

CFG 방법을 이용한 필기체 한글에서의 자소추출과 인식에 관한 연구

(A Study on Phoneme Extractions and Recognitions for Handwritten
Korean Characters using Context-Free Grammar)

金炯來*, 朴仁甲*, 徐東必*, 金에녹**

(Hyung Lae Kim, In Kap Park, Dong Phil Seo, and Enoch Kim)

要 約

본 논문에서는 Context-Free 문법을 이용하여 필기체 한글을 인식하는 알고리즘을 개발한다. 입력 문자를 세션화시키고, 세션화된 문자는 형식에 따라 자소를 1차원 스트링 값으로 변환시킨다. 만일 자소간에 접촉된 부분이 있으면 인덱스 기호 (#)를 사용하여 자소를 분리한다. Context-Free 문법은 자소의 유사성을 고려하여 군 문법을 구성하고, 푸쉬다운 오토마타 방법을 이용하여 자소별로 한글 문자를 구문 분석한다. 필기체에서는 획이 휘어지기 때문에 이것을 교정하기 위해서 본 논문에서는 두 문장사이에 오류된 부분을 정정한 변환 거리를 측정하고, 그 값이 최소값이 될때 인식하는 파싱거리 측정 방법을 도입하여 잘못된 부분을 정정하여 인식한다. 끝으로 실험 결과는 필기체 한글문자의 275개에 대해서 93.8%인식율을 보였다.

Abstract

This paper presents a method which can recognized the Handwritten Korean characters by using a Context-Free Grammar. The input characters are thinned in order to dwindle the mount of data, the thinned characters are converted into one-dimension strings according to six-forms. when the point of contact among phonemes is found, two phonemes are seperated respectively by marking the index mark (#) at the points, the Context-Free Grammar to input characters is classified into group grammars concerning the similarity of phonemes, input characters are parsed by making use of the Pushdown automata method. As the bent parts in the Handwritten characters are found frequently, We try to correct the bent parts by using the parsing distance measure, which recognize characters according to minium value caused by measuring the weight distance between two sentences. In this experiment, the recognition rate shows 93.8% to 275 Handwritten Korean characters.

*正會員, 建國大學校 電子工學科
(Dept. of Elec. Eng., Kunkuk Univ.)

**正會員, 連庵工業專門大學 電子計算科
(Dept. of Comp. Eng., Yunam Junior College)

接受日字: 1991年 10月 30日

I. 서 론

패턴 인식 방법은 패턴의 통계적인 특징을 이용하여 결정론적인 인식 방법과 패턴을 표현하는 원소와 이들

의 구조적 관계를 이용하는 구문론적 인식 방법으로 대별되며, 획의 구조적 관계로 파악할 수 있는 문자의 경우 구문론적 방법을 중심으로 연구되어 왔다.^[1, 2, 3] 따라서, 이 방법은 한글과 같이 계층적 구조를 가지고 있는 문자나 글자 모양의 변형이 심한 필기체 문자의 인식에 적합하므로 1970년대 후반 이후부터 많은 연구가 되고 있다.^[5, 6, 15] 한글 문자를 구조적 관점에서 검토하면 문자의 패턴은 6개의 형식으로 규격화할 수 있다. 따라서, 한글 인식 연구는 자소를 이용하는 구조 해석적 기법을 주로 이용하고, 문자 인식의 과정에서 쓰이는 특징이 무엇인가에 따라 세선화에 의한 방법과 윤곽선에 의한 방법으로 나눌 수 있다.^[16] 본 논문에서는 세선화에 의한 방법으로 자소를 분리한다.

세선화된 입력 패턴의 원소에 속성을 부여하여 문맥 정보를 이용하는 방법을 제시하였고,^[13] 김 상운^[11]은 세선화의 단점을 해결하기 위해서 획의 방향 성분을 추출한뒤, 이를 상대 공간에 묘사하고 이를 균일 비용 검색으로 가능한 한글의 조합을 검증하는 방법이 제시되었다.

필기체 한글을 인식하는데 있어서는 크게 두가지 문제점이 있다. 첫째는 획이 변형된 부분을 어떻게 정정하여 인식하는 방법이고, 다음은 자소의 접촉 부분을 어떻게 분리하여 자소를 추출하는 방법이다. 본 논문에서는 구부러진 자소를 정정하기 위해서 Fu와 Lu^[10, 11]가 제시한 구문론적 패턴에서 두개의 문장사이의 파싱(parsing) 거리를 측정하여 인식하는 방법을 도입하여 한글 문자의 인식율을 높인다. 그리고, 필기체 한글에서 자소의 접촉된 부분은 한글의 구조적인 특징과 위치 정보를 이용하여 자소를 분리하고, 분리된 자소를 필기 순서에 따라 1차원 스트링 값으로 자소를 특정 추출한다. Context-Free 문법은 자소의 유사성을 고려하여 군 문법으로 구성하고, 푸쉬다운 오토마타 방법중 Predictive 파서기로 자소를 구문 분석하여 인식한다.

II. 전처리 과정

1. 정규화

이진 영상 P(i, j)를 정규화하여 32 X 32 영상 V를 얻는다. 정규화 패턴 V는 식(1)과 같다.

$$V = \{ P(i, j) \mid 0 < i < 32, 0 < j < 32, p(1, 0) \} \quad (1)$$

p는 패턴의 화소이고, '1'은 문자 부분이고, '0'은 배경 부분이다.

2. 세선화

Deutsch 세선화 알고리즘을 사용하여 불필요한 외곽점 제거 작업을 반복 수행하여 직선화된 골격선을 추출한다.^[9] 그리고, 끝점(end-point), 굴곡점(corner-point), 분기점(branch-point) 및 교차점(cross-point)을 Crossing Number의 값을 이용하여 다음과 같이 나타낸다.

$$\text{끝 점} : P(i, j) = 1 \ \& \ \text{sum} = 1 \quad (2)$$

$$\text{굴곡점} : P(i, j) = 1 \ \& \ \text{sum} = 2 \ \& \ \text{differ} = 4 \quad (3)$$

$$\text{분기점} : P(i, j) = 1 \ \& \ \text{differ} = 6 \quad (4)$$

끝점은 P(i, j)=2로 나타내고, 굴곡점과 분기점은 P(i, j)=3으로 나타낸다.

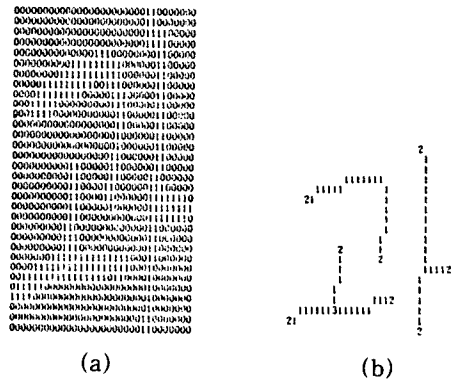


그림 1. 전처리 과정

- (a) 정규화된 패턴
- (b) 세선화된 패턴

Fig. 1. Preprocessing process.

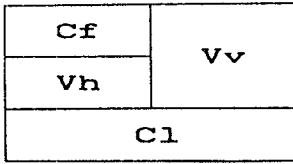
- (a) Normalized pattern,
- (b) thinned pattern.

III. 자소별 특징 추출

1. 한글 구조 분석

한글 문자는 자음과 모음이 쓰여지는 위치에 따라 그림 2과 같은 구조를 갖는 조합 문자이다.^[7]

그림 2은 기본자소 Ci와 Vj가 모음 중심으로 결합하여 구성되는 조합 문자로서 Cf는 3성음 구조에서 초성에 해당하는 것으로서 자음 Ci, 쌍자음 CiCj 이고, Vv, Vh는중성으로서 모음 Vi와 복모음 ViVj, ViVjVk이고, 여기서 중성 모음 Vv, Vh를 그 위치에 따라 수직모음(Vv)와 수평모음(Vh)으로 구분하여



Cf: 초성자음
 Vh: 수평모음
 Vv: 수직모음
 Cl: 종성자음

그림 2. 한글 문자의 구조

Fig. 2. Structure of Korean characters.

모두 분리하여 인식한다.

그림 2의 한글 문자의 구조는 Cf, Vv, Vh, Cl의 존재 유무에 따라 표 2과 같이 한글의 형식별 기본 자소로 분류할 수 있다.

표 1. 한글의 형식별 기본 자소

Table 1. Fundamental phonemes by the forms of Korean characters.

변수	형식					
	1형식	2형식	3형식	4형식	5형식	6형식
Cf	1	1	1	1	1	1
Vh	0	1	1	1	0	1
Vv	1	0	1	0	1	1
Cl	0	0	0	1	1	1

2. 한글 패턴 분류

1) 수평 모음 검출

수평 모음은 상좌단에서 우하단으로 세선화된 영상 M(i, j)을 검색하여 식 (5)을 만족할 경우, 시작점이 (sx, sy) 위치에 있으면 4, 6형식이 된다.

$$(sx, sy) = [M(i, j) = 2 \mid i_{max}-5 < i < i_{max}, \quad 0 < j < 4] \quad (5)$$

그러나, 초성의 끝점이 (sx, sy)에 존재할 수 있으므로 정확한 수평모음의 시작점은 식 (6)을 만족할때, (sx, sy)에 시작점이 있다.

$$(x, y) = [M(x, y) = 1 \mid sx+1 < x < sx+2, sy+5 < y < sy+7] \quad (6)$$

마찬가지로 식 (7)을 만족할 경우, 시작점이 (sx1, sy1) 위치에 있으면 2, 3형식이 된다.

$$(sx1, sy1) = [M(i, j) = 2 \mid i_{max}/2-3 < i < i_{max}+3, \quad 0 < j < 4] \quad (7)$$

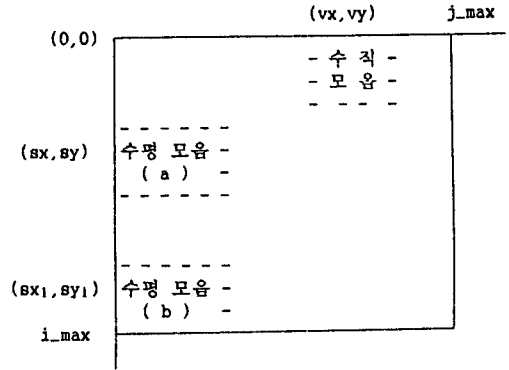


그림 3. 모음의 시작위치 표시

Fig. 3. Description of the starting position of vowel.

2) 수직 모음 검출

수직 모음은 좌상단에서 우하단으로 M(i,j)을 검색하여 식 (8)을 만족할 경우, 시작점이 (vx, vy) 위치에 있으면 1, 3, 5, 6형식이 된다.

$$(rx, ry) = [M(i, j) = 2 \mid 0 < i < i_{max}/3, \quad j_{max}/2 < j < j_{max}] \quad (8)$$

3) 종성 검출

종성은 수평 모음에 시작점이 식 (3-3) 위치에 있으면 종성이 존재하고 또한, 수평모음이 존재하지 않을 때는 식 (3-5)을 만족한 경우, 시작점이 (lx, ly) 위치에 있으면 5형식이 된다.

$$(lx, ly) = [M(i, j) = 2 \mid i_{max}/2+5 < i < i_{max}, \quad 5 < j < j_{max}/2] \quad (9)$$

3. 자소의 특징 추출

1) 자소 분리

(1) 수평 모음 분리

수평 모음의 분기점에서 초성, 수직 모음 및 종성 쪽으로 끝점이 나타나지 않고 접촉점이 생기면, 8-방향 코드를 이용하여 자소의 접촉 부분을 다음과 같이 추적하여 자소를 분리한다.^[4]

(i) 그림 4(a)처럼 초성에 접촉될때, 수평 모음에서

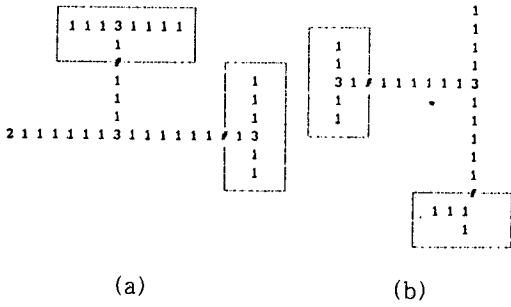


그림 4. 접촉된 자소 분리
(a) 수평 모음 일때
(b) 수직 모음 일때

Fig. 4. Segment of the contacted phonemes.
(a) Horizontal vowel,
(b) Vertical vowel.

분기점(x, y)을 찾고, 그 위치에서 범위(x-2<a(x+2, y-6<b(y-3)를 검사하여 분기점이 존재하면 (a, b+1) 위치에 #를 표시하여 자소를 분리한다.

(ii) 그림 4(a)처럼 수직 모음에 접촉될때, 수평 모음에서 분기점(x, y)을 찾고, 그 위치에서 범위(x+7 <a(x+13, y-2<b(y+2)를 검사하여 분기점이 존재하면 (a-1, b) 위치에 “#”를 표시하여 자소를 분리한다.

(iii) 종성에 접촉될때, 수평 모음에서 분기점(x, y)을 찾고, 그 위치에서 범위(x-2<a(x+2, y+3<b(y+6)를 검사하여 분기점이 존재하면 (a, b-1) 위치에 “#”를 표시하여 자소를 분리한다.

(2) 수직 모음 분리

수직 모음의 분기점에서 초성 및 종성 쪽으로 끝점이 나타나지 않고 접촉점이 존재하면, 위와 같은 방법을 이용하여 그림 4 (b)처럼 접촉점 부근에 “#”를 표시한다.

2) 자소별 특징화

한글의 필기순에 따라 초성, 수평모음, 수직모음 및 종성을 차례대로 Freeman의 8방향 코드를 이용하여 소문자 알파벳(a, b, c, d, e, f, g, h)로 된 1차원 열(string)로 표시한다.^[4]

자소를 특징화 할때, 자소의 시작점인 M(i, j)=2점을 찾고 그 점을 ‘s’로 바꾸고, 그 다음부터 M(i, j)=1인 점을 추적하여 소문자 알파벳 기호로 나타낸다.

기본 자소에서 굴곡점은 스트링 표현이 부적당 하기 때문에 끝점과 분기점으로만 구성된 입력 패턴을 스트링으로 나타내고, 스트링으로 나타낼때는 끝점에 “*”를 삽입하고, 분기점에 “+”를 삽입한다.

스트링 연산자(*, +)는 연산자 사이의 부분 스트링을 독립된 선분으로 나타내고, 2차원 영상을 유추

가능 하도록 할 뿐만 아니라, 구문 문법 작성을 쉽게 할 수 있다.

스트링 연산자를 삽입하여 나타낸 스트링 값은 모든 처리 결과의 끝을 나타내는 스트링 연산자 “*”를 종단 기호(“\$”)로 대체한다. 그래서 자소의 스트링 원소는 8개의 기본 생성원(a, b, c, d, e, f, g, h)이고, 스트링 연산자(*, +)이다. 그리고, 종단 기호(\$)로 구성된다.

그림 5에서 ‘곽’자에 대한 필기 순서에 따라 자소를 특징화 한다.

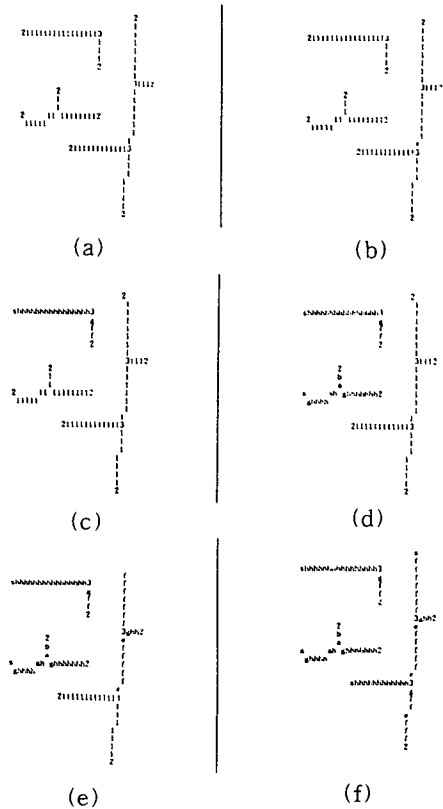


그림 5. 한글 자소 검출 과정(접 ‘곽’)

- (a) 세션화된 패턴
- (b) 자소 분리
- (c) 초성 검출
- (d) 수평 모음 검출
- (e) 수직 모음 검출
- (f) 종성 검출

Fig. 5. Detection of phonemes.

- (a) Thinned pattern,
- (b) Phoneme segment,
- (c) Detection of first consonant,
- (d) Detection of horizontal vowel,
- (e) Detection of vertical vowel,
- (f) Detection of last consonant,

1) 그림 5(a)는 세션화된 패턴이고, 끝점이 2이고 분기점이 3이다.

2) 그림 5(b)는 자소를 특징화시키기 전에 자소의 접촉 부분이 있는지를 검증하고 있으면 Index mark (#)를 사용하여 자소를 분리한다.

3) 그림 5(c)는 형식에 따라 자소의 시작점인 2를 찾고, 그 다음에 8-방향 코드를 사용하여 초성을 1차원 스트링으로 특징화한다.

4) 그림 3-4(d)는 수평 모음(⊃)를 1차원 스트링으로 특징화하고, 원소중에 hah를 변형 규칙의 알고리즘을 이용하여 hhh로 바꾼다.

5) 위의 방법으로 수직 모음과 중성을 특징화한다.

IV. 구문 분석 및 인식

성형 문법의 생성 규칙에 가해지는 제한에 따라 4가지 비제한 문법(Type0), context-sensitive 문법(Type1), context-free 문법(Type2), 정규 문법(Type3)으로 구분된다. Type0과 Type1의 문법은 추론 및 계산 이론에서 중요하게 사용되고, Type2와 Type3의 문법은 언어 이론과 구문 이론에서 주로 사용된다.^[1, 2, 3]

1. CONTEXT-FREE 문법

[정의 4-1] Context-free 문법 G는 $G = (V_n, V_t, P, S)$ 로 정의되고, 생성규칙은

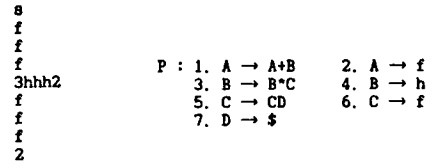
$$A \in V_n$$
$$A \rightarrow \beta \quad \beta = V^+ \text{ or } (V_n \cup V_t)^* \cdot \{\epsilon\} \quad (10)$$

의 형태를 가진다.

V_n 은 비종단 기호(Nonterminal Symbol)의 유한 집합이고, V_t 는 종단기호(Terminal Symbol)의 유한 집합이고, 단어 V 는 $V = V_n \cup V_t, V_n \cap V_t = \emptyset$ 이다. P는 생성규칙(Production Rule)의 유한 집합이고, S는 비종단 기호로 문장의 시작기호(Start Symbol)이다. 만일 문법 G가 context-free 문법이면, $L(G)$ 는 Context-free 언어이다. 또한, A가 S에 속하거나 비종단(Nonterminal)이면, $\omega \in V^*$ 가 A로 부터 유도된 스트링(String)이다.

$$L(G, A) = \{ \omega \mid \omega \in V_t^*, A \Rightarrow \omega \} \quad (11)$$

로 표현하면, 특정한 조건이 붙지않는 경우 $L(G,$



String : sfff+hhh*fff\$

그림 6. Context-free 문법 구성('†')

Fig. 6. Composition of context-free grammar.

=L(G)이다.

'†'자에 대한 Context-Free 문법을 구성하면, $G = (\{A, B, C, D\}, \{f, +, h, *, \$\}, P, A)$ 이고, 생성 규칙 P는

2. 푸쉬다운 오토마타(Pushdown Automata)

[정의 4-2] 푸쉬다운 오토마타 M는 식 (12)로 정의한다.

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \quad (12)$$

여기서,

- 1) Q는 상태 기호의 유한 집합이고,
- 2) Σ 는 입력 알파벳(Input Alphabet)의 유한 집합이고,
- 3) Γ 는 스택 알파벳(Stack Alphabet)의 유한 집합이고,
- 4) q_0 는 $q_0 \in Q$ 인 시작 기호(Start Symbol)이고,
- 5) Z_0 는 $Z_0 \in \Gamma$ 인 스택에 초기치로 나타난 시작 기호이고,
- 6) F는 $F \subseteq Q$ 인 최종 상태의 집합이고,
- 7) δ 는 $Q \times (\Sigma \cup \{\epsilon\}) \times T \Rightarrow Q\Gamma^*$ 로 표현되는 사상 함수이다.

푸쉬다운 오토마타의 동작을 나타내는 이전 운동은 두 가지 형태를 갖는다.

첫번째 형태로서 사상 함수는

$$\delta(q, a, Z) = \{(p_1, \beta_1), (p_2, \beta_2), \dots, (p_m, \beta_m)\} \quad (13)$$

으로 표현하며, q는 현재 상태이고, $p_i (1 \leq i < m)$ 은 차기 상태를 나타낸다.

두번째 형태로서 사상 함수는

$$\delta(q, \epsilon, Z) = \{(p_1, \beta_1), (p_2, \beta_2), \dots, (p_m, \beta_m)\} \quad (14)$$

으로 표현된다. 따라서 식 (13)과 식 (14)의 사상 함수에 의한 이전 운동을 하는 푸쉬다운 오토마타 $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ 에서의 사상 함수가

$$\delta(q, a, Z) = (p, \beta) \tag{15}$$

일때, 푸쉬다운 오토마타 M 의 현 상태가 q_0 , 입력 버퍼의 내용이 aw , 그리고 스택의 내용물이 $Z\alpha$ 이면, 푸쉬다운 오토마타 M 의 이전 운동은 식 (16)으로 표현하고,

$$(q_0, aw, \alpha) \xrightarrow{M} (p, w, \beta\alpha) \tag{16}$$

M 에 의해 한 상태에서 다른 상태로의 이동은 M 로 표시하고, 영변 이상의 이동은 \dot{M} 나타내며, 최종 상태로 수락된 언어 $L(M)$ 은 식 (17)로 정의한다.

$$L(M) = \{ \omega \mid (q_0, a, Z_0) \xrightarrow{\dot{M}} (p, \varepsilon, \beta), p \in F, \beta \in \Gamma^* \} \tag{17}$$

본 논문에서는 언어 $L(M)$ 의 이전 운동이 단지 한 개인 결정적 푸쉬다운 오토마타를 구성하여 입력 포인터의 역행 문제를 제거하기 위해 다음과 같은 조건이 필요하다.

1) 사상 함수 $(q, \varepsilon, Z) = \emptyset$ 이고, 사상 함수 $\delta(q, \varepsilon, Z)$ 가 Σ 내의 각 입력 기호 a 에 대하여 최대 1개의 원소만을 포함하거나,

2) 모든 입력 기호 $a \in \Sigma$ 에 대하여 사상 함수 $\delta(q, a, Z) = \emptyset$ 이고, 사상 함수 $\delta(q, \varepsilon, Z)$ 가 최대 1개의 원소만을 포함해야 한다.

그림 6에서 특정화된 '卜'자 자소에 대해서 푸쉬다운 오토마타(M)를 구성하면,

$$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{f, +, h, *, \$\}, \{A, B, C, D\}, \Gamma, \delta, q_0, A, \emptyset)$$

- $\delta(q_0, \varepsilon, A) (q_1, A+B)$
- $\delta(q_1, \varepsilon, B) (q_2, B * C)$
- $\delta(q_2, \varepsilon, C) (q_3, CD)$
- $\delta(q_3, \varepsilon, D) (q_4, \$)$
- $\delta(q_4, z, z) (q_4, \varepsilon), z \in \{f, +, h, *, \$\}$

3. Prediction 파서기(parser)

본 논문에서는 Top-Down 방법에서 역행 문제를 제거하고, 주어진 문장을 결정적인 구문 분석할 수 있는 —

표 2. 구문 분석 과정

Table 2. Process of syntactic analysis.

스택의 내용	입력 스트림	적용된 생성규칙 번호
A	f+h*f\$	
A+B	f+h*f\$	1
f+B	f+h*f\$	12
+B	+h*f\$	12
B	h*f\$	12
B*C	h*f\$	123
h*C	h*f\$	1234
*C	*f\$	1234
C	f\$	1234
CD	f\$	12345
fD	f\$	123456
D	\$	123456
\$	\$	1234567
	ε	수 락

Predictive 파서기를 사용한다. 그리고, 표 2는 그림 6에서 모음 '卜'자에 대한 구문 분석 과정을 나타낸다.

4. 파싱(Parsing) 거리 측정

문법 G 를 이용해 구성된 푸쉬다운 오토마타 M 이 인식 대상 패턴을 파싱할때 문자 패턴의 올바른 인식을 위해 거리 측정의 개념을 도입한다. 두 영상 사이의 Levenshtein 거리는 한 영상을 다른 영상으로 변환하는데 필요한 대치, 삭제, 및 삽입 무게 변환의 최소수로 정의 4-3과 같이 정의한다.^[10, 11, 12]

[정의 4-3] 어떤 두 문장 $x, y \in Vt^*$ 에서 변환 T 는 $y = T(x)$ 로 정의 하고, 세가지 변환에 대한 무게를 다음과 같이 정의한다.

1) 대치 무게 변환

$$w_1 a w_2 \xrightarrow{T_s, S(a, b)} w_1 b w_2 \quad w_1, w_2 \in Vt^* \tag{18}$$

a, b 모두는 Vt 에 포함되고, $S(a, b)$ 는 a 에서 b 로 대치되는 비용 또는 무게이고, $S(a, a) = 0$ 이된다.

2) 삭제 무게 변환

$$w_1 a w_2 \xrightarrow{T_d, D(a)} w_1 w_2 \tag{19}$$

a 모두는 V_t 에 포함되고, $D(a)$ 는 a가 제거되는 비용 또는 무게이다.

3) 삽입 무게 변환

$$w_1 w_2 \left| \begin{array}{l} T_i, I(a) \\ \hline w_1 b w_2 \end{array} \right. \quad w_1, w_2 \in V_t^* \quad (20)$$

a 모두는 V_t 에 포함되고, $I(a)$ 는 a가 삽입되는 비용 또는 무게이다.

어떤 두 문장 $x, y \in V_t^*$ 에서, y로부터 x를 유도하기 위해 요구되는 일련의 변환 J라 하고, J내에서 변환에 관련된 무게의 합은 J라 하며, x와 y사이의 가중거리(Weighted distance) $d(x, y)$ 는

$$d(x, y) = \min_j \{ |J| \} \quad (21)$$

로 정의하며, 또한 문장 X가 $I(G)$ 에 수락할때, X와 $I(G)$ 사이의 가중거리 $d(X, I(G))$ 는

$$d(X, I(G)) = \min \{ d(X, Y) \}, Y \in I(G) \quad (22)$$

으로 정의한다.

그림 6에서 'ㅌ'가 Context-Free 문법 G에 의해서 문장 X를 파싱할때, 그림 7처럼 굴곡된 자소를 정정할 수 있다.

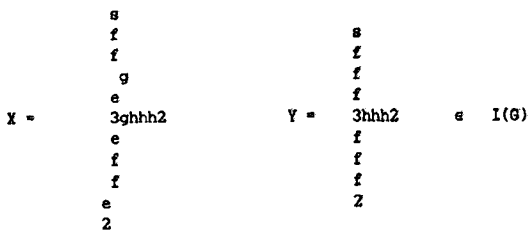


그림 7. 굴곡된 자소 정정
Fig. 7. Correction of a bent phoneme.

대치 변환 T_s 는 2이고, 삭제 변환 T_d 는 3이다. 그리고 가중거리 $d(X, Y)$ 는 5가 된다.

V. 실험 및 고찰

실험에 사용한 글자는 그림 8처럼 6가지 형식에 따라 랜덤하게 55개를 선정하여, 성인 5명에게 정자체로 필기한 문자를 사용하였다.

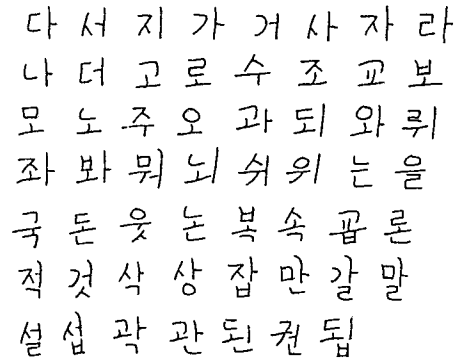


그림 8. 필기체 한글 문자 예
Fig. 8. Example of handwritten Korean characters.

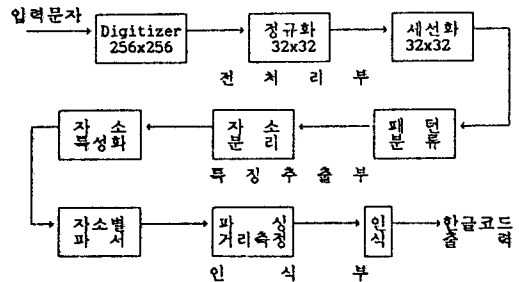


그림 9. 필기체 한글 인식 시스템
Fig. 9. Handwritten character recognition system.

필기체 한글 인식 시스템은 그림 9와 같이 전처리부, 특징 추출부 및 인식부로 나타낸다.

한글 패턴을 CCTV 카메라로 입력하여 연속 화상을 얻고, 영상처리 보드에 의해서 256계조도의 256X256 해상도를 갖는 디지털 영상으로 변환한다. 디지털 영상을 임계치 T로 2진화 한 뒤에 본 논문에서 제시한 알고리즘을 이용하여 실험한다.

275개의 문자중에 자소간 접촉 부분이 생긴 문자는 86개로 나타났고, 본 논문에서 제시한 방법으로 Index mark(#)를 사용하여 자소를 분리한 결과 98%의 분리율을 보였다. 자소분리가 되지 않는 문자는 필기자의 오류로 인해서 인식 불능으로 처리되었다.

문법 작성은 한글 자소의 유사성을 고려하여 자소를 군집합(자음의 예: ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㄴ, ㄹ, ㄷ, ㅌ, ㄴ, ㄹ)을 구성한다. 또한 인식 단계에서 두 문장의 파싱(Parsing) 거리를 측정하여 인식여부를 결정한다. 본 논문에서는 가중치와 임계치를 다음과 같이 주었다. 그리고 임계치 이하일때 정인식으로 결정된다.

- 1) 가중치: 알파벳(a, b, c, d, e, f, g, h)은 1을 주고, 스트링 연산자(*, +)는 3을 준다.

2) 임계치 : 모음은 10 이하 이고, 자음은 12 이하 이다.

한글 인식율은 표 3처럼 93.8% 나왔다.

표 3. 형식별 필기체 문자의 인식결과
Table 3. Recognition results of handwritten Korean characters by the forms.

구분 형식	1형식	2형식	3형식	4형식	5형식	6형식	계
실험 문자수	50	50	50	50	50	25	275
인식 문자수	49	49	46	46	46	22	258
오인식	1	1	4	4	4	3	17
인식율(%)	98.0	98.0	92.0	92.0	92.0	88.0	93.8

대부분의 오인식은 문자의 입력시 오류, 세션화시 기본선소의 상실, 그리고 특징 추출에서 자소를 잘못 특징화하는 경우 등이다. 오인식의 원인을 검토해 보면,

1) 세션화 과정에서 선소를 상실하여 오인식의 원인이 되었다.

따라서, 세션화 과정시 자소의 선소 부분을 개선할 수 있는 알고리즘 개발이 필요하다.

2) 특징 추출에서는 필기체 한글의 자소의 변형이 심하고, 패턴에 따라 자소의 위치가 불분명하여 오인식의 원인이 된다.

한 문자당 인식 시간(전처리부-인식부)은 평균 1.75초 걸렸고, 인식 단계별로 볼때 전처리 과정이 1.37초로 전체 인식시간에 78%를 차지한다. 따라서, 한글 인식의 실시간을 위해서는 전처리 과정에서의 하드웨어 구축이 필요하다.

VI. 결 론

본 논문에서는 필기체 한글에서 문제가 되는 자소간의 접촉 부분을 문자의 구조적인 특징과 자소의 위치 정보를 이용하여 자소를 분리한 결과 분리율이 98%로 나왔다.

자소의 특징 추출은 8방향 코드를 사용하여 1차원 스트링으로 구성하고, 자소 인식이 용이하게 하기 위해서 스트링 연산자(+, *)를 사용하여 선소의 연결 부분을 표시하였다. 또한, 필기체 한글에서는 획이 휘어지기 때문에 자소를 인식하는데 어려움이 있다. 따라서, 본 논문에서는 자소의 굵은 부분을 정정하기

위해서 파싱(Parsing)거리 측정 방법을 사용하였고, 실험에서 증명된 결과처럼 정인식의 경우는 변환거리가 모음은 10 이하이고, 자음은 12 이하로 얻었고, 그 이상일때는 오인식이 된다. 파싱거리 측정 방법을 사용하여 한글을 인식한 결과 자소의 문법이 간략해지고, 오류된 부분을 교정할 수 있었어 인식 시간이 단축되었고, 인식율을 높였다.

실험 결과 문제가 되는 것을 지적하면 실시간 처리를 하기 위해서는 전처리 과정을 하드웨어로 구축하는 것을 검토해야 하고, 인식율을 크게 저하시키는 세션화 방법에 대해서 꾸준한 연구가 필요하다. 그리고, Fuzzy 및 Neural network(신경망 모델)을 사용하여 본 논문의 한글 인식과 비교하는 것도 흥미있는 과제가 될 것이다.

參 考 文 獻

- [1] J. E. Hopcroft, and J. D. Ullman, Introduction to Automata Theory, Language, and computation, Addison-Wesley, Canada, 1979.
- [2] K. S. Fu, Syntactic Pattern Recognition and Application, Prentice-Hall, New Jersey, 1982.
- [3] R. C. Gonzalez and M. G. Thomason, Syntactic Pattern Recognition, Addison-Wesley, Cannada, 1978.
- [4] R. C. Gonzalez, and Paul Wintz, Digital Image Precessing, Addison-Wesley, Canada, 1978.
- [5] 전 병민, "배열문법을 이용한 필기체 한글 문자의 인식에 관한 연구", 연세대학교 대학원, 박사학위논문, 1988. 6.
- [6] 연 승호, "구문적 패턴을 이용한 필기체 숫자의 인식에 관한 연구", 충북대학교 대학원, 석사학위논문, 1988. 8.
- [7] 이 주근, "한글 문자의 인식에 관한 연구(IV)" 대한전자 공학회지, 제9권, 제4호, pp. 25 - 32, 1972. 9.
- [8] 유 승필, 김 태균, "속성문법을 이용한 필기체 한글 문서내의 자모 인식", 대한전자공학회지, 제 26권, 제3호, 1989. 3.
- [9] E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal, and triangular arrays", Comm. of the ACM, vol. 15, no. 9, pp. 827-837, 1972.

- [10] S. Y. Lu and K. S. Fu, "A Sentence-to-Sentence Clustering Procedure for Pattern Analysis", *IEEE Sys. Man Cyber.* vol. SMC-8, no. 5, pp. 381-389, May 1979.
- [11] K. S. Fu and S. Y. Lu, "A Clustering Procedure for Syntactic Pattern", *IEEE Trans. Syst. Man Cybernet* vol. SMC-7, no. 10, pp. 734-742, Oct. 1977.
- [12] K. S. Fu, "A Step Towards Unification of Syntactic and Statistical Pattern Recognition", *IEEE Trans. Pattern Anal. Mach. Intell.* vol. PAMI-8, no. 3, May. 1986.
- [13] K. H. Lee, K. B. Eom and R. L. Kashyp, "Character Recognition using Attributed Grammar", *IEEE Computer Society Conf. On Computer vision and Pattern Recognition*, June, Ann Arbor, 1988, pp. 418-423.
- [14] 김 상운, 방향 투영에 의한 인쇄체 한글의 기본 선소 추출 및 자동인식에 관한 연구, 연세대학교 대학원, 박사학위논문, 1987. 12.
- [15] 최 낙승, "Hough 변환을 이용한 필기체 한글의 인식에 관한 연구", 건국대학교 대학원, 석사학위논문, 1990. 2.
- [16] 양 황규, "구문론적 인식을 위한 인쇄체 한글의 자소 분리", 경북대학교 대학원, 석사학위논문, 1989. 12.

 著 者 紹 介



金 炯 來(正會員)

1947年 8月 27日生. 1970年 2月 연세대학교 이공대학 전기공학과 졸업(공학사: 전자공학 전공). 1972년 2월 연세대학교 대학원 전기공학과 졸업(공학석사: 전자공학 전공). 1982년 2월 연세대학교 대학원 전기공학과 졸업(공학박사). 1975년 5월~현재 건국대학교 공과대학 전자공학과 교수(학과장 역임). 1982년 7월~1984년 2월 미국 Northwestern대학교 교환교수. 1988년 3월~1991년 2월 건국대학교 공과대학 학장보 역임. 1992년 1월~현재 대한 전자공학회 학술담당 협동이사. 대한전자공학회 회로 및 시스템 연구회 위원장. 주관심분야는 패턴인식, 화상처리, 음성인식, 로보틱스 등임.



徐 東 必(正會員)

1963年 7月 2日生. 1987年 2월 건국대학교 전자공학과 졸업(공학사). 1987년 1월~1989년 4월 주) 금성사 전자기획관리실 근무. 1991년 8월 건국대학교 대학원 전자공학과 졸업(공학석사). 1992년 8월~현재(주)한양 전자 산업 기술 연구소 연구원. 주관심분야는 문자인식, 화상처리 등임.



朴 仁 甲(正會員)

1950年 2月 1日生. 1973年 3월 고려대학교 전자공학과 졸업(공학사). 1975년 2월 고려대학교 대학원 전자공학과 졸업(공학석사). 1986년 8월 고려대학교 대학원 전자공학과 졸업(공학박사). 1989년 7월~1990년 7월 Oregon State University 전기전산 공학과(PostDoc.) 1980년 9월~현재 건국대학교 전자공학과 교수(학과장 역임). 주관심분야는 컴퓨터 네트워크, 분산처리, 병렬처리, B-ISDN 등임.



金 에 녹(正會員)

1955年 6월 15日生. 1977年 2월 광운대학교 전자계산학과(이학사). 1982년 2월 건국대학교 대학원 전자공학과(공학석사). 1989년 2월 건국대학교 대학원 전자공학과 박사과정 수료. 1983년 12월~현재 연암공업전문대학 전자계산과 부교수. 주관심분야는 디지털 신호처리, Fuzzy 시스템 등임.