A Novel Approach to Recover Writing Order From Single Stroke Offline Handwritten Images

Yu Qiao, Mikihiko Nishiara and Makato Yasuhara
University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585, Japan
{qiaoyu, mikihiko,yas}@math-sys.is.uec.ac.jp

Abstract

Problem of recovering the writing order from single-stroked handwritten image can be seen as finding the smoothest Euler path in its graph representation. In this paper, a novel approach is proposed to solve the recovery problem within the framework of the Edge Contiguous Relation (ECR). Firstly, we make local analyses to obtain the possible ECRs at each of the nodes; secondly a global trace is executed to find all of the candidate Euler paths and the smoothest one is selected as a final result. Based on two simple assumptions, we prove a series of theorems to obtain possible ECRs at even node. Double-traced lines are identified by using the weighted matching of general graph. Experiments on the scanned images and offline images converted from the online data of Unipen database have shown that our method achieved 95.2% correct recovery rate.

1. Introduction

Automatic handwriting analysis and recognition has received significant attentions in the field of computer vision and pattern recognition for more than 60 years [1]. Basically handwriting recognition can be categorized into two classes: online and offline handwriting recognition, which differ in the input device and available information. It is well known that online recognition can achieve higher accuracy than the offline one [1], [4], [6], [10]. The success of online recognition depends largely on availability of temporal information of pen-tip trajectory. In this paper, we focus on the problem to recover the drawing order from single-stroked handwritten images. This is to convert two-dimensional image to a sequence of vectors composed of pen-tip positions along time. The recovery of writing order can be seen as a bridge from offline type to online type handwriting recognition.

The approaches of recovering writing order information can be roughly divided into two main categories: local tracing method and global graph searching method. (1) In the local tracing method, the contiguous path is selected at each junction according to its present local configuration and the past tracing history. Lee and Pan [2] traced the skeleton of offline signature by a set of heuristic rules. Doremann and Rosenfeld [9] described a stroke recovery platform based on local regional and temporal clues, and reported detailed information which can be used for recovery. Liu, Huang, and Suen [3] proposed a stroke segmentation method for Chinese characters by using of polygonal approximation and certain rules. V. Govindaraju and N. Sriharo [7] presented an approach for separating handwritten text from interfering strokes based on Gestalt's segmentation and grouping principles. Plamondon and Privitera [10] developed a scanning method to find the natural course of strokes from calculating the curvature of contour. 2) In the global graph search methods, graph models are constructed to describe an input image and then the recovery problem is transformed to find a path (single stroke) or paths (multi-stroke), which cover all the edges and are most suitable to human writing habit. Both Huang & Yasuhara [4] and Jäger [5] used the graph model to represent the skeleton and searched an Euler and a Hamilton path, respectively, which minimized certain cost function. However, both may caused combinatorial explosion problem. Kato and Yasuhara [6] presented a 2-phase analysis by using pre-labeled information and basic tracing algorithm, and avoided the combinatorial explosion. In [12], Qiao and Yasuhara proposed a graph based method in a probability framework.

In this paper, a novel approach is proposed within the framework of *Edge Contiguous Relation* (*ECR*). The scheme is given in Fig. 1. At first, we construct the graph model from the skeleton image (Section 2), secondly the possible ECRs at even nodes are obtained by analyzing the node structures (Section 3), a neural network classifier is used for 4-degree nodes, and a theoretical framework is presented for analyzing node of degree 6 or greater, thirdly the double-traced lines are identified by employing weighted matching of

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

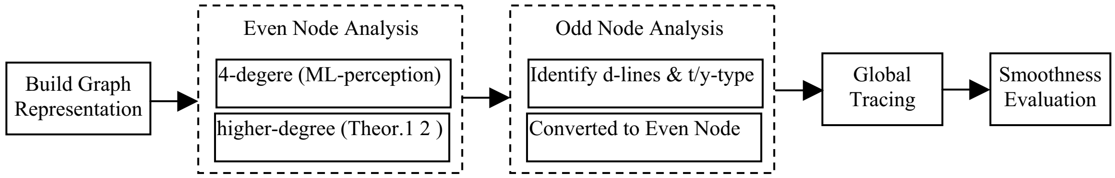


Figure 1 Scheme of the Recovery Method

general graph and then we show how to obtain the possible ECRs at odd node (Section 4); in Section 5 we execute global tracing to find all the possible paths from start to end node, evaluate smoothness for each of them, and select the smoothest one as a final result. Finally experimental results and conclusions are given in Section 6.

2. Construction of graph model

The first step is to construct *graph model* $G=(M, E, R)$ which represents both topological and geographical structures of input handwriting image [12]. M and E are sets of nodes and edges of G, respectively, and R is a set of geographical information of M and E. G is constructed from a *skeleton image* of the input image. We use smoothing filter proposed in [13] to reduce various types of noise such as peaks and holes in the input image before applying thinning algorithm to obtain the skeleton. The smoothing and the thinning algorithms influence substantially on the graph model constructed.

In order to construct graph model G, it is necessary to extract vertices and segments from the skeleton. *Vertex* is a local geometrical point at which a segment terminates (*terminal vertex*) or multiple segments joint (*joint vertex*). The terminal vertex can easily be detected as a pixel having one 8-connected neighbor, while the joint vertex may be a pixel or a cluster of connected feature pixels. The configuration of feature pixels depends on the thinning process, and can be found by the method in [8]. A part of skeleton between two vertices is referred to *segment*. There are two kinds of segments [6], [11], [12]: 1) *Real segment* (*r-segment*) corresponds to a part of real stroke in an original image; 2) *spurious segment* (*s-segment*) is unwanted output from the thinning process that never exists in the original image. This undesirable s-segment will distort the structure of the skeleton image. Hence, in order to construct the graph model, it is necessary to differentiate the s-segment from the r-segment. We identify the s-segment based on its length and stroke width in the original image [12]. After all the s-segments are identified, we cluster connected s-segments together with associated vertices into a node. The terminal vertices are converted also to nodes of G. Real segments are reserved as edges of G.

In this paper, *stroke* is a writing path from pen-down to pen-up. *line* is a part of the stroke, which may contain one or more edges. The recovered handwriting trajectory from *start node* to *end node* is called *path.* The node composed of connected s-segments can be represented by a node graph model $G_N=(V_N, S_N, R_N)$, where V_N and S_N are sets of vertices and s-segments of G_N, respectively, and the geographical information such as node location or edge formation is kept in set R_N. In this paper, we assume that all the nodes are of tree structure.

3. Edge contiguous relation at even node

After constructing graph model G, the recovery of writing order can be seen as the problem of finding a smoothest path passing through each of the edges at least once. It is easy to see that exhaustive search of all the possible paths will lead to computational explosion. In order to reduce the complexity, our approach is implemented within the framework of *ECR*. ECR is defined as a set of contiguous pairs of edges connected to a node. By the contiguous pair of edges, we mean the two edges that are written contiguously along the stroke at the node. Formally, for node N of degree n (Fig. 2),

$$ECR(N)=\{(e_i, e_j)\} \quad i, j\in \{1,2, ...,2n\}, \qquad (1)$$

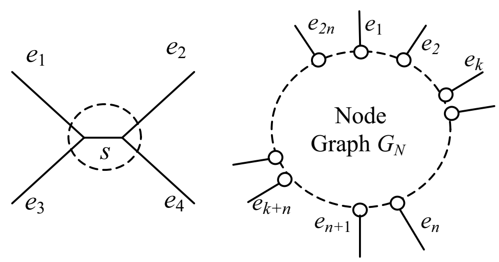
where (e_i, e_j) denotes a contiguous pair: a pair of two contiguous edges e_i and e_j. A particular contiguous pair $(e_t, NULL)$ is introduced, where e_t denotes a terminal edge and has no contiguous edge at N.

In the local phase, we find all of the possible ECRs at each of the nodes by examining their local configurations (remove the impossible ECRs at nodes), and in the global phase, we search for all the candidate paths based on the possible ECRs obtained, and select one that gives the smoothest path by evaluating global smoothness for each of the candidates.

In this section, we focus on the problem to obtain the possible ECRs at even node. Different from the tracing methods in the previous researches, which determined the contiguous pair of each edge separately, our method finds the possible ECRs at the node by discriminating between crossing and touching. For the node of degree 4 (*4-degree node*), a neural network is trained as a screen function, and for the node of degree 6 or greater, we propose a method in more theoretical or mathematical fashion by introducing 2 mild assumptions. The problem of obtaining the possible ECRs at odd node can be converted to that at even node after finding double-traced lines, which will be discussed in Section 4.

3.1 ECR at node of degree 4

4-degree node is composed of s-segment s and 4 edges e_1 to e_4 connected one another as shown in Fig. 2(a). There are 3 cases of ECR: $\{(e_1, e_4), (e_2, e_3)\}$, $\{(e_1, e_3), (e_2, e_4)\}$, and $\{(e_1, e_2), (e_3, e_4)\}$. The first ECR presents crossing type, where the two lines traversing through the node cross each other, and the other two ECRs give touching types, that is, the two lines traverse through the node without crossing.



(a) 4-Degree Node (b) Node of Degree 6 or Greater

Figure 2 Node Graph Model and ECR

We can obtain the ECR at node by detecting the touching or crossing type of node. The difficulty here is that local configurations of many touching nodes are very similar to that of the crossing nodes. This makes it difficult to discriminate between touching and crossing ECRs at some nodes by using only local features. Fortunately, we find that most 4-degree nodes are of crossing type (95.5% from our experimental examination). Take advantage of this a prior knowledge, we use a screening function to examine whether or not the touching ECR is possible at N. If so, we reserve all the three cases above as the possible ECRs, otherwise we keep only the crossing type as the possible ECR. A 3-layered neural network is trained for this task. The input includes 6 features: 4 tangent angles of e_1 to e_4, direction angle of s and length of s (if there exists no s-segment in N, the last two features are set equal to zero). Details of the structure and performances are described in [16]. The screening error rate was only 0.48% among all of the nodes tested.

3.2 ECR at node of degree 6 or greater

The problem of obtaining the ECRs at nodes of degree 6 or greater is much more complex. The complex structure of the node graph and the difficulty to collect enough training samples make it hard to employ the neural network or other learning based classifiers on these nodes.

Here a theoretical approach is introduced based on the analysis of interior structure of the node graph. Similar to the 4-degree node, we divide the nodes of degree 6 or greater into 2 groups: (1) crossing nodes, where all the lines that traverse through the node cross exactly once one another, and (2) touching nodes, where there exists at least one pair of touching lines.

For the crossing nodes, we can obtain the ECRs by the following *Crossing Node Traversing Rule (CNTR)*, Fig. 2(b),

CNTR: *For crossing node N of degree 2n, an edge contiguous to e_k (k=1,2,…,2n) is determined uniquely as n-th edge e_{k+n} counted from e_k either clockwise or counter-clockwise.*

We have the following theorem on CNTR:

Theorem 1: *CNTR is held if and only if N is a crossing node.*

ECR at crossing node N is expressed by:

$$ECR_x(N)=\{(e_k, e_{k+n})\} \quad k=1,2,\ldots,n \tag{2}$$

Not all the nodes are of crossing type, and the touching pair of lines must be possible as well. For the node of degree 6 or greater, the problem of finding the possible ECRs is turned to that of finding possible touching lines. Before describing our approach to this problem, we introduce two assumptions.

Assumption 1: *There exists at most one pair of touching lines at a node.*

This assumption is due to the smoothness constraint of human writing habit that the touching lines occur usually with higher curvature than the crossing lines. Under Assumption 1, there exists no other line between two touching lines traversing through the node, that is the two touching lines must locate adjacently each other.

Assumption 2: *Two touching lines have no common segment or vertex interior of a node through which both touching lines pass.*

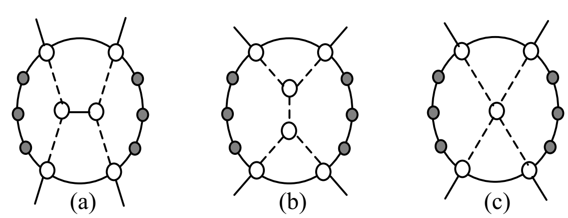


Figure 3 Configurations of the s-segments of two touching lines (dashed lines) inside a node

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

According to assumption 2, only case (a) in Fig. 3 is acceptable. Note that this paper focuses only on the node of tree structure. In order to find a possible touching pair of lines at the node that satisfies Assumption 1 and Assumption 2, we introduce *multiplicity* of s-segment and *parity* of vertex. *Multiplicity* $\rho(s)$ of interior s-segment s of node N of degree $2n$ is defined as the number of times by which s is passed along whole the stroke line under the assumption that N holds CNTR. Formally, given $ECR_x(N)=\{(e_k, e_{k+n})\}$, $\rho(s)$ can be calculated by:

find interior path $L_{k\text{-}k+n}$ which is the shortest path interior N between e_k and e_{k+n}, then,

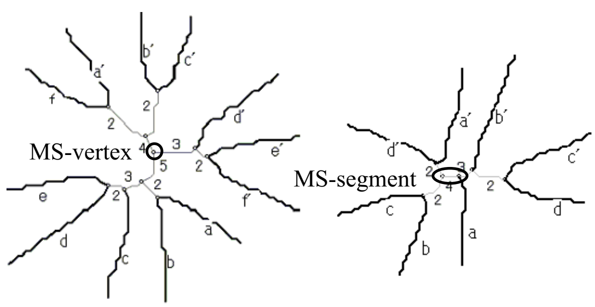
$$\rho(s)= \Sigma_k\, g(s, L_{k\text{-}k+n}) \quad k=1,2,\ldots, n \tag{3}$$

$$\text{where, } g(s, L_{k\text{-}k+n})=1 \quad \text{if } s\in L_{k\text{-}k+n},$$
$$=0 \quad \text{otherwise.}$$

Parity $p(v)$ of interior vertex v is defined as the sum of multiplicity of each of the s-segments connected to v:

$$p(v)= \Sigma_i \rho(s) \tag{4}$$

S-segment with multiplicity n is referred to *Maximum Spanning segment* (*MS-segment*). *Maximum Spanning vertex* (*MS-vertex*) is defined as a cut vertex



(a) MS-vertex (b) MS-segment

Figure 4 MS-segment and MS-vertex: Thick lines represent r-segments/edges and thin lines represent s-segments. The number attached to each s-segment is its multiplicity.

that has parity $2n$. Examples are shown in Fig. 4.

We have the following theorem on MS-segment and MS-vertex contained in the tree-structured node.

Theorem 2 *There exists either exact one MS-segment or exact one MS-vertex interior N of tree structure.*

Together with Assumption 1 and 2, we have the following useful theorem:

Theorem 3:

(1) If N contains exact one MS-vertex, N is of crossing type.

(2) Suppose that N contains one MS segment Σ,

(a) if a pair of touching lines traverses through N, then one of the two touching lines pass through one of the end vertices of Σ and another touching line pass through another end vertex,

(b) otherwise, N is of crossing type.

According to Theorem 3, the possible *ECR* can be obtained by executing the following procedures:

(1) If N has an MS-vertex, there is only one case of possible ECR: $ECR_x(N)$, given by Eq. (2).

(2) If N has an MS-segment, there are 2 possible *ECRs*. One is the $ECR_x(N)$, another has a pair of touching lines which can be found according to Theorem 3.2.a.

4. Edge contiguous relation at odd node

Different from the even node, the odd node must be connected by double-edge (*y-leg*) or terminal edge (*t-leg*). A node connected by a y-leg is referred to *y-node* and a node connected by a t-leg to *t-node* [12], [16]. To obtain ECR at odd node, it is necessary to identify the y-leg or t-leg at first. The problem to find the d/t-leg is related with double-traced lines (*d-lines*). A *d-line* must locate between two odd nodes, if assuming there is at most one y/t-leg incident to a node. There are two kinds of do- line: (1) *yt-type* between y-node and t-node, (2) *yy-type* between two y-nodes. In the single-stroked image, all the odd nodes must be connected by a d-line except start or end node. The difficulty to find the d-line comes from the fact that it is not easy to determine which edge connected to odd node is a d-line depending merely on its local structure. The robust decision should be made in more global fashion. In this paper, we propose a novel method to find d-line by applying *maximum weighted matching* in general graph.

Matching is a classical problem in graph theory. A maximum matching tries to find as many as possible independent edges in a graph, no two of which shares the same node. In the maximum weighted matching, the goal is to find the maximum matching with the minimum total weight. Gabow proposed an efficient algorithm of $O(n(m+n\log n))$ (n and m is the number of nodes and edges, respectively) for this problem [14]. In our problem, all the d-lines can be seen independent because we assume that there must exist exact one d-line connected to the odd node. This enables us to apply the maximum weighted matching to identify the d-lines. This procedure is described as following and the details can be found in [16]:

(1) Construction of matching graph G_c:

Construct graph $G_c=(O, C)$, where O is a set of odd nodes in G and C is a set of edges (actually candidate d-lines), which will be found as follows: For each pair of two odd nodes, we calculate the shortest path between them. If the shortest path is straight, it is regarded as a candidate d-line and put in C as an edge.

(2) Calculation of weight of edge in G_c:

Weight of edge e with two end nodes N_1 and N_2 is estimated as the tracing cost when e is traced twice. It is calculated as sum of the tracing costs at N_1 and N_2. We first examine whether e is a t-leg or a y-leg of N_i (i=1.2) by the method which we have proposed in [12]. If it is a y-leg, then the tracing cost at N_i is set as angle difference between e and its two contiguous edges, otherwise the cost is set as constant.

(3) Matching:

Employ Gabow's matching algorithm [14] on G_c to obtain all the independent d-lines by minimizing the total weight: $Cost_{sum}=\Sigma_k Cost(l_k)$. If there are two isolated nodes to which no matched line is connected, they must be either start or end node. Otherwise, we select the top-left odd node in G as a start node and remove it together with a connected line from G_c, then employ matching algorithm again. In experiments, such the case has been found very rare. Each of the matched lines corresponds to a double-traced line.

After identifying the t-leg or y-leg connected to the odd node in G, it becomes easy to obtain the possible ECRs at the odd nodes. In fact, by removing the t-leg or by duplicating the y-leg identified, the problem to obtain the possible ECRs at the odd node can be transferred to that at the even node explained in Section 3.

5. Global Tracing and Evaluation

After obtaining the possible ECRs at all of the nodes, we use global trace algorithm (GTA) to find all the possible paths from start to end node. After finding all the candidate paths, we calculate the smoothness for each of the candidates by using SPLINE approximation [4] [12] and then select the smoothest one as a final result. To improve the speed, the tracing and evaluation can be executed simultaneously [16].

6. Experiment and Conclusion

We have applied the proposed method to two sets of images: (1) one contains 147 scanned images collected by ourselves; (2) another contains 13306 offline images obtained by converting the online data in the Unipen database [15]. The stroke-width is set as 3 in conversion. By comparing the recovered path with the online data, we have observed that 95.2% are recovered correctly.

In this paper, a novel approach is proposed to recover the writing order from single stroke offline handwritten images, which combines an efficient local analysis and a robust global tracing. For further study, we have the following questions: 1) How to extend this method to multiple-stroke scripts; 2) How to develop the theory to obtain the ECRs at node of cycle structure (not tree).

References

[1] R.Plamondon and S.N.Srihari: "On-Line and Off-line Handwriting Recognition: A Comprehensive Survey", *IEEE Trans. on PAMI*, vol.22, No. 1, pp.63-84, 2000.

[2] S. Lee and J.C. Pan: "Offline Tracing and Representation of Signatures", *IEEE Trans. Systems, Man, and Cybernetics,* vol. 22, no. 4, pp. 755-771, 1992.

[3] K Liu, Y S Huang, and C Y Suen: "Robust Stroke Segmentation Method for Handwritten Chinese Character Recognition", *ICDAR,,* pp. 211-215,1997.

[4] T. Huang and M. Yasuhara: "A Total Stroke SLALOM Method for Searching for the Optimal Drawing Order of Off-Line Handwriting", *Proc. IEEE SMC.,* pp. 2789-2794, 1995.

[5] S. Jäger: "Recovering Writing Traces in Off-Line Handwriting Recognition: "Using a Global Optimization technique", *ICPR,* pp. 150-154, 1996.

[6] Y. Kato and M. Yasuhara: "Recovery of Drawing Order from Single-Stroke Handwriting Images", *IEEE Trans. on PAMI,* vol. 22, no. 9, pp. 938-949, 2000.

[7] v. Govindaraju and S. Srihari: "Separating Handwritten Text from Non-Textual Interference", *From Pixels to Features: II, J. Simon & S. Impedovo (editors), Elsevier Science Publishers, North-Holland,* pp. 17-28, 1992.

[8] Ke Liu, Yea S. Huang, Ching Y. Suen: "Identification of Fork Points on the Skeletons of Handwritten Chinese Characters" *IEEE Trans. on PAMI,* vol. 21, no. 10, pp. 1095-1100, 1999.

[9] D.S. Doermann and A. Rosenfeld, "Recovery of Temporal Information from Static Images of Handwriting", *Int'l J. Computer vision,* vol. 15, pp. 143-164, 1995.

[10] R. Plamondon and C.M. Privitera: "The Segmentation of Cursive Handwriting: An Approach Based on Off-Line Recovery of the Motor-Temporal Information", *IEEE Trans. on Image Processing,* vol. 8, no.1, pp.80-91, 1991.

[11] Lam. L, SW. Lee, CY. Suen: "Thinning Methodologies - A Comprehensive Survey", *IEEE Trans. on PAMI,* vol. 14, no. 9, pp.869-885, 1992.

[12] Y. Qiao and M. Yasuhara: "Recovering Dynamic Information from Static Handwritten Images", *IWFHR* Oct. 2004 Tokyo, Japan

[13] H.D. Chang and J.F. Wang: "Preclassification for Handwritten Chinese Character Recognition by a Peripheral Shape Coding Method", *Pattern Recognition,* vol. 26, no. 5, pp. 711-719, 1993.

[14] H.N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. *In Annual ACM-SIAM Symposium on discrete algorithms (SODA),* volume 1, pages 434-443, 1990.

[15] Guyon, I., Schomaker, L., Plamondon, R., Liberman, M. & Janet, S.: "UNIPEN project of on-line data exchange and recognizer Benchmarks", *Proceedings of 12th ICPR*, pp. 29-33, Jerusalem, Israel, October 1994.

[16] Y. Qiao, M. Nishiara and M. Yasuhara : "A Framework toward Restoration of Writing Order from Single-Stroked Handwriting Image" *IEEE Trans. on PAMI, Submitted*

IEEE COMPUTER SOCIETY