Computer description and recognition of printed Chinese characters

by WILLIAM STALLINGS
Honeywell Information Systems Inc.
Waltham, Massachusetts

INTRODUCTION

An approach to pattern recognition

An increasingly important aspect of computer pattern recognition research is automatic pattern description. Investigators have emphasized that pattern analysis should be basic to any pattern recognition scheme. Pattern analysis may be defined as the identification of elements of a structure and the description of the relationship among those elements. Chinese characters, by virtue of their regular structure, seem well suited for pattern recognition based upon pattern analysis.

With this point of view, a scheme for automatic pattern recognition has been developed which includes the following tasks:

1. Description. A systematic scheme for the description of the pictorial structure of the patterns to be recognized is developed.
2. Analysis. An algorithm is designed which analyzes the structure of the patterns, producing a representation of the structure conforming to the descriptive scheme.
3. Encoding. From the structural representation of a pattern, a code is generated which uniquely identifies the pattern.

This method has been applied to the recognition of Chinese characters. A program has been written which analyzes Chinese characters; the program produces a data structure which describes a character in terms of basic picture elements and the relationship among them. A procedure has been developed for generating a numeric code from the structural representation. Recognition is achieved by building up a dictionary matching characters with their codes; the code for any new instance of a character can then be looked up in the dictionary.

The methodology of pattern recognition still lacks any solid theoretical foundation. The author feels that progress in this area requires building up a body of effective pattern recognition techniques. It is hoped that the methods and techniques developed in this project on a specific class of patterns will be applicable to other pattern recognition problems.

Specifically, this project deals with a class of patterns which displays a rich structure developed from a small number of basic elements, all of which are relatively simple. The patterns are characterized by the fact that these elements may be combined in many complex ways. Other classes of patterns which fit this description will hopefully benefit from knowledge gained here.

A Chinese reading machine

Two obstacles have hindered the access of interested non-Chinese groups to the vast body of written Chinese produced each year. The first is the difficulty of the language itself. Chinese is very complex and takes so long to master that few Westerners ever learn it well. And second, of course, is the size of the printed output in Chinese. Manual translation is slow and tedious, and can never be relied on to handle more than a tiny fraction of the material.

To make available to Westerners the culture and technology of one-quarter of the human race, some form of automation must be introduced. A Chinese reading machine, which could scan printed Chinese and produce English output, would provide the most desirable means of improvement. Such a machine is a long way down the road, but individual steps which advance toward that goal are to be encouraged.

Considerable work has been done in the area of automatic translation of Chinese. These efforts have been only partially successful. Even if a good translation
device were available, however, the formidable problem of encoding Chinese characters for input would remain. One answer to the problem would be the development of a practical Chinese character recognition machine, toward which the effort of this project is directed. It is hoped that advances in this area would provide additional incentive for work in translation devices. On a more modest scale, a Chinese character recognition device could be used as a type of on-line dictionary to speed up the process of human translation. Even this limited application would be a welcome advance.

THE STRUCTURE OF CHINESE CHARACTERS

Chinese is a pictorial and symbolic language which differs markedly from written Western languages. The characters are of uniform dimension; they are generally square; they are not alphabetic.

The characters possess a great deal of structure and hence are well suited to the method of recognition outlined above. Many regularities of stroke configuration occur. Quite frequently, a character is simply a two-dimensional arrangement of two or more simpler characters. Nevertheless, the system is rich; strokes and collections of strokes are combined in many different ways to produce thousands of different character patterns.

Chinese characters consist of sets of connected strokes. Each stroke is roughly a vertical, horizontal, or diagonal straight line segment. Sets of connected strokes form units hereafter referred to as components. Each character consists of an arrangement of disjoint components. Figure 1 shows a character having three components.

The structure of a Chinese character may therefore be specified on two levels:

(1) A description of the internal structure of each component, and
(2) A description of the arrangement of components in two dimensions.

Components

Two questions needed to be answered when deciding how to describe the internal structure of a component:

(1) What class of objects should be considered as the basic picture element?

(2) What sort of structure should be used to describe the relationship between elements?

Three criteria were used in answering these questions:

(1) The structure should be easy to generate from the original pattern.
(2) It should be easy to generate a unique numeric code from the structure.
(3) The structure should represent the pattern in a natural manner.

A natural method of representing the internal structure of a component would be in terms of strokes. This indeed is the approach taken by several previous recognition schemes. These schemes make use of on-line input, in which strokes are drawn one at a time. The difficulty with this approach for printed characters is that strokes do overlap and are not easily isolated. Also, the description of the relationship between strokes becomes complex.

---

Figure 1—Character with three components

* For a discussion of a system for the description of Chinese characters in terms of strokes, see Fujimura and Kagaya. The authors are primarily interested in computer generation of Chinese characters.
It is more promising to describe components in terms of stroke segments. This can best be understood with reference to Figure 2. As can be seen, a component can be depicted as a graph. The branches of the graph correspond to segments of strokes. These segments are bounded by stroke intersections and ends of strokes.

It will be shown later that this representation satisfies criteria (1) and (2). That it satisfies criterion (3) is fairly clear.

**Characters**

The arrangement of components in two dimensions to form characters can be described using the concept of frame. Each character is viewed as occupying a hypothetical square. The segmentation of a character into components segments its square accordingly. The square, or frame $\square$, may be segmented in one of three ways: (a) East-West $\|$, (b) North-South $\|$, (c) Border-Interior $\square$. Each of these segmentations corresponds to a two-component character. For example, 犁 would be represented by (a), which decomposes the character into 犁 and . 犭 would be represented by (b). Finally, either partial or complete enclosure, such as 犭 and 犭 would be represented by (c). Frames for characters composed of more than two components are obtained by embedding (a), (b), or (c) in one of the sub-frames of (a), (b), or (c). The process of embedding is recursive, in that any sub-frame of a derived frame may be used for further embedding. For example, the four-component character of Figure 3 can be described by the frame arrangement of Figure 4a. The frame description can be conveniently represented by a tree, as indicated in Figure 4b.

This description of the arrangement of components is based on the work of Rankin, who introduced the concept of frame-embedding. The definition of component used here is slightly different from that of Rankin. Despite this, Rankin's claim that the three relations used in his scheme are sufficient to describe accurately virtually all characters seems to apply.
The program operates on a representation of one character at a time. The representation is in the form of a matrix whose entries have value zero or one corresponding to white or black on the original printed picture.

The device used to obtain the matrix is a flying spot scanner which measures the intensity of light reflected from an opaque sheet of paper at a number of points inside a small area or "window" on the sheet. If the intensity reflected relative to the incident intensity at a point is below a certain threshold, the value one is transmitted for that point (BLACK), otherwise a zero is transmitted (WHITE). The effect is that of placing a grid over the picture and making each square either all black or all white.

A program has been written which operates the scanner and stores the resulting matrix in core. The size of the matrix is 80×80 for a single character. However, both the window size and the sampling rate (the fineness of the grid) may be varied so that characters of varying size can be processed. Additionally, the threshold may be adjusted to achieve the best quality.

Certain functions of the program depend on the fact that there are no gaps or holes in any of the strokes. This is not always the case due to the quality of the printed input. Accordingly, a smoothing operation is performed to fill in the gaps. The resulting matrix is used as the data base for the program.

The digitized form of a character can be displayed on a CRT. Figures 1, 3, 5 and 6 are photographs of such displays.

ANALYSIS OF COMPONENTS

A program has been written to perform the analysis of components. For a given component, the output of the program is a connected graph in which branches correspond to stroke segments and nodes correspond to the endpoints of stroke segments.

The program is quite complex, and will be described in detail in a future paper. A brief outline of the procedure is given here.

The procedure begins by finding an arbitrary stroke segment of a component. This is done by scanning along various rows and columns of the matrix until a stroke is encountered (See Figure 5). This initial stroke segment is the first branch of the graph. Next, the two endpoints of this initial segment are found. This is done by crawling along the stroke in both directions. Crawling is accomplished by following along the boundary points of the stroke using a standard contour-tracing algorithm. The crawling halts either at an intersection of strokes, characterized by encountering a large black area, or at the end of a stroke. Both of these two conditions for halting correspond to a node of the graph being encountered.

Thus, one initial branch and its two nodes are found. The procedure continues in the same manner. For each of the two nodes, all segments leading from it are investigated (by crawling), finding more nodes. Seg-
ments leading from each new node are similarly investigated until the entire component has been covered. During the execution of this recursive procedure, the graph of the component is developed as each new node is encountered.

ANALYSIS OF CHARACTERS

The algorithm for analyzing the pictorial structure of a character is in two parts:

1. A collection of connected graphs is produced, one for each component.
2. The relationship among components is determined.

Finding all components

The program described in the previous section must be applied to all the components of a character. Some method must be used for finding each of the components and keeping track of those which have been analyzed.

To do this, the following procedure is employed. As a component is being analyzed, its outline is drawn on a separate pattern. That is, the contour points of a component are filled in on a new pattern as they are encountered during analysis. The auxiliary pattern contains, at any time, the outline of all the components of a character which have been processed.

After a component has been processed, a search is made for a stroke of a new component. If a stroke is found, its boundary points are checked against the auxiliary pattern to determine whether or not this stroke belongs to an already-analyzed component. In this way, new components may be found and analyzed. The procedure halts when no new strokes can be found. The result is to produce a collection of connected graphs.

Figure 6 shows the result of applying the algorithm to the character of Figure 3.

Constructing the frame

Representation of the frame description of a character is done conveniently by means of a tree. The root node of the tree has as its value one of the three relations indicating how the overall frame is broken into two sub-frames. The two sons represent the structure of the two sub-frames. Terminal elements correspond to components (see Figure 4b).

The method of obtaining such a tree will be briefly described. First, each component in the character is inscribed in a rectangle. This is easy to do since the coordinates of each node are known. The relationship between all possible pairs of components is determined by determining the relationship between their rectangles. The one of the three permitted relationships (East-West, North-South, Border-Interior) which most nearly approximates the true relationship is chosen. Then it is determined if one of the components has the same relation to all other components. This will usually be the case. If so, that component becomes one son of the root node of the tree; the value of the node is the appropriate relation; the other son is a tree representation developed for the remaining components. This sub-tree is determined in the same way.

If no single component is found, a more complicated procedure is used to determine if any two components have the same relation to all others, and so on.

ENCODING OF COMPONENTS

For recognition purposes, a procedure has been developed for generating a numeric code for each character. The first step in this procedure is the generation of a code for each component in a character.
Figure 6—Outline of a character

The code for a component is generated from its graph. To this end, the branches of a graph are labeled at each end. The label on a branch at a node indicates the direction or slope of that branch quantized into eight directions. An algorithm can then be specified for starting at a particular node of a graph and traversing all of its branches. The sequence of branch numbers encountered is the code produced. An example appears in Figure 7.

The algorithm obeys the following rules:

1. Start at the node in the upper left-hand corner of the graph. Exit by the branch with the lowest-valued label. Mark the exiting branch to indicate its having been taken, and write down the branch label.
2. Upon entering a node, check to see if it is being visited for the first time. If so, mark the entering branch to indicate this.

3. Upon leaving a node, if there are available unused directions other than along the first entering branch, choose the one among these with the lowest-valued label. Leave by the first entering branch only as a last resort. Mark the exiting branch to indicate its having been taken and write down the label on the branch.

Since at each node there are just as many exiting branches as entering branches, the procedure can only halt at the starting node. At the starting node, all exiting branches have been used (otherwise the procedure could have been continued), hence all entering branches have been used since there are just as many of these. The same reasoning can be applied to the second node that is visited. The first entering branch is from the starting node and this branch has been covered both ways. But this branch would only have been used for exit from the second node if all other exits had been exhausted. Therefore all branches at the second node have been covered both ways. In this manner, we find that the branches of all nodes visited have been traversed both ways. Since the graph is connected, this means that the whole graph has been covered.

All branches are traversed exactly once in each direction by this procedure, so all labels are picked up. The code consists of the branch labels in the graph written down in the order in which they are encountered.

This algorithm is based on a procedure for traversing graphs described in Ore.² While it is true that this scheme will always generate identical codes for identical (isomorphic with the same labels) graphs, the goal of generating a unique code for each character is not achieved. Two types of difficulties are encountered.
The first is that two characters might generate the same graph, hence the same code. Figure 8 gives an example of such a situation. This situation appears to be very rare, however, and it seems to be no great hindrance to make special cases of these characters.

The second difficulty is that two different graphs might generate the same code. Figure 9 illustrates this situation. The author has been unable to find any pair of characters whose graphs exhibit such a property. Again, this appears to present no great problem.

The method, then, will generate different codes for different characters with only a few exceptions. The algorithm is simple and its execution is quite rapid.

ENCODING OF CHARACTERS

The representation of a character is in the form of a tree. The nodes of the tree are binary relations; the terminal elements correspond to components. Considering the relations as binary operators, the tree can easily be flattened to prefix form. This is done by walking around the tree counterclockwise, starting from the root node, and picking up nodes and terminals the first time they are encountered. As is well-known, the string generated in such a fashion is unique; the tree can readily be reconstructed from it. To generate a numeric string, the following code can be used:

- 0→terminals (components)
- 1→left node
- 2→above node
- 3→surround node.

Figure 10 shows the generation of code from the tree of Figure 4.

We can consider that the code so generated defines a class of Chinese characters all of which have the same frame description. Therefore, a Chinese character may be specified by first giving its frame description code and then giving the code for each of the components.
that fits into one of the sub-frames. A character having $n$ components will have a code consisting of the concatenation of $n+1$ numbers:

$$N_0, N_1, \ldots, N_n$$

where $N_0$ is the code generated from the tree and $N_1$ through $N_n$ are the codes of the components listed according to the order in which the components were encountered in the tree flattening.

RESULTS

The algorithms discussed in this paper have been implemented as a computer program which recognizes printed Chinese characters.

Most of the program is written in FORTRAN. A considerable fraction, however, is written in assembly language. The assembly language routines augment the power of FORTRAN to permit list-processing operations and also to permit FORTRAN subroutines to be called recursively. The combination, while less than ideal, provides for the creation and manipulation of complex data structures in a fairly natural manner.

The program runs on a PDP-9 computer with a core memory of $32K$ words of 18 bits. The computer also has a large auxiliary disk and magnetic tape storage facilities.

The program has been tested with a number of characters from several different sources. The tests were designed to consider four questions:

1. How successful is the program in analyzing the structure of Chinese characters?
2. Does the program generate consistent codes for characters of the same font? That is, will two instances of the same character from the same source yield the same code?
3. Does the program work for characters from different sources?
4. Do factors such as character size and character complexity affect program performance?

Initial results were obtained from a set of characters obtained from a Taiwan printer. A sample of this set appears in Figure 11. To start, 225 different characters were processed. This was to provide a dictionary for later tests, and to test the pattern analysis capabilities of the program.

The results show a reasonable structural representation produced for about 94 percent of the characters. The failures were all due to a particular component not being analyzed; for all characters the relationship among components was correctly determined. The

Figure 11—Example of character set
generated characters were used. These were about double the size of the originals. Both were of about the same style. Fifty instances were taken from each source. The percentage of instances generating the same code as the corresponding character from the original set was 89 percent for the magazine source and 95 percent for the computer source. Discrepancies mostly had to do with stroke segments appearing at somewhat different angles and with strokes touching in one case but not the other.

CONCLUSIONS

Pattern description

A descriptive scheme for the structure of Chinese characters has been proposed and a program for computer analysis conforming to the scheme has been written. The description is on two levels: The internal structure of components, and the relationship among components.

The first level of description is straightforward: a connected part of a character is represented by a graph. This representation is adequate for the description of components; it is reasonable for the human percipient to think of components as graphs.

Analysis on this level works fairly well; difficulty is encountered with some complex characters. Some work has been done on modifying the described approach. The modification consists of “shrinking” a component to a skeleton and obtaining the graph from the skeleton. This procedure is sensitive to contour noise, and it seems that use of this method would result in many components generating several different graphs from different instances.

The second level of description is based on the work of Rankin. With the exception of a very few characters whose components do not fit neatly into the frame description, it is an effective means of describing the structure of Chinese characters in terms of components. The analysis program for this level has been successful for all characters tested.

Pattern recognition

It would be overly optimistic to claim that the results of this thesis prove the feasibility of a hardware Chinese character recognition device. The development of such a device would be an impressive accomplishment. Nevertheless, the author feels that this thesis points the way to such a device by providing a good method for encoding Chinese characters.

There are two separate, but related, problems associated with Chinese character recognition. The first is that complex characters may fail to be encoded, or may generate several codes. The second is that there would need to be a quite large dictionary of characters to handle most material. A college graduate in China is supposed to know about 5000 characters, which gives some indication of what would be needed. The problem of handling a large dictionary arises. This is especially true if many of the characters require more than one code.*

It is likely that a recognition device would be used to process material from mainland China. If this is true, several developments make things look brighter. They result from the desire of the Communist Government to simplify the language. The first is that the Government has recommended the general use of only 2000 characters. A list of these characters has been published. Publishers, being Government-controlled, are under instructions to stay within the total of 2000 as far as possible. Secondly, the Government has simplified a large number of characters. In 1956, a list of 515 simplified characters to be used in lieu of the original complex forms in all publications was released. The average number of strokes per character for the 515 was reduced from 16 to 8. Overall, since 1952 the average number of strokes per character has been reduced from around 13 to around 10 for the 6000 most frequently used characters. The continuing Communist policy of language simplification contributes to the author’s opinion that a Chinese character recognition device is a realistic objective.

ACKNOWLEDGMENTS

The author would like to thank Professor Francis Lee of M.I.T., whose guidance and advice were invaluable to this project. The author is also grateful to Professor Thomas Huang of M.I.T. and Professor Herbert Teager of Boston University for their help. This paper is based on a thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical Engineering at the Massachusetts Institute of Technology on August 15, 1971.

* One way to reduce the number of characters requiring more than one code would be to use a stylized font especially designed for use with a character recognition device.
REFERENCES

1 CASEY NAGY
Recognition of printed Chinese characters
IEEE Transactions on Electronic Computers
February 1966 pp 91–101

2 Y CHU
A comparative study of language reforms in China and Japan
Skidmore College Faculty Research Lecture
Skidmore College Bulletin December 1969

3 FUJIMURA KAGAYA
Structural patterns of Chinese characters
Research Institute of Logopedics and Phoniatrics
U of Tokyo Annual Bulletin #3 April 1968–July 1969
pp 131–148

4 U GRENA NDER
A unified approach to pattern analysis
pp 175–210

5 GRONER HEAFNER ROBINSON
On-line computer classification of handprinted Chinese characters as a translation aid
IEEE Transactions on Electronic Computers
December 1966 pp 856–860

6 A V HERSHEY
Calligraphy for computers
U S Naval Weapons Laboratory 1 August 1967
AD 622 398

7 KING CHANG
Machine translation of Chinese
Scientific American June 1963 pp 124–135

8 J H LIU
Real time Chinese handwriting recognition machine
EE Thesis M. I. T. September 1966

9 O ORE
Theory of graphs
American Mathematical Society 1962

10 RANKIN TAN
Component combination and frame-embedding in Chinese character grammars
NBS Technical Note 492
February 1970

11 K M SAYRE
Recognition: A study in the philosophy of artificial intelligence
U of Notre Dame Press 1965

12 L YUTANG
Final report on Chinese-English machine translation research
IBM RADC-TDR-63-303 10 June 1963

From the collection of the Computer History Museum (www.computerhistory.org)