Pergamon

Pattern Recognition, Vol. 30, No. 9, pp. 1501–1504, 1997

0031-3203/97 $17.00+.00

PII: S0031-3203(96)00118-5

SEGMENTATION AND RECOGNITION OF CONNECTED HANDWRITTEN NUMERAL STRINGS

ZHIXIN SHI[†] and VENU GOVINDARAJU[‡,*]

[†]Center of Excellence for Document, Analysis and Recognition (CEDAR), State University of New York at Buffalo, Buffalo, NY 14228, U.S.A.

[‡]Department of Computer Science, State University of New York at Buffalo, Buffalo, NY 14260, U.S.A.

(*Received* 10 *April* 1996; *in revised form* 15 *July* 1996; *received for publication* 2 *August* 1996)

Abstract—A new segmentation method for segmenting connected handwritten digit strings is presented. Unlike traditional methods where segmentation points are uniquely determined to cut the piece of stroke joining the connected numerals, our approach is one of identifying regions which serve as potential segmentation points. The regions are identified by a thorough analysis of the trajectory of strokes. © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

Segmentation   Digit recognition   Handwritten numerals   Contours

1. INTRODUCTION

Recognition of handwritten numerals is an important task in the field of document analysis and recognition as evidenced by its pivotal role in numerous applications. Postal addresses interpretation, determining the value of a bank check, reading tax forms and census forms are some examples. In each of these applications, handwriting is unconstrained, leading to frequent touching of adjacent numerals. Recognition of the touching numerals [Fig. 1(a)] must be necessarily preceded by a segmentation process that separates the individual digits.

Digit string segmentation problems encountered in document analysis are of two types. In the first type, the length of the digit string is unknown while in the second type, the length is a given. In both cases the task is to return subimages corresponding to each digit in the input string. The first type poses an additional subproblem of determining the number of the digits in the string (length). The research presented in this paper deal with the separation of pairs of touching digits, which is a problem of the second type.

Given an image containing a pair of touching or connected digits of adequate quality (sufficient resolution, good writing style, and contrast), humans can readily identify them. The idea is to find a (possibly crooked) line that can separate the image into two parts, each containing a single digit. Touching digits either touch at a point or are connected by a ligature which is a trailing stroke from the first digit to the second.(1) The task is one of identifying the touching point or the ligature. This is non-trivial mainly because of non-uniform stroke width. Thinning has been suggested(1) as a possible solution to achieve uniformly single-pixel wide strokes. However, thinning is computationally expensive, and introduces several artifacts.

In this paper we present a novel approach to the problem of segmentation. Our method involves the use of stroke information such as stroke direction, stroke turning points and end point information, derived from chaincode contour representation of digit images and identifying potential segmentation regions on the contour.

2. BACKGROUND

A survey of segmentation strategies is provided by Casey and Lecolinet.(2) Common approaches are often heuristic. While they have the advantage of efficiency, their accuracy is limited. Vertical histograms have been widely used for segmentation but are error-prone.

Another method uses the upper and lower profiles in conjunction with a set of heuristics to determine the segmentation points.(3) A segmentation path is constructed upward from the highest point on the lower profile of a numeral or downward from the lowest point in the upper profile. The approach is further refined in reference (4). A graph based method has been described in reference (5). This method analyses the contours of the connected numerals. Vertically oriented edges derived from adjacent strokes form the vertices of the graph, which are potential points of segmentation. There are other methods where a recognizer is used to aid in the segmentation process.(1) Potential segmentation points are validated by submitting the resulting segments to a digit recognizer. The recognizer provides a means of quantifying the "goodness" of a segmentation point, otherwise unavailable to methods using heuristics alone. However, frequent calls to a recognizer during the course of segmentation makes the process inefficient.

Stroke following is an attractive but difficult alternative. Although some work has been done in extracting

* Author to whom correspondence should be addressed.

(a) (b) (c)

Fig. 1. (a) Original image, (b) Chaincode representation with significant right turning contour points marked, (c) Segmented digits.

stroke information, its complete potential is still unexplored.

3. METHODOLOGY

A close look at the touching points and ligatures between two digits reveals that at each touching point, the chaincode contour makes significant right turns [Fig. 1(b)].

The chaincode element corresponding to a contour point is a structure which includes the coordinates of the contour point, the slope and the curvature. The structure is extended to contain the contour point type `type`, the opposite contour point (`ox`, `oy`) and modified opposite point (`new_ox`, `new_oy`) which will be used in estimating stroke width and rebuilding the images of segmented characters.

To determine the opposite point (`ox`, `oy`), we first compute an approximation of the norm vector `P_norm` at the current contour point $P_i(x, y)$. The first contour point encountered in a trace along `P_norm` starting from P_i is taken to be the opposite point P_i'.

As for determining the *significant* right turning contour points, we first compute vectors `P_in` leading in to P_i from contour point P_{i-k} and `P_out` going out of P_i to contour point P_{i+k}, where the offset k is determined by the estimated stroke width. These vectors are normalized and placed in a Cartesian coordinate system with `P_in` along the x-axis [Fig. 2(a)]. A threshold THR may then be selected such that any significant right turn satisfies the conditions:

$$x_1y_2 - x_2y_1 < 0, \quad (1)$$

$$x_1y_1 + x_2y_2 < \mathrm{THR}, \quad (2)$$

where equation (1) indicates that the turn is to the right and equation (2) indicates that the turn is significant. If we use normalized vectors `P_in` and `P_out`, the threshold THR [the x-coordinate of the thresholding line in Fig. 2(a)] can be determined experimentally to be a number close to zero. This ensures that the angle made by `P_in` and `P_out` is close to 90°—a significant turning angle.

Significant right turning points together with their opposite contour points located earlier divide the contour into contour pieces [Fig. 2(b)]. Contour pieces may now be classified as belonging to one or the other digit.

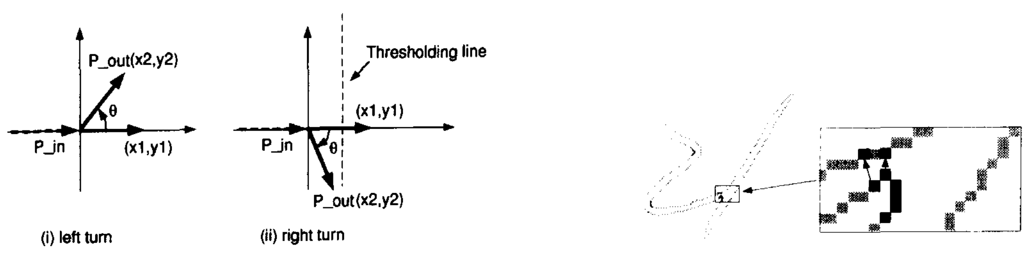
The simplest way of doing the classification is by using the vertical line bisecting the image as our guide. A contour piece is classified as part of the left digit if the *center of mass* of the piece is on the left of the line, otherwise, as part of the right digit. In most cases, the two digits have similar widths and the touching strokes lie close to the center line. However this is not always the case. The method described fails, for example, for an image containing touching digits 5 and 1.

We describe an algorithm for determining a more appropriate decision line for the general case. The digit strokes are assumed to be free of slant. Slant normalization[6] is performed for images with significant slant. Therefore a vertical decision line suffices for the purposes of classifying contour pieces.

We first compute the histogram of vertical extents where the vertical extent of the image at each column is defined as the distance between the uppermost black pixel and the lower most black pixel in that column. The touching point may be expected to lie near the valley of the histogram.

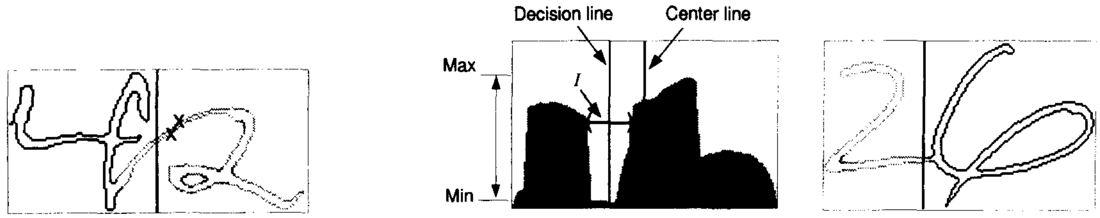
After we compute the histogram, we find the global maximum Max and the global minimum Min of the histogram. Let h=Max−Min. We draw horizontal lines in the histogram graph successively at the height of Min+$2h/3$, Min+$h/2$ and Min+$h/3$ and find a line that intersects the histogram at least twice. Each part of the line crossing a valley makes an interval on the line. We denote these intervals as I_i, $i = 1, \ldots$ [Fig. 3(b)].

We then find the intervals closest to the vertical bisecting line, and find the minimal point within the intervals. The one which is closest to the center line is taken as the position of the decision line (Fig. 3).



(a) (b)

Fig. 2. The distance betwen the thresholding line and the y-axis gives a threshold for determining a significant right turn. (b) The significant right turning points and their opposite contour points are marked.



(a) (b)

Fig. 3. (a) Simple vertical line as decision line of. The X's denote the center of mass of the two contour pieces crossing the decision line. (b) Histogram of vertical extents is used to find decision line, and the segmentation results.

Table 1. Results show correct rates for `zipseg` and the method described in this paper on images selected from BL, BHA and BU sets in CEDAR CD-ROM database

	Number of images	`zipseg`	Our system w/o recog	With recog
BL	249	71.5%	78.3%	79.5%
BHA	628	76.4%	79.8%	80.4%
BU	1089	75.9%	76.9%	80.8%

Each image contains two touching digits.

We use the decision line as a guide line to classify the contour segments as described earlier. Our segmentation algorithm allows us to classify entire strokes as belonging to one or the other digit, and results in a "cleaner" segmentation compared to traditional segmentation methods.

Images of segmented digits may be recovered from the segments by drawing line segments from each contour point to its opposite contour point, or the modified opposite contour point if the line segment to the opposite point is too long. The image can then be enhanced using a simple smoothing algorithm to fill in small gaps.

In fact, image recovery may be unnecessary since a chaincode based recognizer can directly use the separated contour segments for recognition.

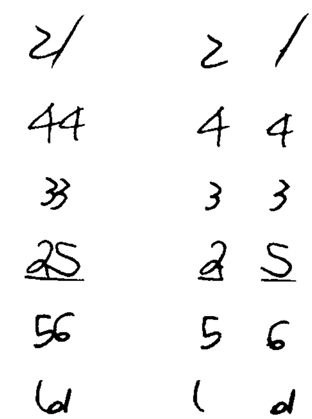


Fig. 4. Examples of segmentation results.

4. RESULTS

The segmentation algorithm proposed in this paper has been implemented and tested using images of touching digit pairs from the CEDAR CD-ROM database. For each input image containing a touching digit pair, the segmentation system returns two images corresponding to the segmented digits. The performance of the system is evaluated based on correct recognition of both digits.

The system may also be configured to give multiple segmentation candidates when the touching digits have a crossing ligature, and recognition can be used to select the best candidate. The performance of the system was compared with that of `zipseg`, a system developed for segmenting zipcodes.[7] zipseg internally uses a polynomial digit recognizer in making segmentation decision. Table 1 shows the segmentation results from (i) `zipseg`, (ii) our system without any recognition in decision making and (iii) our system with a chaincode recognizer is used in making segmentation decisions when ligatures are present (approximately 40% of total).

Some of the examples from the test set and the corresponding results of the segmentation algorithm are shown in Fig. 4. The unsuccessful case shown in the last row is due to the uneven size of the two digits.

5. CONCLUSION

This paper presents a new algorithm for the segmentation of handwritten digit strings. The experimental results demonstrate the effectiveness of the new algorithm in terms of segmentation and recognition.

Acknowledgements—We would like to thank Evelyn Kleinberg and Dar-Shyang Lee for the chaincode recognizer, Uma Mahadevan for helpful discussions and help in system implementation, and Sriganesh Madhvanath for his valuable comments.

REFERENCES

1. S. Seshadri and D. Sivakumar, A technique for segmenting handwritten digits, *Pre-Proc. of the Int. Workshop on Frontiers in Handwriting Recognition III*, Buffalo, New York, 25–27 May, 443–448 (1993).
2. R. G. Casey and E. Lecolinet, Strategies in character segmentation: A survey, *Proc. Int. Conf. on Document Analysis and Recognition*, Montreal, Canada, 1028–1033 (1995).
3. M. Shridhar and A. Badreldin, Recognition of isolated and simply connected handwritten numerals, *Pattern Recognition* **19**, 1–12 (1986).
4. F. Kimura and M. Shridhar, Segmentation-recognition algorithm for handwritten numeral strings, *Machine Vision Appl.* **5**, 199–210 (1992).
5. J. M. Westall and M. S. Narasimha, Vertex directed segmentation of handwritten numerals, *Pattern Recognition* **26**(10), 1473–1486 (1993).
6. G. Kim and V. Govindaraju, Handwritten word recognition for real-time applications, *Proc. Int. Conf. on Document Analysis and Recognition*, Montreal, Canada, 24–27 (1995).
7. R. Fenrich, Segmentation of automatically located handwritten numeric strings, *From Pixels to Features III: Frontiers in Handwriting Recognition*, S. Impedovo and J. C. Simon, eds, pp. 47–59. Elsevier, Amsterdam (1992).

About the Author — ZHIXIN SHI received his Ph.D. in Applied Mathematics and MS in Computer Science from the State University of New York at Buffalo in 1995. He is currently a Research Scientist at the Center of Excellence for Document Analysis and Recognition (CEDAR), State University of New York at Buffalo. His research interests include applied mathematics, computer vision, image processing, pattern recognition, information retrieval and artificial intelligence. He has been published in the areas of mathematics, image processing and handwritten numeral string segmentation and recognition.

About the Author — VENU GOVINDARAJU received his Ph.D. and MS in Computer Science from the State University of New York at Buffalo in 1988 and 1992, respectively. He received his B.Tech. (Hons.) in Computer Science from Indian Institute of Technology, Kharagpur, in 1986. He is currently the Associate Director of the Center of Excellence for Document Analysis and Recognition (CEDAR) and a Research Assistant Professor in the Department of Computer Science at SUNY Buffalo. His research interests include image processing, pattern recognition, information retrieval and multimedia technology.