인쇄체 영문의 구문론적 인식

°박 동춘 * , 박 성한 **

* 금성 중앙연구소.

** 한양대학교 전자계산학과.

A CHARACTER RECOGNITION SYSTEM BASED ON SYNTACTIC APPROACH

Dong-choon Park * and Sung-Han Park **

* Gold Star Central Research Lab.
** Dept. of Computer Science & Engineering
Hanyang University.

Abstract: This paper proposes a new set of topological features (primitives) for use with a syntactic recognizer for high-accuracy recognition of printed alphanumeric characters. The recognition is accomplished on nine character groups, where each group has different combinations of four feature points. A skeleton enhancement eliminating isolated points and smoothing irregular points is developed. The tree automata processed in parallel enables the realization of high-recognition speeds and font-type independent recognition. The proposed character recongition system is tested for alphanumeric character fonts of dot matrix printer and plotter using IBM-PC/XT.

1. Introduction

The recognition of printed and handprinted characters has been one of earliest applications of pattern recognition methodologies[1]. In recent years, the field has developed to the point that practical character recognition systems are in use [2]. However, the problem of handprinted character recognition at a high recognition rate remains largely without practical solution. Thus the state of art of the character recognition consist of relatively good success in the recognition of constrained machine printed characters and relatively little success of the recognition of unconstrainted handprinted characters [3].

In the mean time, the number of fonts used with line printers, typewriters and printing machines is further increased by the modern thectniques of phototypesetting and raster printers. Thus multifont reading becomes an essential feature of modern reading machine. The size and composition of the character set is a fundamental parameter influencing system effort and performance for the every reading machine. The conventional character pattern recognition system by decision theoretic approach has high-recognition speed, but becomes very complicated for this multifont capability. In addition,the intented change to electronic media calls automated conversion of every text data from its graphic form into the coded form of electronic data processing as in the case of mixed mode communication [4]. On these back grounds, a syntactic appraoch is used to provide reading machine multifont capability without increasing cost of the machine.

The proposed system consists of preprosesor, feature point extracter, group classifier and recognizer. The binary pattern of alphanumeric character image is thinned by using the Safe Point Thinning Algorithm (SPTA) [5]. However, the thinned pattern includes noises and irregularities which make exact feature point extraction and primitive generation difficult. To circumvent this problem, we develop an algorithm to eliminate noise points and irregular points in the thinned pattern is developed. The algorithm deletes an isolated point neighboring main skeleton and smooths line segment having neighbor of 2. Using the thinned pattern obtained in preprocessor, four feature point- end point, break point, touch point and cross ponit are extracted according to the neighbor count of each point. The primitive pattern is then generated by coding the branch connecting these feature points into eight Freeman chain codes. Character patterns are then classified into nine groups depending upon combination of feature points. For example, characters classified as group 3 have two breaks and any number of cross points, and characters 0,8 and B belongs to group 3. To increase recognition rate, each character is allowed to be belonged to several groups. Based on these data structures of feature points and group classifier, input patern is represented in terms of tree grammer. Finally, recognition is preformed by tree automata. The tree automata is processed in parallel from each node of tree frontier to root through bottom up parsing. The proposed character recognition system is tested for 532 alphanumeric character fonts of dot matrix printer and plotter using IBM PC/XT.

2. Preprocessing

The first part of preprocessing stage consists of a thresholding operation followed by a self-point thinning algorithm to provide a skeleton of characters. Although the SPTA is a fast and effective skeletization algorithm, the thinned pattern shows two kinds of error point -noise point and irregular point,as shown in Fig.1 and Fig.2, respectively.

The noise point is defined as the isolated point which is attached to the main skeleton. The irregular points are the points which cause an irregularity between two neighboring points along one of eight chain code directions.

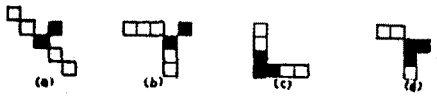


Fig. 1. Noise points

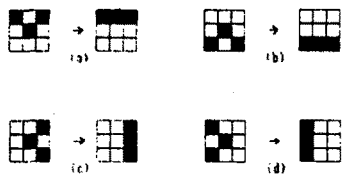
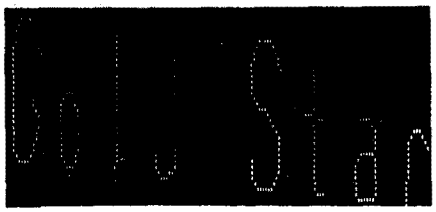


Fig. 2. Irregular points and pattern smoothing

To reduce incorrect feature points extraction and primitives, noise point elimination algorithm and irregular point smoothing algorithm are developed in the second part of the preprocessing stage. Fig.3 shows a thinned character pattern before and after these error correction algorithms are applied.

(a) Skeleton after SPTA



(b) Skeleton after error correction algorithm

Fig. 3. Thinned character patterns

3. Feature Point Extraction

Features representing local properties of characters are derived from the four feature points-end point,break point,touch point and cross point,which has neighbor count of 1,2,3 and 4, respectively, as shown in Fig. 4.

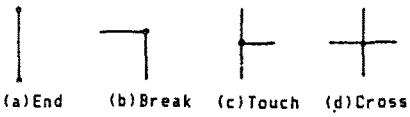


Fig. 4. Feature points

These feature points are determined depending on the connectivity between neighboring points along a direction of eight Freeman chain code of Fig. 5.

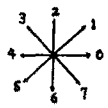


Fig. 5. Freeman Chain Code

Primitive pattern,to be used in a syntactic pattern classification mode, is then generated as the direction code value of a branch connecting each feature points. Table 1 is the data structure for feature points.

	end	break	touch	cross
total number of feature point	EC	BC	TC	CC
X-Y coordinate of feature point	EP (1:EC)	BP (1:BC)	TP (1:TC	CP (1:CC)
X-Y coordinate of neighboring feature point	ENB (1:EC)	BNB (1:BC:2)	TNB (1:TC:3)	CNB (1:CC,4)
direction connecting neighboring feature pointing	ED (1:EC)	BD (1:BD,2)	TD (1:TC,3)	CD (1:CC,4)

Table 1. Data Structure for Feature Point

4. Syntactic Recognition Algorithm

The first step in the syntactic recognition system is the determination of "groups" consisting of topological descriptors. The next step in this process is the construction of Tree Grammar pertaining to each group.

A. Determination of Groups

A group is defined as a set of characters characterizing a specific combination of feature points. Nine groups are defined for recognition and shown in Table 2.

	End	TOuch	Cross	Characters
Group 1	2	0	0	c,C,G,f,J,l,L,M,n,r,s,S,t, u,U,v,V,w,W,z,Z,1,2,3,5,7
Group 2	1	1	0	a,b,d,e,g,p,P,q,Q,4,6,9
Group 3	0	2	0	0,8,B
Group 4	0	0	0	D,o,O,0
Group 5	2	2	0	a,A,d,g,p,P,Q,R,4
Group 6	3	1	0	E,f,G,F,h,J,m,M,n,r,t,T,v, V,w,W,y,Y,1,3,4
Group 7	4	2	0	f,H,k,K,m,M,N,t,w,W,x,X,4
Group 8	4	0	0	i,j
Group 9	X	X	X	A,f,k,K,t,w,W,x,X,4,8

Table 2. Alphanumeric Character's group

To reduce the thinning effect in recognition, a redundancy is allowed for a character to be members of more than one group. For example, G are member of Group 1 and 6, because the short stroke of G can be deleted in preprocessing stage so that G is represented as $\mathfrak{G}$.

B. Production Rules

Since the structure of a character is defined by the composition of groups, a set of production rules is formulated using tree grammar [6]. Such a grammar is defined as

$$G_t = (V, r, P, s)$$

where $V = V_n \cup V_a$ is the grammar alphabet (nonterminal and terminal)
P(V, r) ranked alphabet
productions of the form $T_i \rightarrow T_j$ where T_i and T_j are subtrees
S(in V) starting tree

In the syntactic system, Freeman chain code defines eight primitives and is used as terminals of tree grammar. In the followings, the rules for the tree representation of each group are illustrated.

Group 1 Tree : .the rank of tree at each node except frontier = 1
.root($) corresponds to the upper end point of two end points

Group 2 Tree : .root($) is the touch point
.rank($) = 2
.one subtree is from touch point to end point and another subtree from touch point to touch point

Group 5 Tree : .root corresponds to the left touch point of two touch points
.subtree 1 is from root to end point, subtree 2 from root to root throuh the touch point which is not root and subtree 3 from root to the remaining end point
.rank($) = 3

Group 6 Tree : .root is the touch point
.each subtree is from root to each end point
.rank($) = 3

Group 7 Tree : .root corresponds to upper left touch point of two touch points which are neighboring
.there are four frontiers two of them are connected to root and other two to the another touch point

Group 8 Tree : .two component
.each component has one subtree

Group 3 and 4 need not to apply the tree grammar for recognition. That is, characters 8, B, and 0 can be recognized according to the three direction code between left touch point and neighboring feature points. characters o,O and D are also recognized by the direction between uppermost left touch point, bottommost left left touch point and their neighboring feature points. Figure 6 shows on tree structure of one character for each Groop.

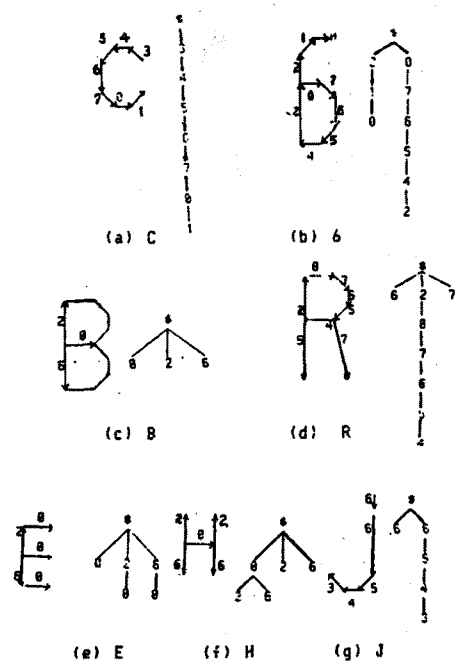


Fig. 6. Tree Structures of Characters

A frontier-to-root tree automation is a system

$$A_t = (Q, F, (f_a / a \text{ in }))$$

where Q is a finite set of states, F is a set of final states, a subset of Q, and fa is a relation in $Q^n * Q$ such that n is a rank of terminal a. A_t is a nondeterministic machine, parallel in nature, that represents the process of tree recognition [1]. A tree automation must begin simultaneously at each node on the frontier of the input tree and proceed along parallel paths toward the root. The group of primitives representing a test pattern is a matched against the state of tree grammar. An input tree is accepted by A_t if the automation can enter final state upon the encountring the root. Fig. 7 illustrates the producton rules for chraracter E, which is a number of group 6, where V = (A1, A2, A3,A4, 0, 1, 2, 6, 7, $), r($)=3, r(0)=0, r(2)=1, r(6)=1, r(7)=1.

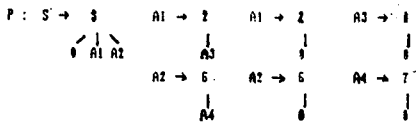


Fig. 7. production rules for the character E.

The automation for this character E becomes

A_E = (Q, F, (f0, f1, f2, f6, f7, f$))
Q = (qzero, q1, q2, q3, q4, qE)
F = (qE)
f: f0 = qzero f1(qzero) = q3
f2(qzero) = q1 f2(q3) = q1
f7(qzero) = q4 f6(qzero) = q2
f6(q4) = q2 f$(qzero,q1,q2) = qE

As example, different patterns of E shown in Fig. 8.

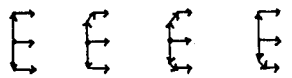


Fig. 8. Pattern of E

will be represented by the following trees respectively.

Fig. 9. Tree structure of fig. 8.

By scanning these five trees from frontier toward root by the above tree automation, all the five different patterns given in Fig. 8. are recognized as E.

In this recognition system, the number of states of tree automata is 160, 288 transition functions are defined,and the final state of each group is determined according to the characteristics of each group.

C. Recognition Results

Sample pattern consists of two types of fonts obtained from dot matrix printer and plotter. The test result is shown in Table 3. Table 4 shows the misrecognition and rejection results.

Recognition	517	97.2 %
Mis recognition	10	1.9 %
Reject	5	0.9 %
Total	532	

Table 3. Test results

Misrecognition	Input character	Mis recognized character	input character	Mis recognized character
	v	Y	Q	D
	s	S	G	C
	Q	O	D	O
	o	D	c	C
Reject	w , a , d , l			

Table 4. Misrecognition and rejection results

The misrecognition of v, Q, G, o, and D occurred because the character patterns are changed in the thinning operation. Rejection is caused by noise included during raw image capturing.

5. Conclusion

It is shown that the proposed set of features when used whid a high quality image capture system can yield high recognition accuracies, independent of font-type and font size by personnal computers. It is also evident that fast recognition is made possible by the parallel binary search of Tree automation.

References

[1] R. C. Gonzalez and M. C. Thomason, "Syntactic Pattern Recognition - An Introduction," Addison - Wesley, 1978
[2] Proceedings Fifth Intern. Conf. On Pattern Recognition, Miami Beach 1980
[3] M. Shridhar and R. Badreldin, "A High - Accuracy Syntactic Recognition Algorhithm for Handwritten Numerals," IEEE Tran. Systems, Man, And Cybernetics, vol. SMC - 15, No. 1, pp. 152 - 158, 1985
[4] J. Schürman, "Reading Machines," Proc. Sixth Intern. Conf. On Pattern Recognition, pp. 1031-1044, 1982
[5] N. J. Naccache and R. Shinghal, "SPTA: A Proposed Algorhithm for Thinning Binary Patterns," IEEE Tran. Systems, Man, and Cybernetics, vol. SMC-14, No.3, pp. 409 - 418, 1984
[6] K. S. Fu, "Syntactic Pattern Recognition and Application," Prentice - Hall, 1982