Front. Electr. Electron. Eng. China 2007, 2(1): 39–44
DOI 10.1007/s11460-007-0007-x

RESEARCH ARTICLE

ZHANG Chuang, LIN Zhiqing, XIAO Bo, GUO Jun

The segmentation algorithm for handwritten numeral strings in bank-check recognition

Abstract In this paper, an integrated system of segmenting unconstrained handwritten numeral strings with unknowing number of digits is proposed, which consists of the extraction of connected components based on vertical projection and isolated components analysis, the length estimation of connected components using syntax analysis and waveform analysis and the segmentation of unconstrained connected handwritten numeral strings using innovative reverse "drop-falling" algorithm. This segmentation system which has promising results is then incorporated into a complete bank check character recognition system.

Keywords bank check recognition, segmentation, length estimation of digit string

1 Introduction

The recognition of unconstrained handwritten numeral strings is applied widely in office automation, bank check processing, zip code recognition system, etc.. Segmentation of connected handwritten digits is the main bottleneck among several key techniques of recognition system. In the past years, many efforts have been devoted to the improvement of segmentation strategies, which strongly affects the recognition capabilities of systems in real environments. Specifically, Chen Yi-kai et al. [1] combine background and foreground analysis to segment single-or multiple-touching handwritten numeral strings. Zhao Bin et al. [2] propose a segmentation method based on recognition which also combining with dissection method and holistic method; Congedo G. et al. [3,4] present a segmentation procedure useing a hypothesis-then-verification strategy in which multiple segmentation algorithms based on contiguous row partition work sequential on the binary image until an acceptable segmentation is obtained.

According to the targeted problems, these methods can be categorized into two classes, some for segmenting connected character strings with fixed numbers of characters [1] and the others with unfixed ones [2–4]. Both of them do not estimate the length of digit strings, which is helpful for the successful segmentation. In the recognition−based methods, the correct rate of segmentation depends too much on the robustness of recognizer and they are at the cost of the complexity of the algorithm.

In this paper, an integrated system of segmenting unconstrained handwritten numeral strings with unknowing number of digits will be proposed. This paper is organized as follows: in Sect. 2, the extraction of connected components will be discussed. We show how to estimate the length of digit strings based on syntax analysis in Sect. 3; we present innovative reverse "drop-falling" algorithm for segmentation in is present Sect. 4. In Sect. 5, an experimentation platform, called multiple method string segmentation toolbox (MMSSToolbox) is developed, which enables the user to quickly change test data and variables of "drop-falling" algorithms. In Sect. 6, we report experimental results and draw some conclusions.

2 Extracting connected components

When the characters of the digit string are well spaced and written with some degree of neatness, segmentation is a straightforward process (Fig. 1). The vertical projection of the numeral strings consists of a simple running count of the black pixel in each column. It can serve for the detection of white space between successive characters letters. In that ideal situation, one just has to pick out each connected

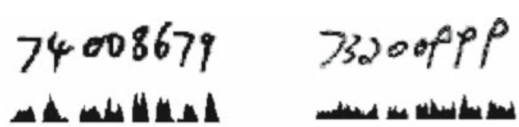

Fig. 1 Vertical projection and vertical histogram of number strings

Translated from *Journal of Beijing University of Posts and Telecommunications*, 2006, 29(1): 13–16 [译自: 北京邮电大学学报]

ZHANG Chuang (✉), LIN Zhiqing, XIAO Bo, GUO Jun
School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
E-mail: zcbupt@sina.com

component. The problem becomes much more difficult when characters are touching, overlapping, or disjointed (Fig. 2).

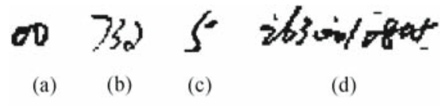

Fig. 2 The split-blocks after vertical histogram analyzing
(a) Touching; (b) Overlap; (c) Disjointed 5; (d) Touching, overlapping and noises

A connected component is defined as a set of black pixels where each is a direct neighbor of at least one other black pixel in the component. After the vertical histogram analyzing, the split-blocks can be easily found including some dispersive noises. The split-blocks will comprise dispersive noises, part of a character, a whole character, or two or more characters. Each isolated connected component in the split-blocks will be represented by a list of component structures, which holds the component's width and height, position of four borders, pixel area, black pixel count, and geometric center. If the isolated component is small enough to be qualified as noise (width and height smaller than 1/3 of the average width and height, or black pixel count smaller than 1/3 of the average count, or the geometric center far from the center of split-blocks), it will be removed.

Most common occurrence of a numerical character being broken into two isolated components is during the time when the top stroke of a "5" is disjointed from the body of the numeral (Fig. 3(a)). Our segmentor uses the size and position information of the connected components to make a decision that whether a given connected component is a disjointed top stroke of "5". We can judge a connected component to be the disjointed top stroke of a "5" if the following criterions are met:

1) The aspect ratio of the connected component is greater than 1.

2) The height of the component is less than 1/3 the height of the previous character.

3) The component is in the upper half of the image.

4) The distance between the position of geometric center and the up/down border is less than 3.

When the disjoined top stroke of a "5" was detected, move the smaller region to the left component to merge them as a whole region (Fig. 3(b)).

After connected component extracting, dispersive noises removed and disjointed region merged, there will only be the isolated connected component which is made up of one or more characters, and then how many digit strings will compose the isolated connected component should be estimated.

Fig. 3 Disjointed "5" detected and merged
(a) Example of a "5" with a disjointed top stroke; (b) Disjointed top stroke merged

3 Length estimation

Most isolated components are composed of one character. The others are composed of two or more characters. Accurate length estimation is very helpful for the successful segmentation of the connected numeral strings. The ratio of a connected component's bounding box height to its width is the most common indicator of what fraction of characters the component is composed of, but it's not enough to determine how many characters make up the component correctly.

Because the valley region in the upper contour of handwritten "4" and "0" will disturb the upper contour distribution of connected components (Fig. 4), we estimate that the wave crest in the lower contour of connected components is based on syntactical recognition instead of difference operator, which one's coding chain is too long and less described of the concavity and convexity features in the wave. How many characters will compose the components can be determined based on the number of wave crest.

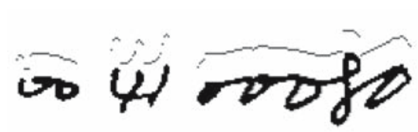

Fig. 4 The unstable upper contour distribution disturbed by valley region in the upper contour of handwritten "4" and "0"

Because of the discontinuity contour of the components, we smooth the low contour of the connected numeral strings. The smooth algorithm is described as follows:

$$\mathrm{dot}[i] = \left(\mathrm{dot}[i-1] + \mathrm{dot}[i] + \frac{\mathrm{dot}[i+1]}{3}\right) \tag{1}$$

Syntactic recognition is a good method to class the waveform [5]. We construct the syntax description of the waveform: suppose $w = w_1 w_2 \ldots w_n$, $w_i \in \{p, n, 0\}$ is the coding method of waveform, p stands for ascending stage, n stands for descending stage and 0 stands for the stage of neither ascending nor descending. In order to recognize the wave crest of the lower contour, a five elements deterministic finite automaton is defined in the following:

$$A = \{\Sigma, Q, \delta, z, F\} \tag{2}$$

where

$$\Sigma = \{n, p, 0\},\ Q = \{z, p_1, p_2\} \tag{3}$$

z is the initial state, $F = Q$ is the final state set; δ is the mapping from $Q \times \Sigma$ to Q.

$$\delta(z,n)=p_2,\ \delta(z,0)=z,\ \delta(z,p)=p_1,$$
$$\delta(p_1,n)=p_2,\ \delta(p_1,0)=p_1,\ \delta(p_1,p)=p_1,$$
$$\delta(p_2,n)=p_2,\ \delta(p_2,0)=p_2,\ \delta(p_2,0)=p_1, \qquad (4)$$

It is easy to recognize each crest and its initial point and final point by the deterministic finite automaton A: it is the position of the crest when $p_1 \to p_2$; it is the final point of current wave, and the initial point of next wave when $p_2 \to p_1$. It is necessary to identify the rationality of waveform, and eliminate the irrational crest caused by irregular handwriting (Fig. 5). We calculate each wave height and width according to the initial and final point, if one of the following constraints is satisfied, the wave is regarded as an irrational one and merged to the adjacent wave (Fig. 6).

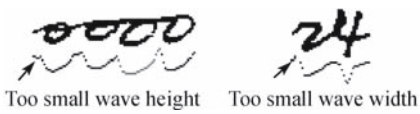

Fig. 5 Irrational waveform caused by irregular handwriting

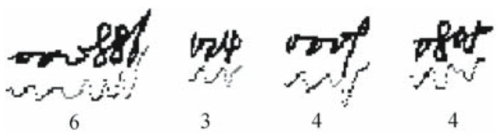

Fig. 6 The accurate length estimation based on syntax analysis and waveform analysis

1) The height of the crest is smaller than 3;

2) The width of the crest is less than 1/3 the average width of the rational crest;

3) The height difference of two adjacent crests is greater than 3/4 the height of the tallest one.

The initial point and final point of the each wave can be a reference border of segmentation.

4 Improved "drop-falling" algorithm

The basic idea of the "drop-falling" algorithm is to perform digit segmentation by simulating a "drop-falling" process. The main issue in its implementation is finding the starting point of drop-falling. There are four different algorithms of this type, up-left, up-right, down-left, down-right [3]. Salman [6] adopted hybrid "drop-falling" algorithms based on up-left and bottom-right which will miss the drop direction where there is the open region in the top of 4 and 0; Jibu [7] improved Salman's work and chose segmentation path based recognition which is time consuming and much depend on the recognition engine. We propose an innovative reverse "drop-falling" algorithm which is to detect the initials point by lower contour analyses and bottom-left /bottom-right next step searching strategy.

4.1 Reference point and initial point

In Sect. 3, we can mark the reference point of segmentation according to the wave crest in the lower contour. According to the reference point, the connected numeral strings will be segmented one by one, N numerals should be segmented $N-1$ times (Fig. 7). The initial point of drop-falling is very important since if the algorithm starts at the wrong place, the 'drop' could easily roll down the left side of the first digit or the right side of the next touching digit. We detect initial point according to the wave crest position in the lower contour and wave height.

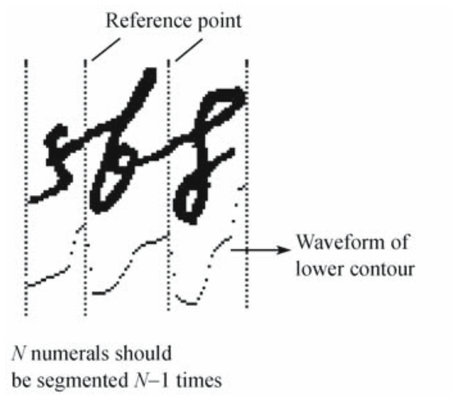

Fig. 7 Reference points and segmentation times

If the wave height of the left digit is less than the right one, the initial point is marked by scanning the horizontal line (left wave crest located) from left to right to detect the first white "∗" satisfying its left neighbor pixel is black (Fig. 8(a)), and if the wave height of the left digit is bigger than the right one, the initial point is marked by scanning the horizontal line (right wave crest located) from right to left to detect the first white "∗" satisfying its left neighbor pixel is black and a black pixel exists on the right of "∗" (Fig. 8(b)).

4.2 Next step searching algorithm

After finding the initial point, the algorithm follows the next step searching algorithm until the bottom of the image is reached. The directions of the drop falling are according to the current pixel position and its five adjacent pixels. The stepwise movement rules are depicted graphically in Fig. 9.

Where x is the abscissa of the current pixel position, and X is the abscissa of the reference point. If all the pixels from 1 to 5 are black, there will be the touching field of the connected numbers, and it's very easy to fall into the blind zone and lose the right way without any directions. We propose a "one of three" optimum searching algorithm, which means there are 3 directions, north, northeast, and northwest to choose when the "drop" falls into the blind zone (Fig. 10), and the optimum one is the one where segmentation path cross over the least stokes. If the numbers of punched stokes are the same, the priority for the falling direction is north,

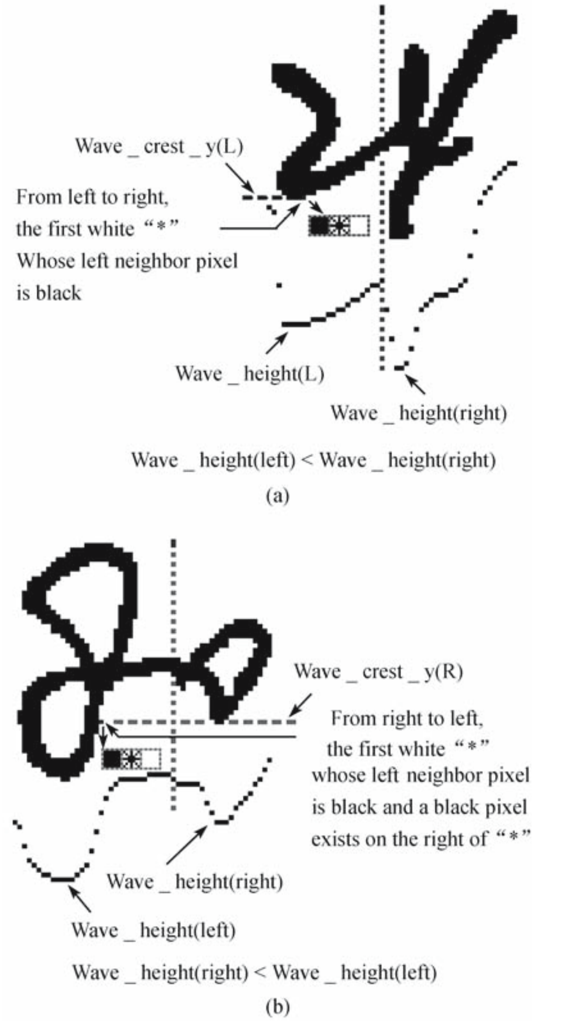

Fig. 8 Detection of the initial point.
(a) Wave_height(left) ⩽ Wave_height(right); (b) Wave_height(left) > Wave_height(right)

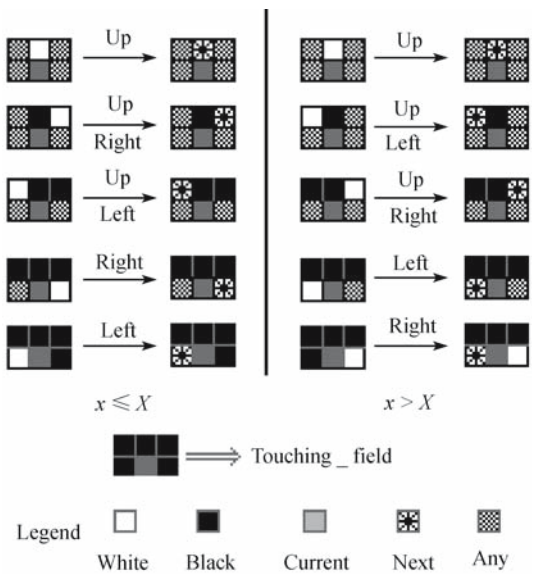

Fig. 9 Stepwise movement searching algorithm

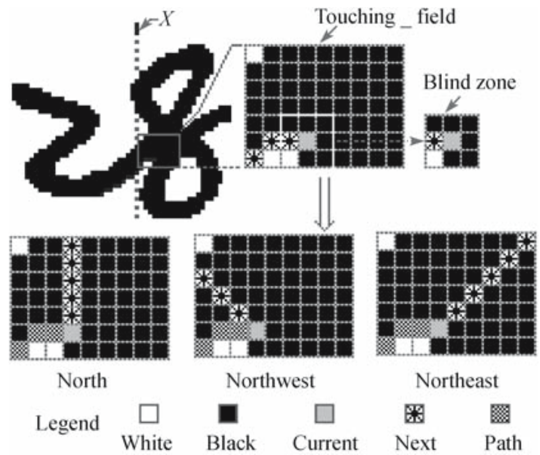

Fig. 10 Different segmentation paths when the “drop” falls into the blind zone

northwest, and northeast. For example, in Fig. 11, both the 2nd path and 3rd path cross over 1 stroke, according to the priority, the 2nd path which drop to northwest is the optimum one. Figure 12 shows the other two situations.

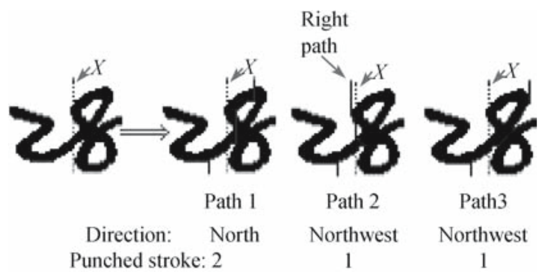

Fig. 11 Punched stroke number of different segmentation path

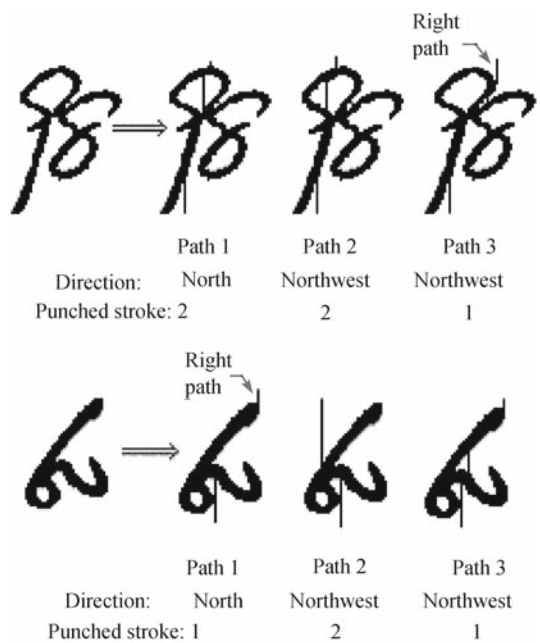

Fig. 12 The correct segmentation path selection based on optimum searching algorithm

5 MMSSToolbox

In order to facilitate the experimentation, a graphical MMSSToolbox was developed. The main purpose of building the toolbox was to provide an easy way to test various segmentation methods. So this tool enables the user to quickly change test data, display the test result, add on new routines, modify current routing, change the algorithms used for any step of the segmentation process, and easily incorporate the segmentation routines into another application.

The interface consists of a main menu, a controls area, and a display area (Fig. 13). Using the controller in this area, the user is able to step through various segmentation steps, vertical histogram analysis, connected components extracting, dispersive noises removal, length estimation of digit strings, improved “drop-falling” algorithm for segmentation and so on. Finally, the display area is where all of the segmentation data are shown to the user.

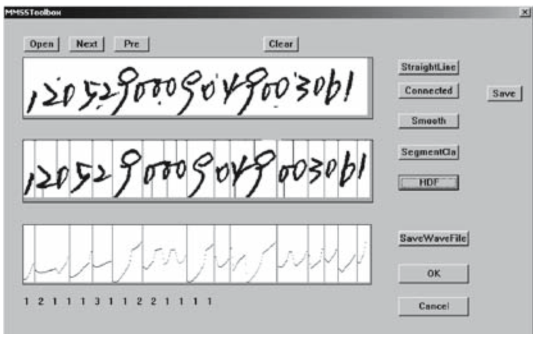

Fig. 13 Interface of MMSSToolbox1.0

6 Experiment and conclusions

The system was tested on the set account numbers of check images taken from a real business application (i.e. images of checks that were actually issued by China Bank of Construction). In order to gauge the accuracy of our segmentation algorithm, we must consider the performance on each segmentation step. The segmentation algorithm was tested by a set of 823 checks. But 16.8% of them (138 checks) are rejected, so 685 checks containing 5 873 digits are segmented. The results are presented in Table 1.

Some correct examples are shown in Fig. 14.

Table 1 Segmentation performance

	Correct	Error	Total
Connected extract	4 702	66	4 768
	98.6%	1.4%	
Length estimate	4 585	117	4 702
	97.5%	2.5%	
Improved drop−falling	730	33	763
	95.7%	4.3%	

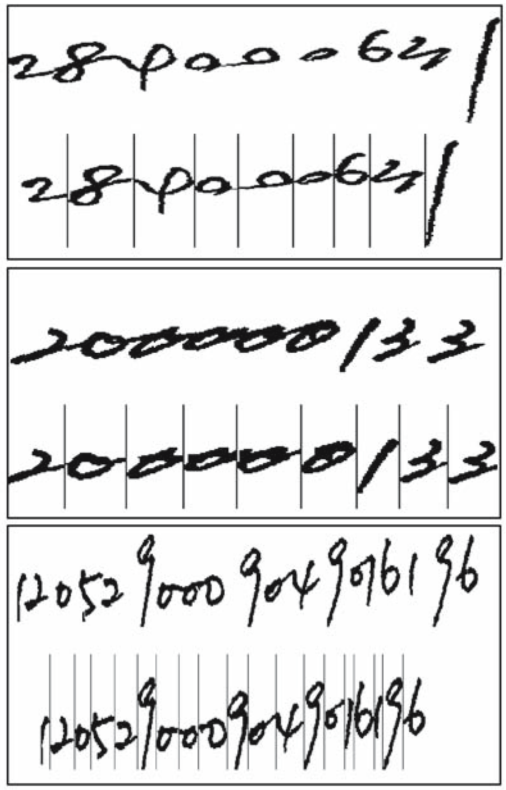

Fig. 14 Some correct examples of our segmentation algorithm

Some error examples are shown in Fig. 15, and the main reason of the mistake is non-typical features in the touching field of the connected components, so it’s necessary to integrate other effective methods into our MMSSToolbox.

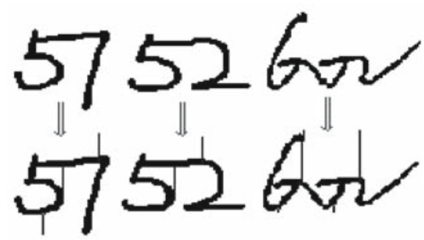

Fig. 15 Some error examples of our algorithm

In this paper, we study an innovative segmentation algorithm for unconstrained handwritten numeral strings. The improved performance of segmentation system has broader impact on improving the quality of automatic bank check recognition system. The significant experiment based on the actually bank check shows that our algorithm is more precise and robust for application.

References

1. Chen Y K, Wang J F. Segmentation of single-or multiple-touching handwritten numeral string using background and foreground analysis. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(11): 1304–1317

2. Zhao B, Su H, Xia S W. A new method for segmenting unconstrained handwritten numeral string. In: Proceedings of the Fourth International Conference on Document Analysis and Recognition. Los Alamitos: IEEE Computer Society Press, 1997, 2: 524–527
3. Congedo G, Dimauro G, Impedovo S, et al. Segmentation of numeric strings. In: Proceedings of the Fourth International Conference on Document Analysis and Recognition. Livres en anglais: IEEE Computer Society Press. Montreal: 1995, 2: 1038–1041
4. Dimauro G, Impedovo S, Pirlo G, et al. Automatic bank check processing: a new engineered system. International Journal of Pattern Recognition and Artificial Intelligence, 1997, 11(4): 467–504
5. Yan G H, Gu J L, Ge L C. Automatic classification of signal waveform useful in diagnosis. Journal of Northwestern Polytechnic University, 1996, 14: 294–298
6. Salman A K. Character segmentation heuristics for check amount. Supervisor Dr. Amar Gupta MIT 1998
7. Jibu Punnoose, An improved segmentation module for identification of handwritten numerals, Supervisor Dr. Amar Gupta MIT 1999