Graph-based Handwritten Digit String Recognition

Alexander Filatov, Alexander Gitis, Igor Kil

Offline Recognition Department
Paragraph International
1309 S.Mary Ave., #150 Sunnyvale, CA 94087 USA
E-mail: kil@paragraph.com

Abstract

This article presents a handwritten digit string recognition algorithm based on matching input subgraphs with prototype symbol graphs. The article defines a set of acceptable graph transformations corresponding to typical variations of the handwritten symbols. The search for a match between the input subgraph and prototype graph is conducted using this set of transformations. This approach allows to solve the problems of structure recognition methods caused by a high variability of handwritten symbol topology. The article presents experimental results of the handwritten digit string recognition system.

1. Introduction

The handwritten digit recognition is a key problem for such applications as automatic postal address and bank check reading. Such methods as statistical classifiers [1], neural networks [2],structural classifiers [3] can be mentioned among different approaches to handwritten digit recognition.

T.Pavlidis and J.Rocha in [4] use graph approach to describe symbol prototype. The input image is skeletonized, and the image skeleton is interpreted as graph, where nodes are breaking points of a pen trace (ending points), pen trace self-crossing points (branch points), points of strong curvature, inflection points and some other points described below, and edges are the skeleton sections between two nodes. Such an approach allows to solve problems of structural classifiers caused by broken symbols. For these purposes small gaps are eliminated by inserting additional edges, which can be either taken into account or ignored by the recognizer, depending on the prototype.

In [4], like in most works using graph for handwritten text recognition, graph is a result of skeletonization of the initial image (one of possible methods is described in [5]). In our approach we use another technique of graph construction, where nodes and edges of graph represent the input image information more completely (similar technique is used in [6]). First lines of approximately the same width - the pen width - are selected. Then all those lines are removed from the image and only their face segments are left. As a result branch and ending points are obtained which can be interpreted as graph nodes and the lines as graph edges. Such graph allows to determine line curvatures more accurately; besides it contains information about shapes and sizes of nodes as well as about edges widths.

The main difference between our approach and that proposed in [4] is as follows. We do not limit the correct matching of the input and prototype graphs by their isomorphism. An analysis of typical handwritten symbol variations for cursive (S.Guberman in [7]), handprinted and mixed writing enabled to construct a set of graph transformations corresponding to these variations. The transformations are used for input graph modification during matching process. This allows to cope with high variability of handwriting.

2. Overview of the algorithm

Image preprocessing includes binarization, normalization, and estimation of the text slope and pen width, as well as building the chain code of the image contour.

The recognition process begins with constructing a graph of the input image (Section 3). The next stage is segmentation of the digit string into single symbols. Obviously, it is not always possible to find the correct segmentation variant without using the recognition process. A possible solution to the problem is described in [8]. The main idea is that each new hypothesis about location of the next symbol is immediately examined by the recognizer. Thus, the segmentation and the recognition

0-8186-7128-9/95 $4.00 © 1995 IEEE

processes occur simultaneously. This technique though comprehensive, is rather time-taking. Our approach is different. A thorough analysis of the segmentation variants before the recognition allows to consider at the next stage only a small number of the variants (less than two per symbol on the average). After that the recognizer chooses the best alternative. The principles of segmentation are described in Section 4.

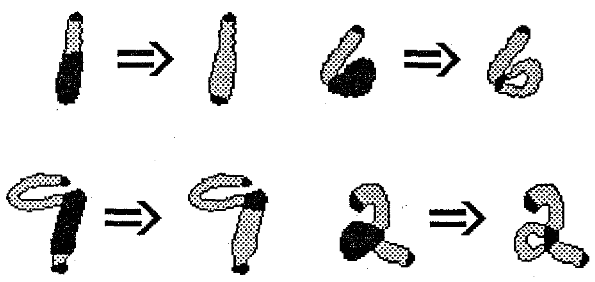
After the subgraphs of the input graph, supposedly matching single symbols, have been selected, each subgraph is compared with prototypes. The structure of the prototype and acceptable input graph transformations are described in Section 5.

3. Graph construction of the input image

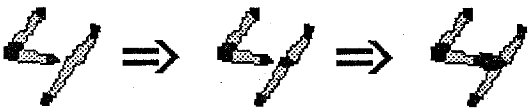
First the graph is constructed using a technique similar to that, described in [6]. The edges of the graph are the line segments of the same width as pen width, and its nodes are breaking and self-crossing points of pen trace as well as areas of thick pen trace (for example produced by forward-backward movements of pen). However, such a graph, is not always suitable for subsequent recognition. So a graph correction involving the following operations is called for:

- **Construction of second-degree node at sharp turns of the boundaries.** (If they have not been constructed before).

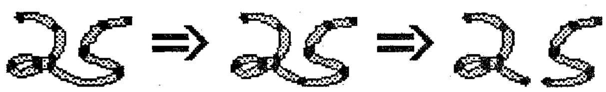
- **Transforming large nodes into edges or loops.**



- **Construction of second-degree node for subsequent gluing (depending on the symbol hypothesis).** A typical example is recognition of figure "4", written in two movements.



- **Dividing long horizontal edges with second-degree nodes.** If the long edge is a connection between symbols, construction of the node helps to separate the symbols correctly.



4. Text segmentation

We use the term "block" for a subgraph which is considered during the recognition process as a separate symbol. Some blocks are complex, i.e. they include several blocks each of which can be a single symbol as well. Text segmentation consists of the following stages.

- **Division of graph into subblocks at bridges.** By bridges we mean the nodes removing of which increases the number of connected components of graph.
- **Combining subblocks into blocks** is done using a set of heuristic rules, involving intervals between subblocks, distances between their centers, differences in their heights, average subblock size, etc. This procedure successively examines all pairs of neighboring subblocks and calculates estimations of their attachment. The most perspective subblocks are combined if the estimation exceeds a certain threshold. The procedure is repeated until the best estimation is less than the threshold.

Block attachment is termed conditional if the estimation of the most perspective pair does not exceed the threshold of unconditional combining. In conditional block attachment candidates for symbols are either block combination or each of its subblock.

5. Separate symbol recognition

The symbol prototype consists of a symbol graph and a description of its elements (geometrical edges characteristics, mutual position of edges and nodes, etc.). In the prototype, an obligatory subgraph is defined, that is the subgraph for each element of which a match in the input subgraph (block) must be found.

The recognition process consists in matching the input block and the prototype graphs by applying a set of transformations to the block. The match is considered to be found if there is an isomorphism between the block and a certain part of a prototype graph containing the obligatory subgraph. During the matching process several prototype graphs corresponding to one block may be found. The best match is chosen by estimating the geometrical parameters of the transformed block which are determined by the prototype.

5.1. Graph Transformations

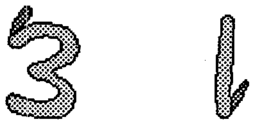
An analysis of the high variability of handwritten symbols shows that a comparatively small number of rules can be formulated to describe transformations of some handwriting elements into others, as well as their disappearance or emergence of extra elements. Such rules enable to considerably reduce a number of the prototypes necessary for describing all handwritten symbols. Thus our system uses 1.4 prototypes per symbol on the average.

In our opinion the main factors determining the variability of handwritten symbols are as follows:

- Some elements can change into others as a result of variation in correlation between vertical and horizontal speed. [7] introduces the sequence of handwriting elements, each of which can transform into both neighboring elements:

- Sometimes variability is caused by a necessity for returning pen to a trace point which has been passed before; This shows in open loops and small dashes appearing at line intersections.

- There are various ways to connect symbols; besides connections between symbols may be absent or be present only partly

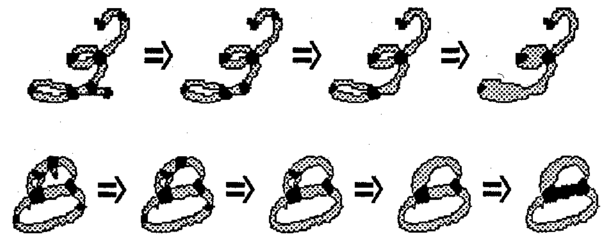


The above mentioned factors define the graph transformation set:

- transforming a loop to an edge
- combining two nearest nodes into one
- deleting thin edges
- deleting edges ending with a first degree node
- deleting second-degree nodes

Note that we simplify the graph, i.e. we suppose that the input subgraph initially contains the prototype graph of the written symbol (at least all the elements described as obligatory for the prototype graph). Thus our strategy is as follows: first to construct the most detailed graph (Section 3) and then to reduce it while matching it with the prototype graph until a one-to-one match is achieved between the input subgraph and the prototype graph (or its obligatory subgraph).

Matching of a block with a prototype begins with construction of the best variants of mapping of block nodes into prototype nodes. The criteria for the mapping quality are based on descriptions of the prototype nodes: their geometrical positions, degrees, presence of loops, etc. Each mapping variant defines division of block elements into two categories: the elements which have their pre-image in the prototype and the extra elements. The block reduction process consists in deleting the extra elements by successive transformations. Below some examples of block reduction process are given.



5.2 Estimation of a match between the input graph and the prototype graph

The estimate of a match between a subgraph which is a block reduction variant and the prototype graph is based on the descriptions of the prototype elements and is calculated as follows:

$$Y(B)= \Pi(1 - W_i{*}F(B,P_i)),$$
$$i=1,..,M,\ 0<= W_i <=1,\ 0<= F <=1.$$

Where :

B - is a set of physical features of the block,
M - is a number of descriptions for the prototype,
P_i - is a fuzzy interval for i-th prototype description,
W_i - is weight of i-th prototype description,
F - is a function measuring deviation of a physical feature from norm.

The descriptions of prototype elements can be divided into a few groups:

- **Node descriptions.** In each prototype, one node, termed the base node, is selected. Its position is determined by a fuzzy zone inside the rectangle containing a symbol. The other prototype nodes are described by specifying the direction and distance relative to a certain node already defined.
- **Edge descriptions.** Three types of edges are considered: stick, arc and loop. Sticks are described by the direction and distance from one node to the other. Arcs are described by the direction, curvature and length. Loops are determined by the direction from the node center to the loop center, and by its size.
- **Descriptions of relative object positions** are defined by the geometrical rules involving such features as distance between nodes and between nodes and edges, angles between the directions of edge ends, etc.

- Descriptions of penalties for graph transformations. Penalties for transformations are proportional to a ratio between the size of the transformed elements (length of the deleted edges, distance between combined nodes, etc.) and the vertical size of a symbol; but for some symbol zones penalties can be either increased or reduced.

6. Results

Based on the ideas described, a handwritten digit string recognition system was developed. We used special data base 1 US National Institute of Standards and Technology (NIST) to test the system. Test set for blind test consisted of 2000 black-white images , each containing a five-digit number. The number of digits was not known to the recognition system in advance. Approximately 25% of the images contained touching or broken digits.

Number recognition results for NIST database.

Table 1. Five-digit numbers. 2000 images.

Correct	**Error**	**Reject**
61.4%	**0.5%**	**38.1%**
64.3%	**1.0%**	**34.7%**
69.6%	**2.0%**	**28.4%**
83.1%	**16.9%**	**0.0%**

Table 2. Single digits from five-digit numbers. 10000 digits from 2000 images.

Correct	**Error**	**Reject**
90.4%	**0.5%**	**9.1%**
92.7%	**1.0%**	**6.3%**
94.3%	**2.0%**	**3.7%**
96.3%	**3.7%**	**0.0%**

Table 3. Previously separated single digits 10000 digits from 2000 images.

Correct	**Error**	**Reject**
93.1%	**0.5%**	**6.4%**
96.8%	**1.0%**	**2.2%**
98.2%	**1.8%**	**0.0%**

The last row in each table shows recognition results without reject. The difference between recognition results in Table 2 and Table 3 is caused by segmentation errors.

7. Conclusions

In this paper an algorithm of digit string recognition is proposed, based on matching input subgraphs with prototype symbol graphs. An advantage of this method is that it allows a purposeful modification of the input subgraph during its matching with the prototype graph. It is achieved by applying a set of transformations which correspond to typical handwriting variations to the input subgraph. This approach enables to use relatively few prototypes for a complete representation of handwritten symbols (e.g. our system uses less than two prototypes per symbol). Our approach helps to successfully deal with some problems caused by a high variability of handwritten symbol topology.

References

[1] J. Franke. On the Functional Classifier, Proc. of 1st Int. Conf. on Document Analysis and Recognition, pp.481-489, St.Malo, 1991.

[2] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, H.S. Baird. Constrained neural network for unconstrained handwritten digit recognition, Proc. of the International Workshop on Frontiers in Handwriting Recognition, pp.145-154, Montreal, 1990.

[3] L.Lam, S.Y.Suen. Structural Classification and Relaxation Matching of Totally Unconstrained Handwritten Zip-Code Numerals, Pattern Recognition, Vol.21, No.1, pp.19-31, 1988.

[4] J.Rocha, T.Pavlidis. A Solution to the Problem of Touching and Broken Characters, Proc. of the Second Int. Conf. on Document Analysis and Recognition, Tsukuba, 1993, pp.602-605.

[5] C.Arcelly Pattern Thinning by Contour Tracing, Computer Graphics and Image Processing, Vol.17, 1981, pp.130-144.

[6] J.-K.Simon. Off-Line Cursive Word Recognition, Proc. of the IEEE, 80(7), 1992, pp.1150-1160.

[7] S.Guberman. A Program for Reading Russian Handwritten Words, preprint, Inst. Appl. Mathem., the USSR Academy of Science, 1987, N199 (In Russian).

[8] S.Seshadri, D. Sivakumar. A Technique For Segmenting Handwritten Digits, Third Int. Workshop on Frontiers in Handwriting Recognition, Buffalo, 1993, pp.443-448.