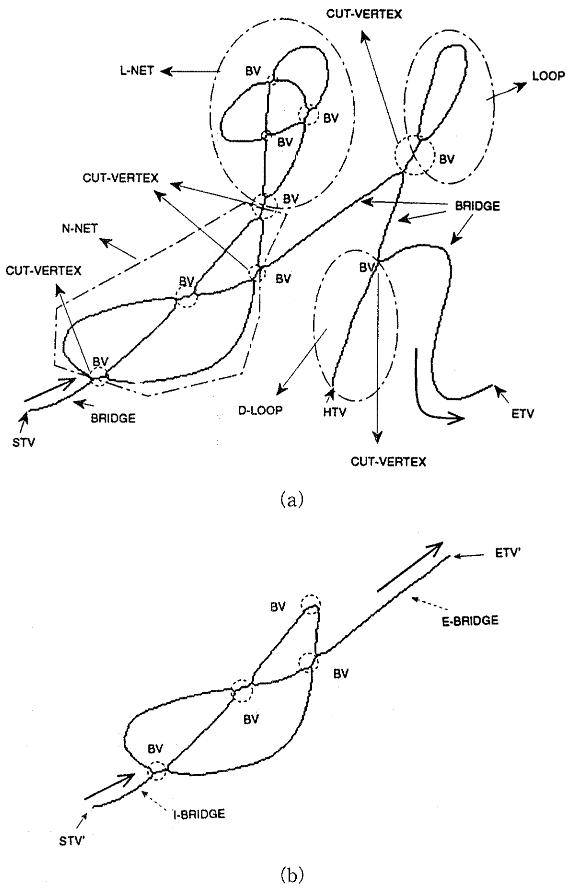


Fig. 4 Vertex of an S-graph.

each of the hypothetical terminal vertices (HTVs) is determined to be of degree 2 by assuming that a D-LOOP, which is a LINE of one round-trip, consists of two P-LINEs.

3.2 Statement of the Problem

Our problem is to find the order in which the line segments (P-LINEs) constituting a hand-written character or characters are drawn.

The only thing we can do is to search for the most likely path followed in writing the characters, by evaluating the good continuity criterion defined in the previous section. Corresponding to one of the possible Eulerian paths in an S-graph, we have an ordered sequence of P-LINEs in an L_t-skeleton along that Eulerian path. This is one of the candidate sequences of P-LINEs. Each of the P-LINEs is sampled at equal intervals from an STP to an ETP along the candidate sequence. By calculating the smoothness of each such the candidate sequence, we can find the smoothest one, thus obtaining a solution.

3.3 Complexity Reduction

Since the number of possible Eulerian paths in an S-graph increases rapidly with the number of edges, the task of searching the smoothest Eulerian path requires us to solve an optimal combinatorial problem. To avoid the problem of computational complexity, we introduce the following concept:

3.3.1 S-graph Decomposition

We decompose an S-graph into the maximal subgraphs having neither an edge called a BRIDGE nor a vertex called a CUT-VERTEX whose removal results in disconnection of the graph. An example is shown in Fig. 4 (a). We refer to the maximal subgraphs as *component graphs* (C-graphs), or simply *components* of an S-graph. Components are classified into four kinds: BRIDGE, LOOP, D-LOOP, and NET. An LOOP and a D-LOOP are corresponding to the LOOP and the D-LOOP in the same positions of L_t-skeleton, respectively. The following notes are provided to clarify our later discussions.

1. New terminal vertices (TV's), STV' and ETV', are defined for each of the C-graphs disconnected. See Fig. 4 (b).
2. An NET contains multiple edges, while each of the others contains a single edge. There are two kinds of NET: one having two BRIDGEs connected each to different branch vertices, a BRIDGE *incident* to one branch vertex (BV) on a NET called I-BRIDGE, and a BRIDGE *emanating* from another BV called an E-BRIDGE; and the other having no BRIDGE connected at all. The former is referred to as a *normal* NET (N-NET) and the latter as a *loop* NET (L-NET). See Fig. 4.
3. We make it a rule that an N-NET should include both an I-BRIDGE and an E-BRIDGE, because the angles of both the LINE incident to and the LINE emanating from a BP in an L_t-skeleton, corresponding to an I-BRIDGE and an E-BRIDGE in an S-graph, respectively, play important roles in our evaluation of the smoothness of each of the possible Eulerian paths in an N-NET. See Fig. 4 (b). Thus, the new terminal vertices STV' and ETV' of an N-NET are redefined as the starting vertex of an I-BRIDGE and the ending vertex of an E-BRIDGE, respectively. Note that it is always possible to differentiate between an I-BRIDGE and an E-BRIDGE

because every edge can be oriented and numbered in the order along which the Eulerian path is constructed in an S-graph.

4. There are two possible orientations for an LOOP and an L-NET, and one round-trip for a D-LOOP. Note that an L-NET can be regarded essentially as a LOOP that expands itself into an ordered sequence along the Eulerian path starting from and ending at the same BV.

Decomposing an S-graph into multiple smaller C-graphs enables us to reduce the original problem to a collection of smaller ones. Thus the complexity of enumerating all the possible Eulerian paths is expected to be greatly reduced.

3.3.2 Hierarchical Search for an Optimal Eulerian Path

Search at the C-graph level: For each of the C-graphs in an S-graph, by determining an ordered sequence of all the edges contained in the component along one of the possible Eulerian paths starting from STV' and ending at ETV', we can evaluate the smoothness of that Eulerian path by applying the SLALOM approximation method described in Section 2. Iteration of the above process for all of the possible Eulerian paths enables us to obtain the smoothest one, which is an optimal Eulerian path for that component.

Search at the S-graph level: By determining an optimal Eulerian path for each of the C-graphs in the above process, we can reconstruct the S-graph by connecting each of the C-graphs back to its original vertex or vertices. At this time, we have an ordered sequence of all the edges contained in the whole S-graph along one of the possible Eulerian paths starting from STV and ending at ETV, which corresponds to a candidate sequence of all the P-LINEs in the L_l-skeleton. We are thus able to evaluate the smoothness of one of the total Eulerian paths once again by applying the SLALOM approximation method. According to the note 4 in the previous subsection, a total of 2^n possible Eulerian paths exist, where n is the total number of LOOPs and L-NETs contained in the S-graph. Thus, the iteration of the above process for all of the 2^n Eulerian paths enables us to find the smoothest one in the global sense, giving us a final solution.*

3.4 Construction of an S-graph

This section describes the construction of an S-graph from an L_l-skeleton, where each of S-LINEs contained in the L_l-skeleton is to be assumed as a vertex. The problem is to find all of the possible S-LINEs contained in the L_l-skeleton.

3.4.1 Identification of an S-LINE

To begin with, we introduce the following theorem 1 without proof. See **Fig. 5**.

Theorem 1: *For a given skeleton image there exists a labeling algorithm that transforms it into an L_p-skeleton, in which we have the following two types of BP/BPs and no others,*

Type 1: classified as consisting of a single BP to which 3 or 4 LINEs are connected.

Type 2: classified as consisting of 2, 3, or 4 BPs in a cluster to which 4 LINEs are connected.

Where 8-connectedness is assumed.

The existence of this labeling algorithm is guaranteed[9] under the assumptions 3 and 4 introduced at the beginning of this section. Note that the number of LINEs connected to a BP or a BP cluster must be 3 or 4.

We are now presented with an L_p-skeleton containing two different TPs: an STP and an ETP. Note once again that an L_p-skeleton consists of a set of BPs, TPs (STP, ETP, and HTP), and BP clusters, and a set of line segments. Each of the line segments is either a P-LINE or an S-LINE, but is not yet identified. For such an L_p-skeleton, we have the following theorem:

Theorem 2: *It is possible to construct at least one single-stroke group of handwritten characters to be drawn from STP to ETP by passing exactly once along each of the P-LINEs contained in an L_l-skeleton if and only if each of the S-LINEs, if it exists, is passed along exactly twice.*

Proof:

1. The proof of necessity: From theorem 1, S-LINEs must occur, if they exist, exclusively at BPs to each of which 3 LINEs are connected. See case (a) of type 1, shown in Fig. 5. Note that we have no problem at all for BPs that follow the case (b) of type 1 and for BP clusters in type

* According to our experiments, the genetic algorithm has been proven quite successful when used with large n in searching for the optimal orientation of each of the LOOPs and L-NETs

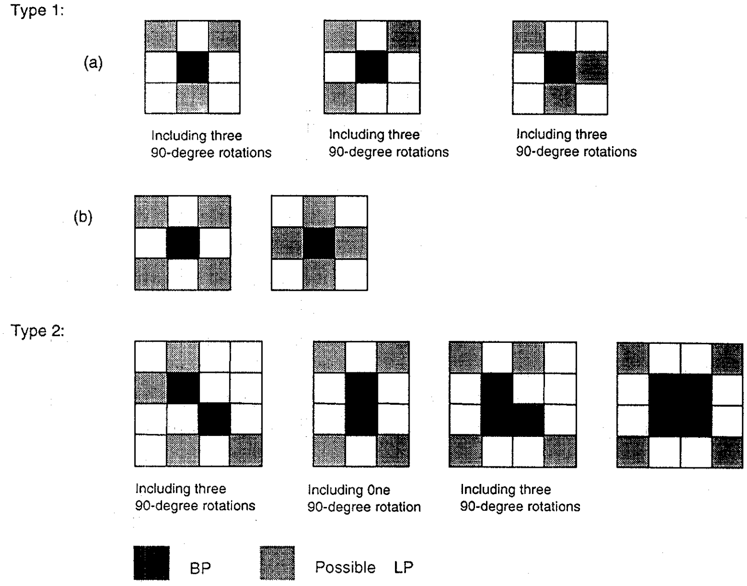


Fig. 5 Identification of an S-LINE.

2, because they each have 4 P-LINEs connected, provided that a BP cluster is represented by a single BP. Since 3 LINEs are connected to one BP, it is necessary for at least one of them to be passed along a number of times, say m times (m=2, 3, 4, …), to allow the construction of at least one single-stroke group of handwritten characters by passing exactly once along each of the P-LINEs contained in an L_l-skeleton. Such a LINE that is passed along a number of times is defined as an S-LINE. Now let p and s be the numbers of P-LINEs and S-LINEs, respectively. Thus $p+s=3$. From assumptions 3 and 4, $p+ms=4$. The integer solution of these two equations is limited to $p=2$, $s=1$, and $m=2$.

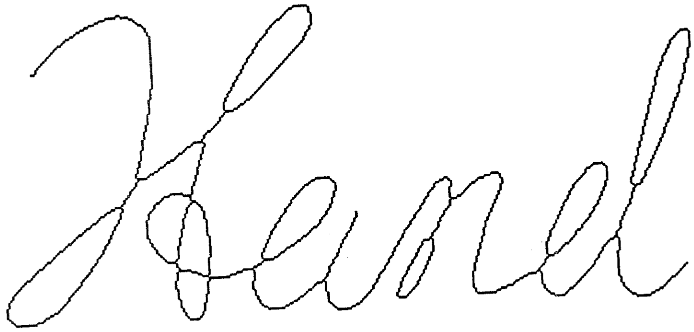
2. The proof of sufficiency: If each of the S-LINEs is passed along exactly twice, and together the theorem 1, if it is assumed that each S-LINE consists of 2 P-LINEs and if each of the BP clusters of type 2 is represented by a single BP, each of the BPs contained in an L_l-skeleton is to have 4 P-LINEs connected to it, thus allowing us to construct at least one single-stroke group of handwritten characters from STP to ETP by passing exactly once along each of the P-LINEs and twice along for each of the S-LINEs contained in the L_l-skeleton.

3.4.2 Algorithm for Identifying S-LINEs and P-LINEs—Algorithm 1

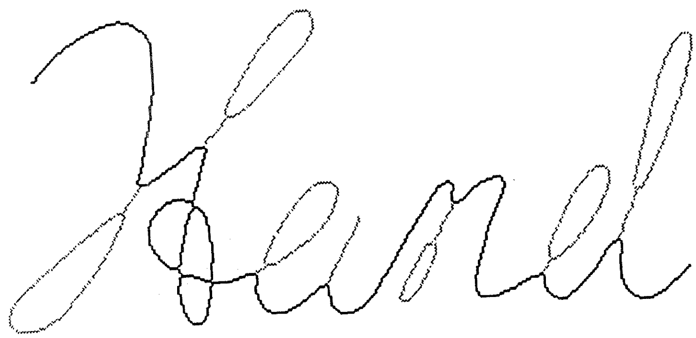
According to theorem 2, an algorithm for identifying both S-LINEs and P-LINEs for a given L_p-skeleton may be given as follows to obtain the L_l-skeleton:

1. For each of the BPs to which 3 LINEs are connected, that is, a *triple LINE branch* (TLB),
   Case A: Execute the following statements (A1)-(A4) if there is a LOOP connected to the TLB (see the case (a) in type 1 of Fig. 5) :
   A1 Identify the LINE forming the LOOP as a P-LINE and remove that LOOP.
   A2 Identify the remaining LINE as an S-LINE.
   A3 Remove the identified S-LINE.
   A4 Identify the 2 LINEs connected to the other end of the removed S-LINE as P-LINEs.
   Case B: Execute the following statements (B1) and (B2) if there exists a D-LOOP connected to the TLB:
   B1 Assume that the LINE forming D-LOOP is made up of 2 P-LINEs, and remove that D-LOOP.
   B2 Identify the 2 remaining LINEs as P-LINEs.
   Case C: Otherwise, do nothing
2. Identify all of the LINEs connected to

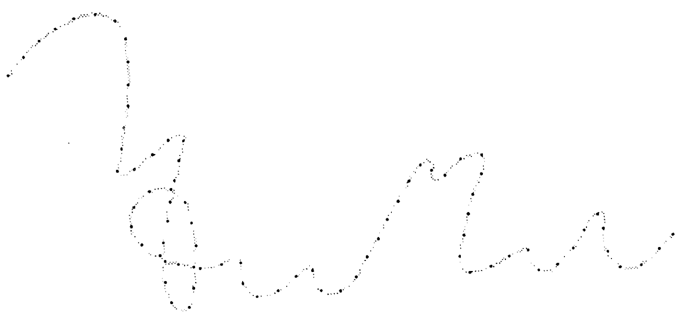
(a)



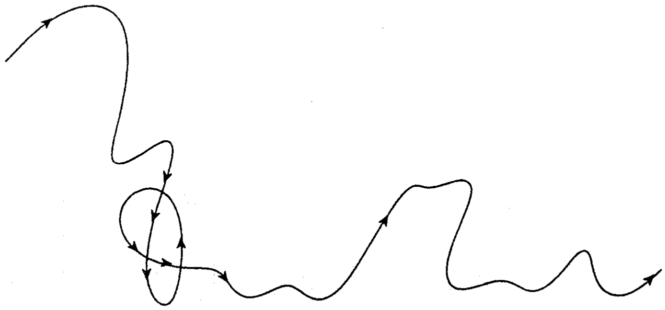
(b)



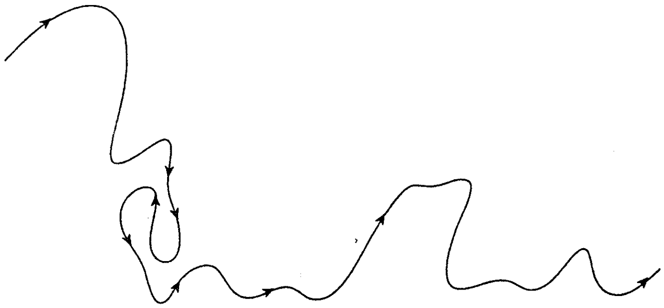
(c)



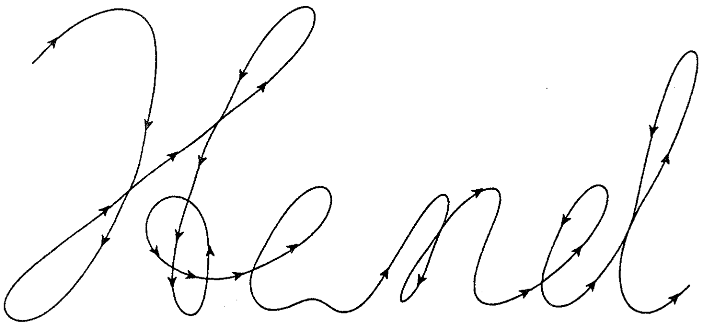
(d)



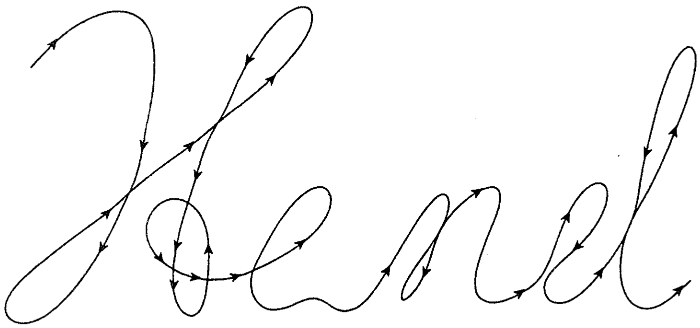
(e)



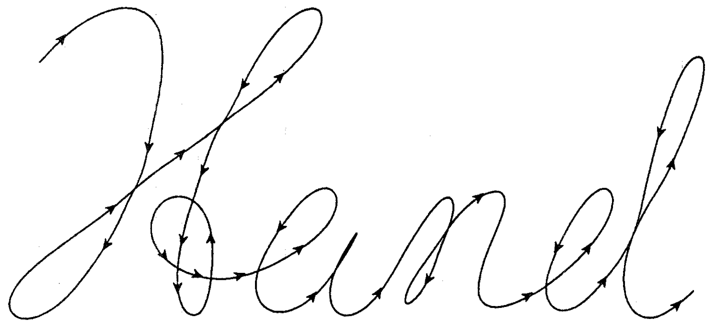
(f)



(g)



(h)



(i)

(j)

Fig. 6 Test sample.

BPs in case (b) of type 1 or to BP clusters in type 2 as P-LINEs.

3. Identify each LINE to which either an STP or an ETP is connected as a P-LINE each.
4. For each of the TLBs, letting p be the number of P-LINEs that have already been identified, execute again
   - Case A: if $p=0$ or 1, identify the LINE having the shortest length as an S-LINE and the others remained as P-LINEs.
   - Case B: if $p=2$, identify any unlabeled LINE as an S-LINE.
   - Case C: if $p=3$, wrong identifications must have occurred in case A above. Error correction may be impossible theoretically.

This algorithm guarantees that each of the TLBs has exactly one S-LINE and two P-LINEs unless case C in statement 4 occurs. Fortunately such cases may be expected to occur only very rarely with normal data.

3.4.3 Algorithm for Constructing S-graph—Algorithm 2

Finally, we have the following algorithm for constructing an S-graph from an L_l-skeleton: Construct a graph corresponding to a given L_l-skeleton by assuming that

1. Each of the BPs, BP clusters, and TPs is a vertex.
2. Each of the S-LINEs identified above is a vertex, into which 2 BPs connected to its ends are to be merged,
3. Each of the P-LINEs is an edge.

Thus we can construct an S-graph from a given L_l-skeleton. It is clear from the algorithms that the constructed S-graph has at least one Eulerian path.

4. Experimental Studies

Our method has been tested through the extensive experimental studies of samples such as that shown in **Fig. 6** (a). Through the thinning process we obtain the skeleton image (b). To begin with, in executing the algorithm for identifying S-LINEs and P-LINEs (algorithm 1) after the labeling process for the skeleton image, we find and remove all of the LOOPs together with the S-LINE, if one exists, connected to each of them and the D-LOOPs. These loops are shown in (c) by gray lines. The remaining is referred to as a *kernel skeleton* (K-skeleton), is shown in (c) by a black line. Note that all the BRIDGEs are included in the K-skeleton. Constructing an S-graph by applying algorithm 2 for the K-skeleton, we determine one of the possible Eulerian paths. Along that Eulerian path, the K-skeleton is sampled at equal intervals from STP to ETP, as shown in (d). Thus we can evaluate the smoothness by using the SLALOM approximation method. For the cases shown in (e) and (f), the S values were −22.93 and −23.96, respectively. Thus the former case (e) is more possible than the latter in this particular example. Next, when the removed LOOPs have been connected back to their original positions, the Eulerian path having the maximum smoothness is searched for again. Of the examples shown in (g) and (h), (g) was found to be the most likely. Finally, the removed D-LOOP is also connected back to its original position, and the final curve along the smoothest Eulerian path is calculated. See (i) and (j).

Note that, for convenience, the actual processes followed in the experiment were slightly different from those explained in section 3. However, it is easy to prove that latter is strictly correct.

5. Discussion

As mentioned in the introduction, only a few papers have been published on recovering time information from static 2 D images of handwriting. However, the key issue has been revealed to be that it is not reliable to apply a good continuity criterion in the local area of each of the junction pixels on the digital plane, because of ambiguities caused by quantization noise. This has remained a problem, especially at crossing points where the angle between two crossing stroke segments is very small.

Although we rely on the good continuity criterion as a key principle, our approach evaluates the criterion not in the area of each junction pixel, but in the whole or global area of the handwriting image. For this purpose we have put forward two key ideas: (1) introduction of a new good continuity criterion based on the SLALOM approximation method, and (2) application of the concept of an Eulerian path. The SLALOM approximation method meets our requirement that the approximated curve need not pass exactly through each of the given points but pass only near them. A disadvantage of (2) is that we incur a heavier computational burden by enumerating the Eulerian paths. Thus we have decomposed the handwriting image into a collection of sub-images called components, to reduce the computational complexity. The following comments also apply to the above discussion.

1. **Figure 7** shows the zoomed-up results for a part of the handwritten word processed in the experiment. Careful inspection reveals that it is possible to recover the trace of drawing successfully even for an extremely thin loop, where the loop is so thin as to have degenerated into a line. Note that researchers recognize the difficulty of determining whether or not a loop or line crossing exits when the loop is thin or the angle between two crossing lines is small.

2. The computational complexity of enumerating the Eulerian paths is $O(k!)$ where k is the number of the edges contained in the graph. Although handwritten characters and words are composed of BRIDGEs, LOOPs, D-LOOPs and NETs, most components of the ordinary handwritten numerals or characters and words in Latin alphabets are BRIDGEs, LOOPs, and D-LOOPs. NETs probably do occur not so frequently. Thus, we should add that the decomposition of an S-graph into a collection of smaller components has in fact greatly reduced the computational cost in those cases.

Finally, representation of one or more handwritten characters as a sequence of sampled points may be considered more compact than that in an image on the digital plane. If it is possible to reduce the number of the sample points and to keep the reconstruction error

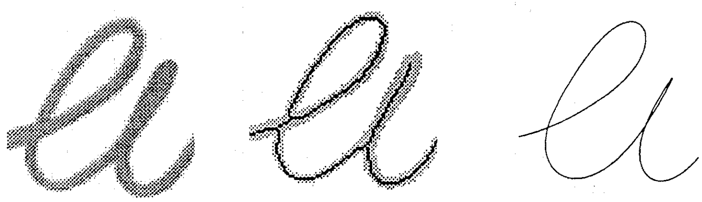


Fig. 7 Zooming-up of handwritten word.

within a permissible level, our method should also be useful for data compression of line drawings.

6. Conclusion

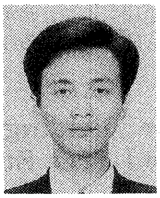
We have proposed a novel approach to recovering information on the drawing order of cursive handwritten characters from their static 2 D images, based on a good continuity criterion derived from the SLALOM approximation method. Departing from convention, we evaluate the smoothness of the recovered handwriting not in the local area of each of the crossing points, but over the total stroke of the handwriting. In this way we can overcome difficulties resulting from local ambiguities or noise. To cope with the increased computational burden resulting from the complexity of the handwriting, we divide the script into smaller components, relying on the principle of "divide and conquer." Extensive experimental studies have verified effectiveness of the proposed method.

Finally, for simplicity, we have introduced four assumptions, the problem of how to do away with these assumptions remains open.

References

1) Berthod, M.: On-line Analysis of Cursive Writing, *Computer Analysis and Perception, Vol. 1: Visual Signals*, Suen, C. Y. and De Mori, R. eds., pp. 55-81, CRC Press, Boca Raton (1982).
2) Tappert, C. C., Suen, C. Y. and Wakahara, T.: On-line Handwriting Recognition—A Survey, *Proc. 9th Int. Conf. on Pattern Recognition*, Rome, Vol. 2, pp. 1123-1132 (1988).
3) Simon, J. C.: Off-Line Cursive Word Recognition, *Proc. IEEE*, Vol. 80, No. 7, pp. 1151-1161 (1992).
4) Doermann, D. S. and Rosenfeld, A.: Recovery of Temporal Information from Static Images of Handwriting, *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition*, Urbana, pp. 162-168 (1992).
5) Boccignone, G., Chianese, A., Cordella, L. P. and Marcelli, A.: Recovering Dynamic Information from Static Handwriting, *Pattern Recognition*, Vol. 26, No. 3, pp. 409-418 (1993).
6) Isomichi, Y.: A New Technique for the Anti-quantization Problem, *The Proceeding of the 1st Symposium on Information Theory and Its Applications*, pp. 27-32 (Nov. 1978) (in Japanese).
7) Isomichi, Y.: Inverse-Quantization Method for Digital Signals and Images, *IEICE Trans.*, Vol. J64-A, No. 4, pp. 285-292 (1981) (in Japanese)
8) Rosenfeld, A. and Kak, A. C.: *Digital Picture Processing*, Academic Press, New York (1977).
9) Pavlidis, T.: *Algorithms for Graphics and Image Processing*, Chapter 9, Computer Science Press (1982).

(Received October 28, 1994)
(Accepted May 12, 1995)



Tong Huang was born in Anhui China, in September 1963. He received the B. S. degree in 1984, from the University of Anhui, China, and M. S. degree in 1991, from the University of Electro-Communications, Japan, respectively. He will receive the Doctor of engineering in September this year, from the University of Electro-Communications. He currently joined the University of Electro-Communications and is an assistant of the Department of Information Management Science, the Graduate School of Information Systems. His research interests include the pattern recognition, neural networks, signal processing, information mathematics and their applications.

Makoto Yasuhara was born in Tokyo in 1939. He received the B. S. from the University of Electro-Communications in 1963, and the Doctors of Medical Science and Engineering in 1974 and in 1986, respectively, both from the University of Tokyo. He joined the University of Electro-Communications in 1966 and was a professor of the Department of Computer Science and Information Mathematics from 1987 to 1993 and currently a professor of the Department of Information Management Science, the Graduate School of Information Systems. His research interests include the stochastic system theory, the control theory and their applications. He is a member of the IEEE, the IEICE, the IPSJ and the SICE.