

構造解析에 의한 筆記體 한글의 온라인 認識

(On-Line Recognition of Handwritten Hangeul by Structure Analysis)

洪性民*, 金銀源*, 朴鍾國**, 趙源敬**

(Sung Min Hong, Eun Won Kim, Chong Kug Park and Won Kyung Cho)

要 約

본 논문에서는 필기체 한글의 온라인인식 알고리즘을 제안하였다. MDP(Minimum Distance Parser)를 구성하여 획을 인식하고, 획순에 의한 자소의 상태 그래프로부터 FSA(Finite-State Automata)를 구성하여 자소를 분리 인식한다.

실제하는 문장들로 총 3,000여자를 실험한 결과, 획의 인식율은 98.5%이고 자소의 분리율은 92.5%이었다.

Abstract

In this paper, an algorithm for the on-line recognition of handwritten Hangeul is proposed. The strokes are recognized by the minimum distance parser. The phonemes are separated by the finite-state automata resulted from the state graph of phonemes which are produced by the order of strokes.

By simulation result for 3,000 characters in practical sentences, the recognition rate of strokes is obtained to be 98.5% and the separation rate of phonemes is obtained to be 92.5%.

I. 序 論

최근 한글 인식에 관한 많은 연구가 진행되고 있으나, 대부분 오프라인(off-line) 인식에 관한 것이다. 그러나 OA, 워드 프로세서등의 정보 처리에서는 온라인(on-line) 인식이 요구된다. 미국, 일본등에서는 이미 많은 연구가 이루어져 실용화 되고 있다.^{1)~5)}

1960년대 말 부터 시작된 한글 인식에 관한 연구는 인쇄체,^{6)~11)} 및 필기체^{12)~14)}에 대하여 많은 연구가 진행 되었으며 인식율도 높게 보고되고 있다. 그러나 조합문자인 한글, 한자의 인식은 자소(부분패턴)의 분리

가 가장 큰 난점이다.

본 논문에서는 tablet을 입력장치로 하는 온라인 인식을 시도한다. 필기시 한글의 획순에 의하여 자소의 상태 그래프를 구성하고, 이 상태 그래프로부터 FSA(Finite-State Automata)를 유도하여 자소를 분리, 인식한다. 획은 MDP(Minimum Distance Parser)에 의한 구문해석 결과로 인식한다.

II. 한글 구조의 분석 및 전처리

1. 한글 구조의 분석

한글은 24개의 자소로 이루어지고, 모아쓰기에 의하여 생성되는 문자는 총 14,364자에 이르며,¹⁵⁾ 현재 실용되는 문자는 1,600여자이다.^{16)~18)}

한글 인식의 관점은 이와같이 문자의 수가 방대하고,

*準會員, **正會員, 慶熙大學校 工科學 電子工學科
(Dept. of Elec. Eng., Kyung Hee Univ.)

接受日字: 1985年 8月 6日

유사패턴이 많으며, 모아 쓰기된 자소의 분리가 어렵다는 점이다. 인쇄체인 경우는 패턴이 일정하므로 비교적 분리가 용이하나, 필기체의 경우는 더욱 어렵게 된다.

하나의 문자는 그림 1 과 같이 2~7 개의 자소로 이루어지고, 각 자소는 1~4 개의 획으로 이루어진다. 그러므로 획을 인식한다면 획의 조합에 의하여 자소를 분리 인식할 수 있다.

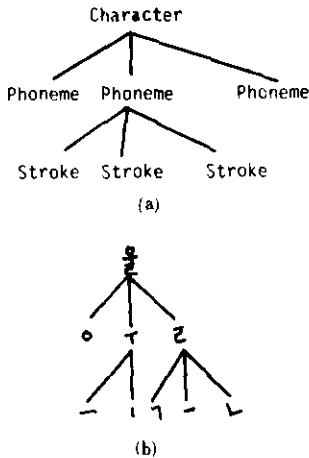


그림 1. 문자의 구성
Fig. 1. Combination structure of a character.

2. 전처리

전처리는 다음과 같은 단계로 진행된다.

1) 샘플링 및 코딩

Tablet으로부터 컴퓨터로 입력되는 연속한 좌표열에서 일정한 거리간격에 의하여 샘플링하고 이웃한 두 점에 하나의 방향 코드를 부여한다. 이때 방향코드는 그림 2의 8 방향 Freeman 체인코드¹⁾을 사용한다.

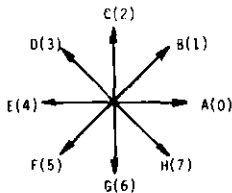


그림 2. 방향 코드
Fig. 2. Freeman's chain code.

2) 평활화 및 최소화

방향 코드로 기술된 문장은 굴곡(winding)이 생길 수 있으므로 평활화(smoothing)가 필요하게 된다. 본문에서는 그림 3 과 같이 연속한 코드를 조사하여 굴

곡을 교정하였다. 표 1은 발생 가능한 굴곡과 이의 교정표이다.

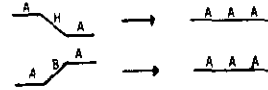


그림 3. 굴곡교정의 예
Fig. 3. Examples of correcting the winding.

표 1. 굴곡 교정표
Table 1. Correction table for winding.

Winding	Correction	Winding	Correction
ABA	AAA	AHA	AAA
BAB	BBB	BCB	BBB
CBC	CCC	CDC	CCC
DCD	DDD	DED	DDD
EDE	EEE	EFE	EEE
FEF	FFF	FGF	FFF
GFG	GGG	GHG	GGG
HGH	HHH	HAH	HHH

방향 코드로 기술된 문장은 동일방향 코드의 중복된 부분을 상당히 포함하고 있다. 문장의 길이는 구분해석 속도의 중요한 파라미터이므로 중복된 방향코드를 삭제하는 최소화(reduction)가 필요하다. 그러나 과도한 최소화는 오차를 증가 시키기 때문에 문장내의 최소 코드 수는 3 이상으로 하였다.

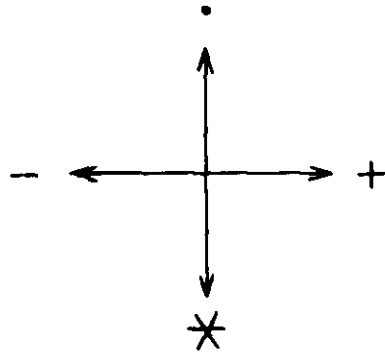


그림 4. 위치 정보 연산자
Fig. 4. Position operator.

3) 위치정보

한 획의 입력이 완료되면 전획과 비교하여 위치 정보를 갖는 연산자를 부여한다. 이때 연산자는 그림 4 와 같이 정의하며, 두 획의 상대 위치는 중심점 $C_{i-1}(\bar{x}, \bar{y})$ 와 $G_i(\bar{x}, \bar{y})$ 의 좌표로부터 유도한다.

중심점 $G_i(\bar{x}, \bar{y})$ 는 식(1)에 의해 구해진다.

$$\bar{x} = \frac{1}{N} \sum_{k=1}^N x_k$$

$$\bar{y} = \frac{1}{N} \sum_{k=1}^N y_k \quad (1)$$

III. 획 인식 알고리즘

한글의 자소는 기본적으로 |, -, /, \, ㄱ, ㄴ, ㄷ의 7가지 획의 조합으로 이루어지지만 필기체에서는 좀더 복잡하게 나타난다.

본 논문에서는 표 2에서 보는 바와 같이 필기체에서 나타나는 여러가지 획을 포함하여 표준패턴을 선정하였다.

정의된 각 획에 대하여 패턴문법 G를 추론하여 MDP를 구성하고 구문해석 결과로부터 유사도를 측정하여 인식한다.

표 2. 획의 표준패턴
Table 2. Reference of strokes.

No.	Stroke	No.	Stroke
1		11	U
2	-	12	ㄷ
3	/	13	ㄷ
4	\	14	ㄷ
5	└	15	ㄷ
6	ㄴ	16	ㄷ
7	ㄷ	17	ㄷ
8	ㄷ	18	ㄷ
9	ㄱ(ㄴ)	19	ㄷ
10	L		

1. 패턴문법

구문론적 패턴 인식은 형식 언어 이론에서 출발하였으며, Chomsky가 제시한 수학적 모델에 그 근거를 두고 있다.^{[10],[17]} 패턴문법에는 1차원적인 자유 문맥형, 감도 문맥형, 유한 상태 문법등과, 다차원적인 트리, 웨브문법등 여러가지가 있으나,^{[10]-[17]} 본 논문에서는 자유 문맥형 문법을 이용하였다.

MDP를 구성하기 위하여 다음의 몇가지 정의를 한다.

<정의 1> 패턴문법 G

$$G = (V_n, V_r, P, S)$$

V_n : 비종단 기호 혹은 부분패턴

V_r : 종단 기호 혹은 프리미티브

P: 프로덕션 혹은 구문법칙

$$a \rightarrow \beta, a \in V_n^*$$

$$\beta \in (V_n \cup V_r)^*$$

V^* 는 공백 문자를 포함한 V의 기호의 모든 가능한 열

S: 시작기호

<정의 2> 두개의 문장 $x, y \in V_r^*$ 에 대하여 변환 T를 다음과 같이 정의한다.

1) 대입 변환

$$w_1 a w_2 \xrightarrow{T_a} w_1 b w_2$$

2) 삭제 변환

$$w_1 a w_2 \xrightarrow{T_D} w_1 w_2$$

3) 삽입 변환

$$w_1 w_2 \xrightarrow{T_I} w_1 a w_2$$

$$w_1, w_2 \in V_r^*, a, b \in V_r$$

<정의 3> 위에서 정의된 각 변환에 웨이트를 부여한다.

1) $w_1 a w_2 \xrightarrow{T_a S(a,b)} w_1 b w_2$

2) $w_1 a w_2 \xrightarrow{T_D D(a)} w_1 w_2$

3) $w_1 w_2 \xrightarrow{T_I I(a)} w_1 a w_2$

여기서 S(a, b), D(a), I(a)는 각 변환의 웨이트이며 이 웨이트는 사용 빈도에 반비례하는 값을 부여한다.

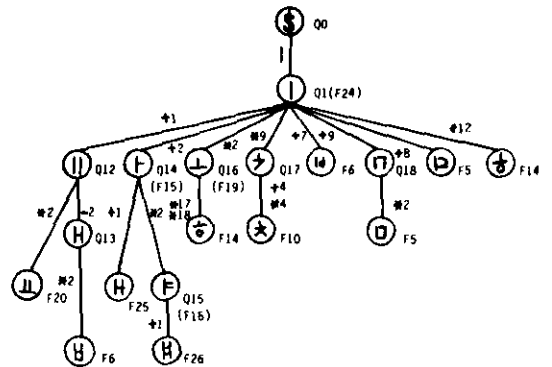


그림 5. 자소의 상태 그래프
Fig. 5. A state graph of phonemes.

2. 유사도 측정

구성된 각 획의 패턴문법에 의하여 입력 문장 x를 구문해석한다. 이때 각 패턴문법 G_i와 입력 x의 유사도는 다음과 같이 웨이트드 디스턴스로 정의한다.

<정의 4> x로부터 표준패턴 $r_i (i=1, 2, \dots, n, r_i \in G_i)$ 를 유도하는데 사용한 변환의 열을 $J = \{T_1, T_2, \dots, T_n\}$ 이라 할 때, 각 변환 T_i의 웨이트를 w_i 라 한다. 여기서 n은 표준패턴의 수이고, $T_i \in \{T_a, T_D, T_I\}$ 이다. 이

때 웨이트드 디스턴스 $d(G_i, x)$ 는 식(2)로 정의한다.

$$d(G_i, x) = \min \left[\sum_{k=1}^n w_k \right] \quad (2)$$

IV. 자소의 분리 및 인식 알고리즘

1. 자소의 구성

모든 자소는 필기시 출현하는 획의 순서에 따라 구성되는 자소의 상태 그래프로 표현될 수 있다. 그림 5는 그 예를 보인다. 여기서 입력기호의 숫자는 획의 번호를 나타내며, 연산자는 위치 정보를 나타낸다.

자소의 표준패턴은 표 3과 같이 자음 14개, 모음 10개, 그리고 종모음의 복합모음 4개, 총 28개로 하였으며, 이들은 키보드의 한글 자판의 자소이다. 복합 모음이나, 복합 자음, 그리고 쌍자음등은 하나의 자소로 인식할 수도 있으나 본 논문에서는 분리 인식한다.

표 3. 자소의 표준패턴

Table 3. Reference of phonemes.

No.	Phoneme	No.	Phoneme
1	ㄱ	15	ㅌ
2	ㅋ	16	ㅍ
3	ㆁ	17	ㄷ
4	ㄴ	18	ㅊ
5	ㄹ	19	ㄱ
6	ㅁ	20	ㅇ
7	ㅂ	21	ㅌ
8	ㅅ	22	ㅍ
9	ㅇ	23	ㅡ
10	ㅈ	24	ㅣ
11	ㅊ	25	ㅏ
12	ㅌ	26	ㅑ
13	ㅍ	27	ㅓ
14	ㅑ	28	ㅕ

2. 유한-상태 오토메이타(FSA)

앞절에서 정의한 자소의 상태 그래프에 기초하여 FSA를 구성한다.

유한-상태 오토메이톤 A는 다음과 같이 정의한다⁽⁴⁾
 $A = (\Sigma, Q, M, Q_0, F)$

Σ : 입력 기호의 유한 집합

Q : 상태의 유한 집합

M : 매핑 함수

Q_0 : 초기 상태

F : 종단 상태의 집합

유한-상태 오토메이톤을 그림으로 표시하면 그림 6과 같이 표현된다.

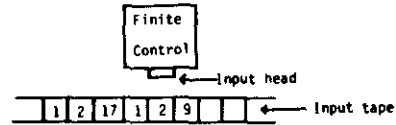


그림 6. 유한상태 오토메이톤
 Fig. 6. A finite-state automaton.

초기 상태 Q_0 에서 입력 기호를 받아들이면 다음 상태 Q_1 으로 천이가 이루어진다. 이것을 매핑 함수로 나타내면

$$M(Q_0, 1) = Q_1, \quad Q_0, Q_1 \in Q, \quad 1 \in \Sigma$$

이 된다. 다시 이것을 상태 그래프로 나타내면 그림 7이 된다.

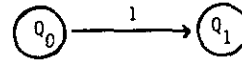


그림 7. $M(Q_0, 1) = Q_1$ 의 상태 그래프
 Fig. 7. Graphical representation of $M(Q_0, 1) = Q_1$.

그림 5의 자소의 상태 그래프에서 매핑 함수를 유도하면 FSA가 구성되며 자소를 분리 인식할 수 있다.

V. 실험 및 고찰

1. 실험 시스템

실험은 마이크로 컴퓨터를 사용하여 시뮬레이션 하였으며 그 개략도는 그림 8에 나타낸다.

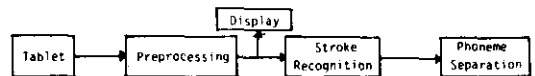


그림 8. 인식 시스템의 개략도
 Fig. 8. Block diagram of recognition system.

입력 장치는 Tablet을 사용하였으며 한문자당 64×64 화소가 되도록 원고지 형식으로 설계하였다.

문법 추론을 위한 샘플 데이터는 연구실내의 피실 현자들이 Tablet위에 쓴 데이터 중 조합 문자 1000자, 28개의 자소 각 100자, 19개의 획 단위 각 100개로 하였으며, 감독자 추론 방식으로 추론하였다.

인식 실험은 피실현자들로부터 성명, 주소와 국민 교육 현장, 격언 및 명언등의 문장을 데이터로 하였다.

그림 9는 전처리 과정에서 그래픽 화면에 출력된 것이다. 이 데이터를 전처리하여 생성된 문장을 구성된 MDP에서 구문해석하면 획을 인식할 수 있다. 인식된 획 번호와 연산자를 FSA 처리에 의하여 자소로

분리 인식할 수 있으며, 그 결과를 그림10에 나타낸다. 여기서 윗줄의 연산자와 숫자는 각각 위치 정보, 인식된 획 번호를 나타내며 아랫줄의 숫자는 분리된 자소와 번호를 나타낸다.

서	울	특	별	시
부	산	직	할	시

그림 9. 입력 문자의 출력 예
Fig. 9. Example of input character display.

RECOGNIZED STROKES ARE
+3 +4 +2 +1
SEPARATED TO PHONEMES
7 17

RECOGNIZED STROKES ARE
+17 *2 *1 *9 *2 *10
SEPARATED TO PHONEMES
8 21 4

그림10. 문자 "서울"의 FSA 처리 결과
Fig. 10. FSA processing result of "서울".

2. 결과 및 고찰

실험의 일반성을 나타내기 위하여 특정 문자가 아닌 일반적인 문장을 대상으로 하여 총 3,000여 자를 인식 실험하였다.

표 4. 획의 인식율
Table 4. Recognition rates of strokes.

Stroke No.	Recognition		Reject		Misrecognition		Total
	Count	Ratio	Count	Ratio	Count	Ratio	
1()	913	99.89	0	0	1	0.11	914
2(一)	823	99.88	0	0	1	0.12	824
3(/)	137	99.28	0	0	1	0.72	138
4(\)	240	97.96	0	0	5	2.04	245
5(ㄱ)	112	94.92	1	0.85	5	4.24	118
6(ㅋ)	32	94.12	1	2.94	1	2.94	34
7(ㆁ)	15	93.75	0	0	1	6.25	16
8(ㄴ)	42	100	0	0	0	0	42
9(ㄷ)	255	99.61	0	0	1	0.39	256
10(ㄹ)	213	97.71	0	0	5	2.29	218
11(ㅁ)	26	100	0	0	0	0	26
12(ㅂ)	12	85.71	1	7.14	1	7.14	14
13(ㅅ)	101	97.12	3	2.88	0	0	104
14(ㅇ)	76	100	0	0	0	0	76
15(ㄷ)	58	100	0	0	0	0	58
16(ㅁ)	25	100	0	0	0	0	25
17(ㅇ)	168	97.67	2	1.16	2	1.16	172
18(ㅇ)	126	93.33	4	2.96	5	3.70	135
19(ㅎ)	99	90.83	6	5.50	4	3.67	109
Total	3473	98.55	18	0.51	33	0.94	3524

실험한 문자중 임의의 800여자를 분석한 결과 획의

인식율은 98.5% 정도 되었으며 상세한 결과는 표4와 같다. 오 인식된 획은 자소의 분리 실패에 직접적인 원인이 되었으며, 인식불능(reject)인 경우는 즉시 재입력 하였다. 인식 불능 경계치는 실험결과 10으로 선택하였으며 최소거리가 이 값을 초과할 경우 인식 불능으로 하였다.

표 5. 자소의 분리율
Table 5. Separation rate of phonemes.

Total	Separation		Reject		Failure	
	Count	Ratio	Count	Ratio	Count	Ratio
3000	2774	92.47	73	2.43	153	5.1

표5는 자소의 분리 인식 결과를 나타낸다. 인식불능 문자는 꺾은선에 의한 재 분리에서도 문자로 성립되지 않은 경우이고, 오인식된 문자중 국(9*2*1*9) → 킨(9*2 + 1*9) 등의 몇 개 문자는 자소단위의 연산자를 부여하면 분리 가능하며, 위(17*2*1+1) → 유(17*2*1 + 1) 등의 오인식은 크기에 대한 고려(scaling variant)를 하면 인식 가능하다. 그 외의 오인식은 연산자에 의한 것과 잡음에 의한 것이었다. 연산자를 중심점에 하나만 부여하지 않고 시작점, 중심점, 끝점의 3점에 부여하고, 2개의 연산자의 일치 여부를 조사 한다면 분리 인식율은 높아 질 것으로 생각된다.

VI. 結 論

본 논문에서는 필기체 한글의 온라인 인식에 관하여 연구하였다. 인쇄체 문자인 경우는 오프라인 처리만이 가능하지만, OA기기, 워드 프로세서 등에서는 온라인 처리가 필요하게 된다. 필기체 한글의 인식에 있어서는 자소의 분리가 가장 어려운 문제이다. 본 논문에서는 자소를 획 단위로 인식한 후 인식된 획의 조합으로 자소를 분리하는 알고리즘을 제안하였다.

실험 결과 획의 인식율이 98.55%, 자소의 분리 인식율이 92.47%, 인식불능율이 2.43%로 본 알고리즘의 효용성을 확인하였다. 오인식의 주 원인은 Tablet 상의 잡음에 있었고 몇 가지 알고리즘의 개선에 의하여 인식율은 더 높아질 수 있을 것이다. 또한 영문자, 숫자들을 포함한 문자 인식에 있어서는 새로운 획의 정의, 매핑 함수의 추가와 함께 자소 단위의 연산자 부여, 한글 모아쓰기 알고리즘의 병행등에 의하여 인식할 수 있을 것이다.

전용 시스템 구성시에는 각 모듈의 파이프 라인 처리, MDP의 병렬 처리, FSA에서의 룩업 테이블 사

용에 의하여 실시간 처리가 가능할 것이다.

參 考 文 獻

- [1] S. Hanaki, Y. Temma and H. Yoshida, "An on-line character recognition aimed at a substitution for a billing machine keyboard," *Pattern Recognition*, vol. 8, pp. 63-71, 1976.
- [2] K. Odaka, T. Wakahara and S. Hashimoto, "On-line handwritten character recognition algorithm" *J. IECE*, vol. 65-D, No. 6, pp. 679-686, June 1982.
- [3] K. Odaka, T. Wakahara and S. Hashimoto, "On-line handwritten character recognizer," *J. IECE*, vol. 65-D, no. 8, pp. 951-958, Aug. 1982.
- [4] T. Wakahara, K. Odaka and M. Umeda, "Electro-luminescent display superposed by transparent Electromagnetic coupling tablet and its application to script input Japanese word processor," *J. IECE*, vol. 67-D, no. 9, pp. 981-988, Sep. 1984.
- [5] S. Hanaki, T. Temma, H. Yoshida, T. Arakawa, M. Suzuki, T. Seki and Y. Kikuchi, "On-line realtime character recognition system," *J. IECE*, vol. 64-D, no. 5, pp. 411-418, May 1981.
- [6] J.K. Lee, *Korean Character Display by Variable Combination and Its Recognition by Decomposition Method*. Ph.D. dissertation in Keio University, Japan, 1972.
- [7] 李柱根, 南宮在贊, 金榮建, "한글Pattern에서 Subpattern분리와 인식에 관한 연구," *電子工學會誌*, 第18卷 第3號, pp. 1-8, 6月 1981年
- [8] 이상구, 송정길, 양소철, "Fuzzy set를 이용한 인쇄체 한글 Pattern인식에 관한 연구," 한국정보과학회, 학술발표집, pp. 3-8, 4월, 1983년.
- [9] 崔炳旭, 市川忠男, 藤田廣一, "한글認識에 있어서의 子素抽出," *電子工學會誌*, 第18卷 第2號, pp. 36-43, 4月 1981年.
- [10] 남궁재찬, "Index-window algorithm을 이용한 한글의 Subpattern분리와 인식에 관한 연구," 인하대학교 대학원 박사학위 청구논문, 2월 1983년.
- [11] 오길남, 신승호, 진용옥, "2 차원 디지털 필터링에 의한 한글 자모의 인식 알고리즘," 전자공학회지, 제21권 第3號, pp. 55-59, 5월 1984년.
- [12] 이주근, 김홍기, "위상회전에 의한 필기체 한글의 자동인식," 전자공학회지, 第13권 第1號, pp. 23-30, 3월 1976년.
- [13] 천대영, 정장순, 이인석, "필기체 한글의 패턴 분류 방법의 개선에 관한 연구," 한국정보과학회, 학술발표집, 4월 1983년.
- [14] 박성삼, "Markovchain행렬에 의한 2진패턴의 인식," 경희대학교 대학원 석사학위 청구논문, 2월 1984년.
- [15] 이주근, "한글문자의 인식에 관한 연구(IV)," 전자공학회지, 第9卷 第4號, pp. 25-32, 9월 1972년.
- [16] K.S. Fu, *Syntactic Methods in Pattern Recognition*, Academic Press, New York, 1984.
- [17] A.V. Aho and J.D. Ullman, *The Theory of Parsing, Translation, and Compiling*, vol. 1, Prentice Hall, Englewood Cliffs, N.J. 1972.
- [18] K.S. Fu, *Digital Pattern Recognition*, Springer-Verlag, New York, 1980.
- [19] K.S. Fu and P.H. Swain, "On syntactic pattern recognition," in software Engineering, vol. 2, J.T. Tou, ed, Academic Press, New York, 1971.
- [20] K.S. Fu and T.L. Booth, "Grammatical Inference: introduction and survey-part 1," *IEEE Trans. SMC*, vol. SMC-5, no. 1, pp. 95-111, Jan. 1975.