TR-84-001

Structural Approach to Handwritten Numeral Recognition

By

KeRen Chuang and Jun S. Huang

Institute of Information Science
Academia Sinica,
Taipei, Taiwan, R.O.C.

March 1984

Abstract:

An efficient and reliable algorithm to recognize handwritten numerals is presented. Structural differences among numerals are investigated here, and features are extracted from these numerals, such as the order of interested points, number of endpoints, number of forkpoints, number of crosspoints, number of breakpoints, number of subpieces, the curvature of each subpiece, the orientation and the length of each subpiece, the properties of the starting point and the ending point of each subpiece. These features are used to classify numerals. A learning table is built in the recognition process to recognize special writings of numerals. An experiment is conducted for 400 different writing samples with no more than 3% failure.

1. Introduction

Handwritten numeral recognition has been widely studied by many authors. The methods used are Fourior Descriptor [1], statistical method [2-4], syntactical method [5-6], and polygonal approximation [7-8], etc. In our approach, we use structural method to solve this problem and it takes advantages of learning with building a dictionary that has a good nature to add new writings easily. There is a problem during implementation that is how to handle the effects of noises caused by confusing writings and thinning process on the picture. We use an extended chain code to reduce the effects of noises, and we treat all perturbated line segments caused by confusing writing as a line subpiece and do not drop them.

After the execution of automatic thresholding and thinning, the feature extraction is carried out. First we find all interested points such as endpoints, forkpoints, and crosspoints and then use extanded chain code to trace every pixel in the numeral by an particular order. We partition the whole numeral into subpieces by these interested points. Each subpiece's curvature and the breakpoints are also calculated during the tracing. At the same time the orientation and the length of each subpiece are also found. Also the properties of the starting point and the ending point of each subpiece are recorded. In the recognition process, we match these features with the contents of a built-in dictionary, and if an exact match occurs the input pattern is assigned to the matched numeral, otherwise a message is printed out. The whole block diagram of the approch is shown in Fig. 1.1.

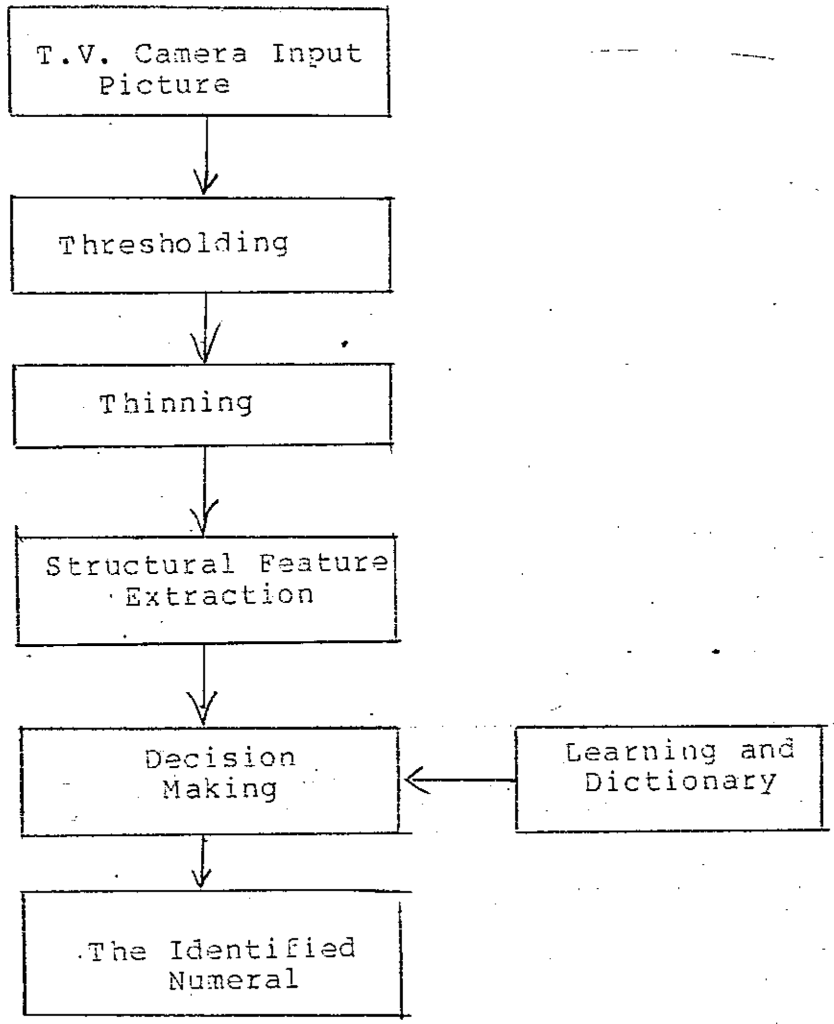

Fig. 1.1 Block diagram of the recognition process

2. Preprocessing

Once the picture has been taken, the first thing is to convert it into binary image by thresholding. we use a nonparametric and unsupervised method of automatic threshold selection for picture segmentation [9].

For the convenience of taking features from the binary image and reducing the noise effects a thinning process, developed by Huang and Lee [10], is used to convert the binary image into skeleton form and the trace of the whole skeleton is carried out.

3. Structural Feature Extraction

The structural feature extraction is the main process in our approach. We use the extanded chain code to compute the direction of change and to detect the breakpoint while tracing the skeleton. The conventional chain code has 8 directions in a 3x3 window. and the extended chain code here has 24 directions in 5x5 window as shown in Fig. 2.1(a)

19	20	22	24	1
18				2
16		x		4
14				6
13	12	10	8	7

(a)

				C
		B		
	A			

(b)

Fig. 2.1 (a) Extended chain code. (b) An example

In Fig. 2.1 (b) the direction from A to B has chain code 24 and the direction from B to C has a chain code 1. The difference between AB and BC is 24 - 1 = 23. If the difference is greater than 12, it is subtracted by 24; if smaller than -12 then added by 24. Hence the standard difference between AB and BC is 23-24=-1. If the sum of two successive difference in chain code tracing is greater than or equal to 6 (positive) or smaller than or equal -6 (negative) then it must be a breakpoint.

The size of extended chain code depends on the resolution. We used 7x7 window (24 direction, each direction has the length 3) once before for low resolution of numerals then too many breakpoints were detected. If we use 3x3 window, (8 directions, length 1) then we lose some breakpoints. Hence we choose the 5x5 window (with modified 24 directions length 2) for the resolution of 128x128 in the numeral picture. The structural features are :

(1) Interested points:

We classify the interested points into four catagories: end-point, forkpoint , crosspoint, and breakpoint, and calculate the following numbers :

a. Number of end points(endpoint).

b. Number of junctions of 3 branches(forkpoint).

c. Number of junctions of 4 branches(crosspoint).

d. Number of breakpoints, each breakpoint corresponding to a direction (positive or negative)

e. Order of endpoints, forkpoints, and crosspoints during the search.

(2) Subpieces

Use interested points mentioned above to partition numeral into subpieces. The starting point and the ending point of each subpiece are found and their corresponding catagories are recorded. The orientation and the length of each subpiece are also recorded, and the length is normalized after tracing the whole numeral. The orintation has 8 directions. That is, a circle is divided into 8 sectors, each with 45 degree, as shown in Fig. 3.1

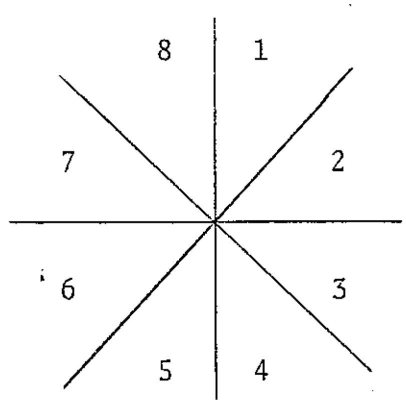

Fig. 3.1

Each subpiece has a curvature. A curve running clockwise is assigned a positive value, otherwise a negative value. We divide a circle into 24 grades. For example, a shape like alphabet " C " has a curvature between -22 and -18. Special shape like " S " has a curvature that always decreases and then increases. Similarly the special shape " $\mathcal{Z}$ " has a curvature that increases and then decreases. We assign these two special shapes with special codes " 97 " and " 99 ". respectively. If a subspiece is a circle or part of a circle and is the last traced one; then it has a curvature code " 98 ".

(3) The tracing process is described as follows :

First of all we find all the endpoints, forkpoints, and cosspoints along the direction left to right, up to down; thus we get an ordered sequence of endpoints, forkpoints, crosspoints. Secondly we choose an endpoint (left to right, up to down) as the starting point in the tracing process. If there is no endpoint left, we take a forkpoint. If there is no forkpoint left , we take a crosspoint as the starting point.

During tracing, we keep calculating the sum of the total chain code differences of subpiece and its length, until a breakpoint, or endpoint , or forkpoint, or crosspoint is met. This point is the ending point of this subpiece. After finishing the tracing, we assign the sum to be the curvature of this subpiece and we record the starting point and the ending point of each subpiece, and also the orientation and the length of each subpiece. The length is normalized at the end of tracing the whole numeral.

4. Decision Making

We build a dictionary which contains several numeral's features

: number of endpoints , number of forkpoints, number of crosspoints, number of breakpoints (including positive or negative), order of interested points, number of subpieces , curvatures of subpieces (upper bound and lower bound) catagories of the starting point and the ending point of each subpiece, and the upper bounds and the lower bounds of the length and the range of orientation of each subpiece. If an input numeral has the same feature values built in the dictionary , it has an exact match. If there is no exact match, then we print out a non-match message.

The match process is divided into two stages. The fist stage is to match the number of endpoints , forkpoints, crosspoints, breakpoints, and subpieces with the data in the dictionary exactly. The second stage is to match the order of interested points, the sign of every breakpoint, and subpieces. Each subpiece must satisfy the catagories of the starting point and ending point, the range of curvature, the range of orientation, the rang of length. These may not be required to have matched exactly. These depend on the data in the dictionary. Generally we'll meet one problem of skelton distortion. Thee proposed method here to solve this problem is to add the special feature of the trouble,tricky writing to the dictionary as a learning process which is efficient and necessary in most recognition processes.

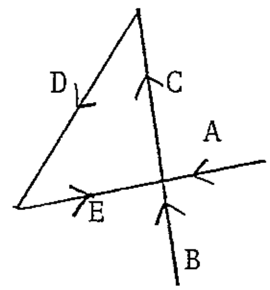

Fig. 4.1

In Fig. 4.1 the features of the numeral are :

1. endpoint : 2
2. forkpoint : Ø
3. crosspoint : 1
4. breakpoint : 2
5. subpiece : 5 , labelled as A, B, C, D, E
6. curvature : A,B,C,D are equal to Ø and E is equal to 98
7. orientation : A is in sector 2

   B is in sector 4

   C is in sector 4

   D is in sector 1

   E is in sector 6
8. length : all under 30 %
9. A : start with endpoint and end with crosspoint.

   B : start with endpoint and end with crosspoint.

   C : start with crosspoint and end with breakpoint.

   D : start with breakpoint, end with breakpoint, and the starting point and ending point are different points.

E : start with breakpoint and end with crosspoint.

The data in the dictionary are :

(1) points

(a) endpoint : 2

forkpoint : 0

crosspoint : 1

breakpoint : 2

(b) points order are : 1. don't care. 2.don't care.

3. endpoint.

(c) don't care the sign of breakpoints

(2) subpieces

(a) don't care the order of subpieces.

(b) there are 5 subpieces.

1. start with endpoint and end with crosspoint.

   curvature: between -3 and +3.

   orientation: between 2 and 3.

   length: between 40% and 0%.

2. start with endpoint and end with crosspoint.

   curvature: between -3 and 3.

   orientation: between 4 and 5.

   length: between 40% and 0%.

3. start with crosspoint and end with breakpoint.

   curvature: between -3 and 3.

   orientation: don't care (between 1 and 8).

   length: between 40% and 0%.

4. start with breakpoint and end with breakpoint the starting point and the ending point are different points.

curvature: between -3 and 3.

orientation: don't care.

length: between 40% and 0%.

5. start with breakpoint and end with crosspoint.

   curvature: between 98 and 98.

   orientation: don't care.

   length: between 40% and 0%.

5. Experiment

An experiment for the recognition of 400 written numerals with many tricky writings has been carried out. The first 200 writings are used for learning to construct the dictionary, and the last 200 writings are used for recognition. The input devices is Hamamatsu T.V. camera and the pictures are stored in PDP 11/70 system. The overall failure rate is about 3%. Some typical experimental results are shown in Fig. 5.1 , 5.2 and 5.3.

(a) (b)

Fig. 5.1 : (a) The source numeral.
(b) The numeral after thinning.

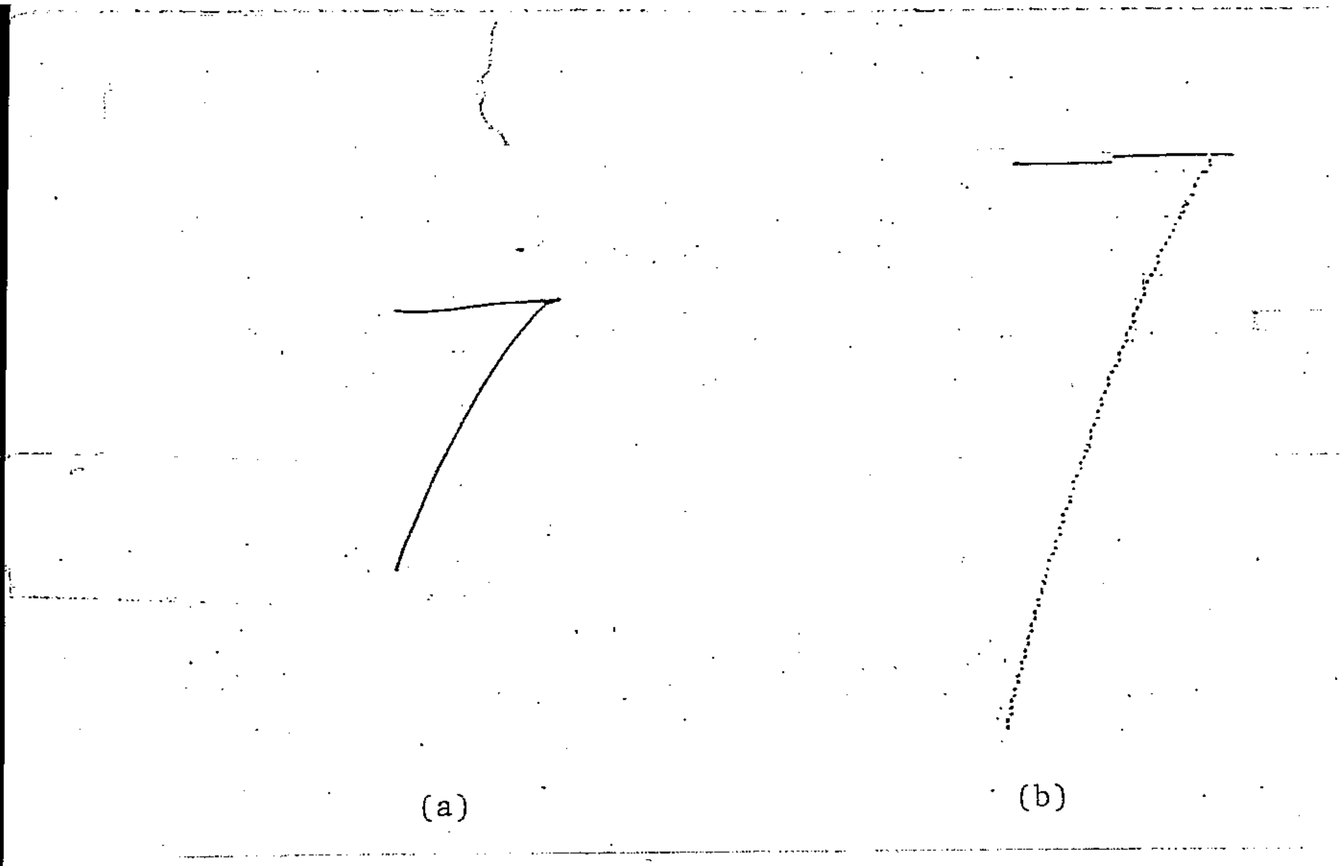

Fig. 5.2 (a) The source numeral.
(b) The numeral after thinning.

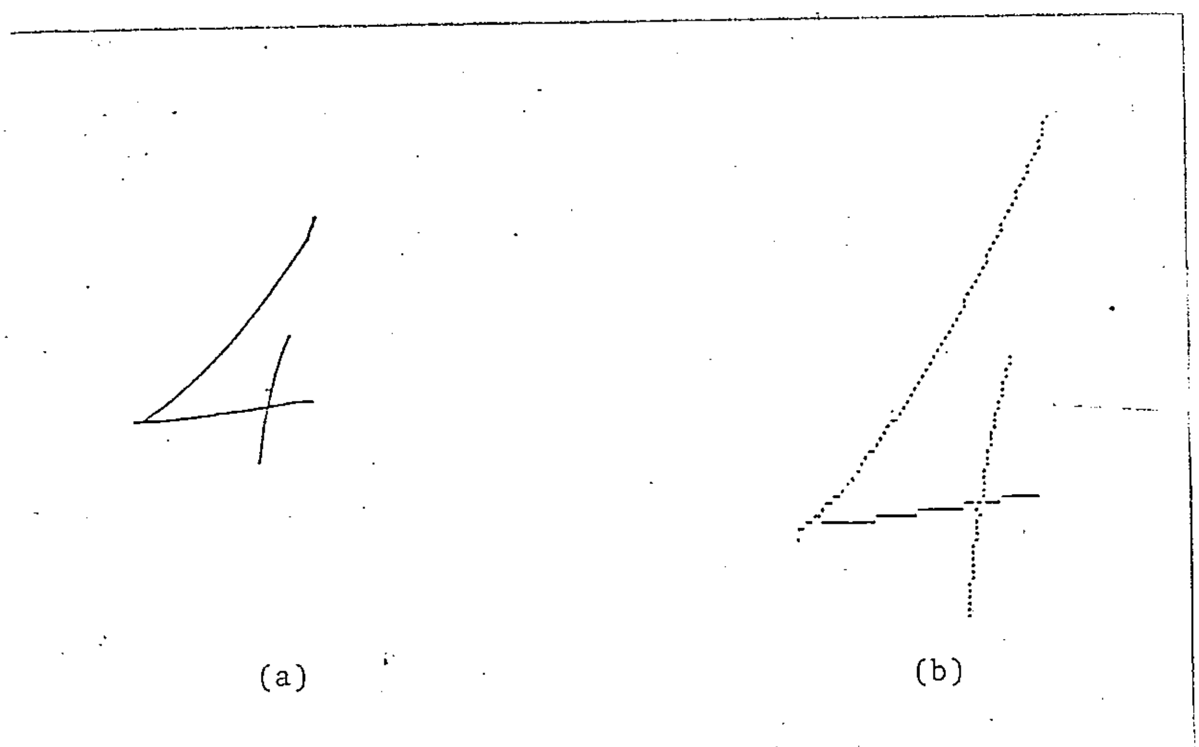

Fig. 5.3 (a) The source numeral.
(b) The numeral after thinning.

6. Discussion

Some special structural properties of numerals are not implemented here. There are (1) the detail study of the curvature of each subpiece. (2) relative position of endpoints, forkpoints, crosspoints, breakpoints. There are some particular features not mentioned here; however they may not be easily implemented.

Here in our dictionary, the order of subpieces of some numeral must be cared and some are not. The case of care or not care depends on the class of patterns to be recognized. If the variation in the pattern class is very high, the order must be cared, and which increase computing time and the size of the dictionary. The dictionary is easy to update; it can not only be used for matching handwritten numerals but also may aid for matching some special shape patterns. In fact, the dictionary is the kernel for simplifying the recognition process.

Reference:

[1] S. Impedovo, B Marangelli and,A.M. Fanelli, A Fourir discriptor set for recognizing Nonstylized numerals,by S., IEEE transactions on systems, man and cybernetics,vol.smc-8 no.8 Aug. 1978

[2] David, Quarmly and John Rastall, Experiments on Handwriten Numeral Classification, IEEE transactions on systems,man and cybernetis,vol smc-1 no. 4 Oct. 1971

[3] E. C.Greanias, P. F. Meagher, R. J. Norman,P. Essinger, The Recognition of Handwriten numerals by Contour Analysis, IBM Journal Jan. 1963

[4] Hu Chi-Hens, Lin peng, Ling Han-Yue, Wu Feng-Feng, A handwritten

numeral recognition machine for Automatic mail-sorting, Signal Proessing: Theories and Applications.

[5] B. Dueer, W. Haettich, H. Tropf and G. Winkler, A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals , Pattern recognition vol.12 pp. 189-199

[6] Hiroshi Genchi, Ken-Ichi Mori, Sadakazu Watanabe, Sumio Katsuragi, Recognition of handwritten numerical characters for automatic letter sorting, Proceedings of the IEEE vol.56 no.8 Aug.1968

[7] Theodosios Pavlidis Farhat ALI, Computer recognition of handwritten numerals by polygonal approximations, IEEE transactions on systems, man and cybernetics, vol.smc-5, no.6 Nov. 1975

[8] Forhat ALI, Theodosios Pavlidis, Syntactic recognition of handwritten numerals , IEEE transactions on systems, man and cybernetics vol. smc-7 no.7 July 1977

[9] Nobuyuki Otsu, A threshold selection method from Gray-lovel histograms, IEEE transaction on systems,man and cybernetis vol. smc-9 no.1 January 1979

[10] Jun S. Huang and Wen J. Lee, A new thinning algorithm : Removing Noisespurs and retaining endpoints, Proceedings of the National Computer Symposium 1983 Republic of China p.471