Segmentation of Handwritten Numerals by Graph Representation

Misako Suwa and Satoshi Naoi
FUJITSU LABS. LTD., 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki-shi, Japan
E-mail: suwa@jp.fujitsu.com

Abstract

A new algorithm is proposed for segmenting simply and multiply connected digits. It also removes ligatures. After thinning the pattern, the edges and vertices are extracted and the pattern is represented as a connected graph. Then the matrices relating to the graph are calculated. To determine the segmentation path, both graph theory techniques and heuristic rules are used. The boundaries of digits are calculated to make the width of touching strokes uniform. The separated digits thus have a more natural shape than can be achieved using algorithms that split patterns using straight lines or line segments.

1. Introduction

Approaches to segmenting handwritten numeral strings can be grouped into two types: those in which the number of digits is given and those in which it is not. The problem of segmenting touching numerals written in adjacent frames with dropout color can be reduced to that of segmenting a pair of touching numerals [see Fig. 1] and this approach belongs to the former type. While many algorithms have been developed for this type, it remains a challenging problem and was thus the focus of our research.

According to Chen and Wang [11], segmentation algorithms can be classified as foreground-based, background-based, or recognition-based. Foreground-based algorithms work with the foreground pixels in binary images. They include constructing a segmentation path from the highest point of the lower profile (and *vice versa*) [1], extracting significant contour points [2][3][4], deriving vertically oriented edges from adjacent strokes [6], judging the path from the distance between the minimum point of the upper contour and the maximum point of the lower one [7], and calculating a splitting line using discriminant analysis [8]. These algorithms are unstable, however, for "touching digits" that share a long segment or multiply connected digits. Background-based algorithms [9][10] first identify the background region and extract feature points from it or from background skeletons. A separation path is then identified to connect these feature points by a line. However, these features cannot be predicted for touching digits that share a long segment, are multiply connected, or have a ligature. The algorithm proposed by Chen and Wang [11] uses both background and foreground skeletons and can be used to segment multiply connected touching digits as well as simply connected ones, but the larger the degree of overlap along the x-axis of the two touching digits, the more unreliable the segmentation. Recognition-based techniques [12][13] use a recognizer to validate the segmentation process, but the accuracy of these techniques very much depends on that of the recognizer.

Most of these algorithms don't work on single segment touching and multiply touching digits and ligatures, simultaneously. In addition, all of them separate patterns by using a straight line or line segments. Therefore, the segmented patterns sometimes have unnatural shapes.

Recently, Elnagar *et al.* proposed an algorithm based on thinning [5]. The pattern is first normalized and thinned, and then the feature points are extracted. Segmentation points are identified near the decision line estimated from the deepest valley and highest hill. After the pattern is separated, it is restored using heuristics. Their method can remove ligatures; however, it cannot segment multiply touching patterns.

Using the same strategy as Elnagar *et al*., we propose a new algorithm for segmenting both multiply connected and simply connected digits as well as eliminating ligatures. A touching pattern is thinned and represented as a connected graph. Then each matrix relating to the graph is calculated. The segmentation path is determined by using both graph theory techniques and heuristic rules. The boundaries of the digits are calculated to make the touching strokes a uniform width. The separated digits therefore have a more natural shape than is produced by methods that use lines or line segments to split patterns.

In the next section, the touching types of pairs of digits are defined. Section 3 explains how a touching pattern can be represented as a connected graph. The proposed method is described in section 4, and section 5 describes experiments done to evaluate the performance of the method. The paper concludes with a summary and some remarks.

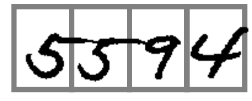


Fig. 1: Touching digits written in frames.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**

2. Types of touching

We defined the touching types of digits as follows:

(1) Simply connected
a. Single-point touching
Two patterns touch at a single location, i.e., they share one point. The end point of one stroke touches one side of another stroke, or the end points of two strokes are touching.
b. Single-segment touching
Two patterns touch at a single location, i.e., they share one stroke.
c. Single-point touching by ligature
Two patterns are connected by a useless stroke.

(2) Multiply connected
Two patterns touch at multiple locations. 'N-tuply' connected is defined as patterns that touch at N locations generating (N-1) loops.

Examples of these types are shown in Fig. 2.

(a) (b) (c) (d)

Fig. 2: Different types of touching digits: (a) single-point touching; (b) single-segment touching; (c) doubly connected (d) ligature

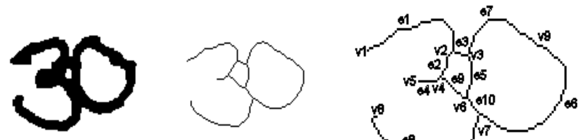
3. Graph representation of touching digits

3.1 Preprocessing

The first step in representing a touching pattern with a graph is to remove noise and smooth the image. The slant is normalized using the chain code of the contour [15].

3.2 Graph representation

Graph representation of character images using thinning has been widely used in stroke-extraction studies of Chinese characters [17][18] and we have used it to segment touching Japanese Kanji [19]. First, the images are thinned using Hilditch's algorithm [16]. Next, the vertices (nodes) of degrees 1, 3, and 4 (end points, T-joint points, and crossing points, respectively) are extracted by checking the eight-neighbors of each pixel. Then, every edge (branch) between two vertices is traced. If a significant turning point exists on an edge, a degree-2 vertex is added to it. Spurious edges that are less than the width of an average stroke and incident to a degree-1 vertex are eliminated, and the degrees of vertices incident to the edge are reduced to one. The graph representation of a touching pattern is depicted in Fig. 3.



(a) (b) (c)

Fig. 3: Graph representation of an image: (a) original pattern, (b) thinned pattern, (c) graph representation

3.3 Matrices relating to a graph

Before describing the proposed algorithm we will briefly explain the matrices relating to a graph. The notations and definitions are described elsewhere [20].

Let connected graph $G(V, E)$ be composed of vertex set $V(G)$ and edge set $E(G)$. Let v_i and v_j be elements of V, and e_j an element of E.

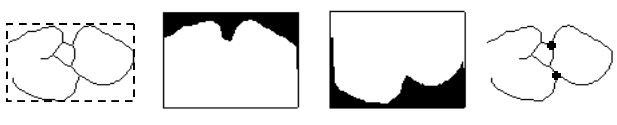
- **Adjacency matrix:** The rows and columns of the matrix correspond to the vertices of G and its element takes 1 or 0 in position (v_i, v_j) according to whether v_i and v_j are adjacent or not.
- **Circuit-based matrix:** There are ambiguities in the selection of a circuit base. We therefore select smaller circuits as a base. The rows correspond to the circuit base in G and the columns correspond to the edges of G. Its element takes 1 or 0 in position e_j of the i^{th} circuit-base $\boldsymbol{c}_i$, according to whether $\boldsymbol{c}_i$ contains an edge or not.
- **Proper edge cutset-base matrix:** If E_1 is a subset of E, and graph $G(V, E\text{-}E_1)$ is disconnected, then E_1 is called an edge cutset of G. The order of a cutset equals the number of edges it contains. The rows correspond to proper cutsets of G and the columns correspond to edges of G. Its element takes 1 or 0 in position e_j of the i^{th} cutset-base $\boldsymbol{k}_i$, according to whether $\boldsymbol{k}_i$ contains an edge or not.

4. Methodology

In our method, segmentation of touching characters requires obtaining two sub-graphs from a connected graph. We describe how to group the edges using graph theory techniques and heuristic rules

4.1 Calculation of upper valley and lower hill points

Touching points generally exist near the deepest point of an upper valley or the highest point of a lower hill. We calculate a vertical histogram from the top of the bounding box of the thinned pattern to the pixel of the upper region and one from the bottom of the bounding box to the pixel of the lower region. The valley of the upper histogram and the hill of the lower one nearest to the centerline are calculated. Then the deepest point p_t of the valley and the highest point p_b of the hill are determined. The vertices (v_t, v_b) nearest to these points are determined to be the segmentation vertices of the top and bottom parts.



(a) (b) (c) (d)

Fig. 4: Determination of segmentation vertices: (a) original graph and bounding box, (b) upper histogram, (c) lower histogram, (d) segmentation vertices of top and bottom parts.

4.2 Segmentation of single-point touches

If the vertices of the top and bottom parts (v_t, v_b) are equal or they are both end points of an edge, and they are incident with an edge e_j that is the element of an order 1 cutset $\boldsymbol{k}_a$, then the touching type of the pattern may be a single-point touch. An order 1 cutset is defined as one that contains a single element e_j, with the graph separating into two sub-graphs if it is removed from the graph. We call e_j a *touching edge*. To determine whether it is a single touch or not, the sizes of the bounding boxes of both sub-graphs g_l and g_r are calculated. If the overlapping degree O_x along the x-axis of g_l and g_r is less than its threshold O_{th}, then the touching type is determined to be a single-point touching. We defined O_x as follows:

$$O_x = l_{lr} / (x_{max} - x_{min} + 1),$$

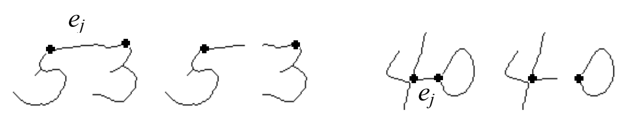
where l_{lr} is the width of the overlapping part of g_l and g_r and ($x_{\min}$, $x_{\max}$) are the minimum and the maximum values of x-coordinates of g_l and g_r, respectively. Members of the edges that belong to each sub-graph can be calculated from the new adjacency matrix computed from the original one. The e_j is classified to the left or right of both sub-graphs g_l and g_r depending on the overlapping degrees of e_j and these sub-graphs along the x-axis. These degrees, o_l and o_r, are defined by the following equation,

$$o_{l,r} = w_{l,r} / L,$$

where L is the length of e_j and $w_{l,r}$ is the length of the overlap with the edge and the left and right sub-graphs along the x-axis. Especially, if o_l and o_r satisfy the conditions:

$$(\min(o_l, o_r) / \max(o_l, o_r) > r_{th}) \cap (o_l + o_r > 0),$$

the touching type is regarded as a straight touch in a single-point touch (third pattern in Fig. 2 (a)). In this inequality, r_{th} means the threshold of the ratio. Then e_j is divided at p_t or p_b, whichever is nearer the center, and classified in both sub-graphs. However, if they do not fulfill these conditions, then e_j is classified in the sub-graph with the larger overlap.

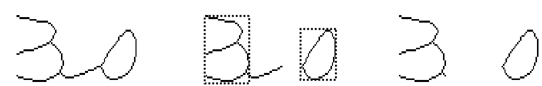


(a) (b)

Fig. 5: Grouping of edges for single-point touches: (a) e_j is divided and grouped in both sub-graphs, (b) e_j is grouped in the left sub-graph.

4.3 Ligature elimination

We used the method of ligature elimination proposed by Elnager *et al*. [5] because it is both simple and effective. If an edge sticks out from the bounding box calculated in 4.1, then the part is regarded as a ligature and is eliminated. This process is depicted in Fig. 6. Unnaturally long touching strokes in the above case are also regarded as ligatures.



(a) (b) (c)

Fig. 6: Ligature elimination: (a) original graph, (b) after grouping, (c) ligature eliminated from left sub-graph.

4.4 Segmentation of single-segment touches

If the segmentation vertices (v_u, v_l) are incident to an edge e_j, the orders of (v_u, v_l) are both greater than 2, and it is not a case of single-point touching, then it might be a case of single-segment touching. The angle θ between the y-axis and e_j, and the center of the mass-coordinates (x_{CM}, y_{CM}) of e_j are calculated. If θ and x_{CM} satisfy the following conditions, then this case is determined to be a single-segment touch:

$$(\theta < \theta_{th}) \cap (w / 3 < x_{CM} < 2w / 3),$$

where θ_{th} is the threshold of the angle and w is the width of the original graph. The origin of the coordinates is set to the upper left corner of the bounding box of the graph. In this case, θ_{th} is set at $\pi/4$ rad. In the case of a single-segment touch, e_j is duplicated and the graph is then separated into two sub-graphs. Each edge except e_j is classified in two groups, according to the new adjacency matrix for the disconnected graphs, which is computed from the original one. Each edge is grouped into left or right according to the x-coordinate of the center of its bounding box. The edge e_j belongs to both sides. An example of this case is shown in Fig. 7.

(a) (b)

Fig. 7: Duplication: (a) original graph, (b) after duplication and grouping.

4.5 Segmentation of multiply connected touches

When the segmentation vertices (v_u, v_l) do not satisfy the conditions described above, the path from v_u to v_l contains plural edges $\{e_j\}$, the touching pattern may be a multiply connected case. First, we calculate the shortest path from v_u to v_l using Dijkstra's algorithm [21]. Figure 8 shows a graph of multiply connected touching with a shrunken part. The black of (b) indicates the part of the shortest path. Next, the coordinates of the center of the bounding box of the segmentation-edge group and other edges are calculated. Each edge except $\{e_j\}$ is grouped left or right. For a N-tuply touching case, N-1 loops are generated near the center of the pattern. Based on this fact, we use the following heuristic rules to group the segmentation edges into sub-graphs:

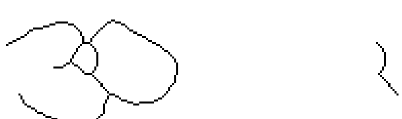
- If a segmentation edge e_j is a member of a circuit-base c_k, and the x-coordinate of the center of the circuit x_c satisfies

$$w / 3 < x_c < 2w / 3,$$

then e_j is grouped into the left or right; when e_j is located on the left/right side of c_k, it is grouped into either the left or right.

- If e_j does not satisfy the above condition, then it is duplicated and grouped into the left and right.

In the case shown in Fig. 9, the segmentation edges consist of $\{e_5, e_{10}\}$; e_{10} is duplicated and e_5 is grouped into the right.



(a) (b)

Fig. 8: Shortest path: (a) original graph, (b) the shortest path

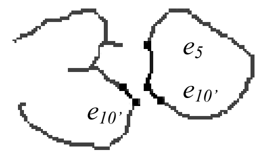


Fig. 9: Grouping of edges in multiply connected case.

4.6 Segmentation of character patterns

To decide the leftmost boundary of a right digit and the rightmost boundary of a left digit, we regard each black pixel of the sub-graphs as a source and calculate the Coulomb-like potential for each sub-graph. The potential threshold is not a fixed value but is adaptively calculated using the estimated stroke width of the digits; that is, it is determined that a stroke with the same width as the estimated value can be completely extracted. The result of the potential distribution of each sub-graph and the boundaries calculated using them are shown in Fig. 10.

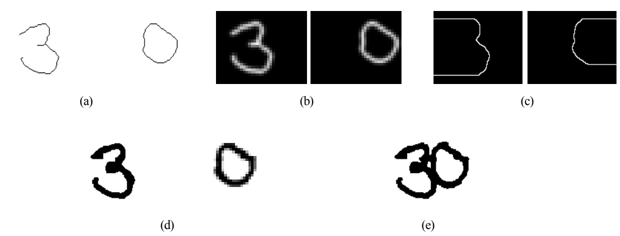


Fig. 10: Boundary calculation: (a) sub-graphs, (b) potential distribution, (c) boundary calculation, (d) segmentation, (e) original image.

5. Experimental results and discussion

We evaluated the performance of our method using touching images extracted from NIST 19 database. 2000 pairs of touching digits were used for test data and another 1000 pairs were used to constructing rules and determine the thresholds. Algorithm flow of the determination of touching type is shown in Fig. 11. Using this method, 89.2% of the patterns were correctly segmented. Table 1 shows the comparison with some algorithms. The segmentation rates were re-calculated to total 100% of the sum of the correct rate, error rate, and rejection rate. The quality of the segmented images seems to be improved in the case of touching digits that share a stroke. Some examples are shown in Fig. 12, and correctly segmented patterns and errors are shown in Fig. 13 (a) and (b).

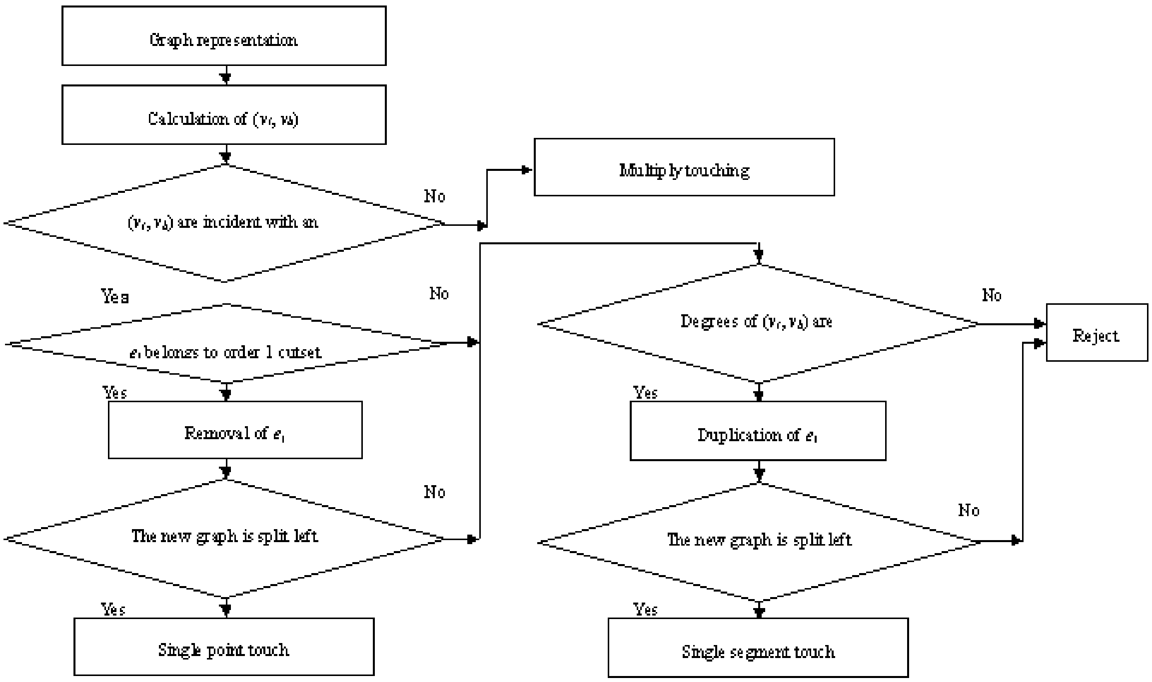


Fig. 11:Algorithm flow of the determination of touching type

Table. 1: Performance comparison

Algorithm	Correct rate	Error rate	Rejection rate	Database	Ligature/MC
[9]	80.80%	19.20%	-	Original data, 120 images	No/No
[10] High rejection	69.25%	2.15%	28.60%	NIST, 3355 images	No/No
[10] Low rejection	88.15%	7.15%	4.70%	NIST, 3355 images	No/No
[11] Outside test	88.06%	3.94%	8.00%	NIST+original data, 3238 images	Yes/Yes
Proposed	88.70%	8.85%	2.45%	NIST, 2000 images	Yes/Yes

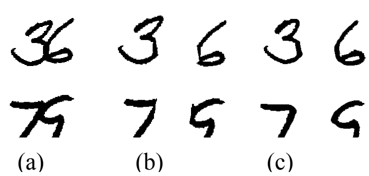


Fig. 12: Results of segmentation: (a) original pattern, (b) Method of ref. [11], (c) proposed method.

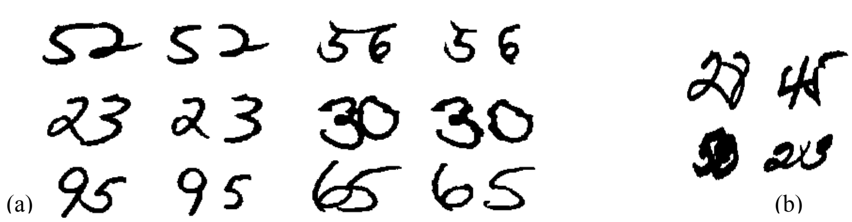


Fig. 12: Segmentation results: (a) correctly segmented patterns, (b) errors

0-7695-2187-8/04 $20.00 © 2004 **IEEE**

6. Conclusion and remarks

A new algorithm was proposed for segmenting simply and multiply connected digits. After thinning the binary pattern, the edges and vertices are extracted and the pattern is represented as a connected graph. Candidate segmentation paths are calculated using both graph theory techniques and heuristic rules. Ligatures are eliminated using the rules proposed by Elnagar *et al*. The boundaries of the digits are calculated to make the width of the touching strokes uniform. As a result, the separated digits have a more natural shape than can be produced by algorithms that split patterns using a straight line.

7. References

[1] M. Shridhar and A. Badreldin, "Recognition of Isolated and Simply Connected Handwritten Numerals", *Pattern Recognition*, **Vol. 19**, No. 1, pp. 1-12, (1986)

[2] N.W. Strathy, C.Y. Suen, and A. Krzyzak, "Segmentation of Handwritten Digits Using Contour Features", *Proc. of ICDAR '93*, pp. 577-580 (1993)

[3] Z. Shi and C. Govindaraju, "Segmentation and Recognition of Connected Handwritten Numeral Strings", *Pattern Recognition*, **Vol. 30**, No. 9, pp. 1501-1504 (1997)

[4] Z. Shi, S.N. Srihari, Y-C. Shin, and V. Ramanaprasad, "A System for Segmentation and Recognition of Totally Unconstrained Handwritten Numeral Strings", *Proc. of ICDAR '97*, pp. 455-458 (1997)

[5] A. Elnagar and R. Alhajj, "Segmentation of Connected Handwritten Numeral Strings", *Pattern Recognition*, **Vol. 36**, No. 3, pp. 625-634 (2003)

[6] J.M. Westall and M.S. Narasimha, "Vertex Directed Segmentation of Handwritten Numerals", *Pattern Recognition*, **Vol. 26**, No. 10, pp. 1473-1486 (1993)

[7] H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation Method for Character Recognition: From Segmentation to Document Structure Analysis", *Proc. of IEEE*, **Vol. 80**, No. 7, pp. 1079-1092 (1992)

[8] F. Kimura and M. Shridhar, "Recognition of Connected Numerals", *Proc. of ICDAR '91*, pp.731-739 (1991)

[9] M. Cheriet, Y.S. Huang, and C.Y. Suen, "Background Region-Based Algorithm for The Segmentation of Connected Digits", *Proc. of ICPR '92*, pp. 619-622 (1992)

[10] Z. Lu, Z. Chi, P. Siu, and P. Shi, "A Background- thinning-based Approach for Separating and Recognizing Connected Handwritten Digit Strings", *Pattern Recognition*, **Vol. 32**, No. 6, pp. 921-933 (1999)

[11] Y-K. Chen and J-F. Wang, "Segmentation of Single- or Multiple-Touching Handwritten Numeral String Using Background and Foreground Analysis", *IEEE Trans. on PAMI*, **Vol. 22**, No. 10, pp. 1304-1317 (2000).

[12] B. Zhao, H. Su, S. Xia, "A New Method for Segmenting Unconstrained Handwritten Numeral String", *Proc. of ICDAR '96*, pp. 524-527 (1997)

[13] N. Arica, F. Yarman-Vural, "A New Scheme for Offline Handwritten Connected Digit Recognition", *Proc. of ICPR '98*, pp. 127-129 (1998)

[14] R.G. Casey and E. Lecolinet, "A Survey of Method and Strategies in Character Segmentation", *IEEE Trans. on PAMI*, **Vol. 18**, No. 7, pp. 690-706 (1996)

[15] F. Kimura, M. Shridhar, and Z. Chen, "Improvements of A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words", *Proc. of ICDAR '93*, pp. 18-22 1993

[16] C.J. Hilditch, "Linear Skelton from Square Cupboards", *Machine Intelligence*, **Vol. 4**, pp. 403-420 (1969)

[17] C-C. Hsieh and H-J. Lee, "Off-line Recognition of Handwritten Chinese Characters", *Pattern Recognition*, **Vol. 25**, No. 11, pp. 1337-1352 (1992)

[18] D.S. Yeung and H.S. Fong, "A Fuzzy Substroke Extractor for Handwritten Chinese Characters", *Pattern Recognition*, **Vol. 29**, No. 4, pp. 1963-1980 (1996)

[19] M. Suwa, "Segmentation of Touching Handwritten Japanese Characters Using Graph Theory Method", *Proc. of SPIE*, **Vol. 4307**, pp. 280-289 (2001)

[20] V. Chachra, P.M. Chare and J.M. Moore, "Applications of Graph Theory Algorithms", *Elsevier North Holland* (1979)

[21] E.W. Dijkstra, "A Note on Two Problems in Connection with Graphs", *Numerical Math*., **Vo1. 1** (1959)

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**