

論文 91-28B-10-1

병렬성을 갖는 WLD 알고리즘을 이용한 온라인 필기체 한글, 영문자 및 숫자 패턴인식

(A Study on the On-Line Handwritten Hangeul & Alphanumeric
Pattern Recognition Using WLD with Parallelism)

金 銀 源*, 趙 源 敬*

(Eun Won Kim and Won Kyung Cho)

要 約

본 논문은 병렬성을 갖는 WLD 알고리즘을 이용한 필기체 문자의 온라인 인식에 대한 연구이다. 한글은 문자의 구조적 특성으로 인하여 자소 단위로 처리하며, 영문자 및 숫자는 문자 단위로 분류할 수 있다. 그리고, 특수 목적 지향 프로세서의 실현을 위하여 WLD 알고리즘의 병렬성과 동시성을 고찰 하였다. 실제하는 문장에서 10,000여자에 대한 시뮬레이션 결과, 회 인식률은 96.57%이고, 자소 및 문자의 분리율은 95.4%이므로 본 연구의 유용성을 볼 수 있었다.

Abstract

In this paper, we studies the on-line recognition of handwritten character using WLD (Weighted Levenshtein Distance) algorithm with parallelism. The Hangeul can be separated for unit of phonemes and the Alphanumeric can be separated for unit of characters. And, we studies the parallelism and the concurrency of the WLD algorithm for realization of special-purpose processor.

By the simulation result for 10,000 characters in practical sentences, the recognition rate of strokes is obtained 96.57% and the separation rate for phonemes and characters is obtained 95.4%.

I. 서 론

다양한 문자들의 구조를 분석함으로써 일정한 형태를 취하는 패턴으로 기술할 수 있으며, 이러한 패턴들을 전용시스템을 통하여 처리하고자 하는 연구가 진행되어 왔다!¹⁻⁸⁾

또한, 고성능의 VLSI 지향적인 어레이 프로세서는 기본적인 동작을 수행하는 다수의 처리요소 (process-

ing Element)들을 이용하며, 동시 처리와 병렬 처리를 특징으로 하고, 처리 요소의 단순-규격화로 인하여 설계가 용이하며, 고집적도를 이룰 수 있다. 특히, 시스톨릭(Systolic) 구조는 많은 데이터의 실시간 처리가 요구되는 분야에서 응용되고 있으며, 고속 처리가 필요한 각종 알고리즘을 어레이(array)에서 실현시키기 위하여 병렬성에 따른 알고리즘의 변환에 관한 연구가 활발하게 진행되고 있다!⁹⁻¹²⁾

본 논문은 구문론적 방법에 의하여 한글 및 영문자, 숫자 문자 패턴에 대한 각 문자 패턴의 구성 원리를 분석하고 인식하고자 하였으며, 코드열로 기술

*正會員, 慶熙大學校 電子工學科
(Dept. of Eng. Elec., Koughee Univ.)
接受日字: 1991年 4月 11日

된 문자들의 거리값을 구하기 위하여 WLD 알고리즘을 사용하였고, 또한 WLD 알고리즘의 병렬성을 고찰함으로써 문자열 인식을 위한 프로세서의 실현성을 보고자 하였다. 근본적으로 한글, 영문자 및 숫자 문자 패턴들은 부분 패턴의 패턴원소로 분해하였을 경우 매우 단순한 획의 성분들로 구성되며, 이 패턴 원소들의 조합원리에 의하여 자소나 문자로 형성됨을 볼 수 있으므로 각 문자 패턴에 대한 패턴 묘사를 할 수 있다. 또한, 패턴 원소들은 코드열(code-string)의 나열로서 기술될 수 있으며, 가상획을 이용하여 각각의 패턴 원소들에 대한 상호 결합상태를 기술할 수 있다. 대상 문자로서 한글, 영문자 및 숫자 문자 패턴을 삼았고, 영문자에 있어서 인쇄체 대 문자를 필기한 필기체 문자로 대상을 한정하였으며, 테블렛을 이용한 온라인 인식을 시도하였다.

II. 한글, 숫자 및 영문자 패턴 상호간의 관계

한글, 숫자 및 영문자 패턴을 인식 처리하기 위한 상호간의 관계 고찰은 각 패턴의 구조 분석을 통하여 특징간의 유사성을 찾기 위한 것이며, 이 유사성은 인식에 있어서 매우 중요한 역할을 한다. 즉, 기본적인 문자 형태에서 구조상의 동일성을 찾아내는 것이며, 이 동일성은 각 패턴 사이에서 애매성을 가중시킨다. 그러므로, 필기체 온라인 인식에 있어서 한글, 숫자 및 영문자 패턴간에는 다음과 같은 특징들이 존재한다.

- 1) 대상 문자 모두 필기자 습성에 따른 변형이 심화된다.
 - 2) 한글 및 영문자 패턴은 대부분 기본적으로 직선 구조를 많이 나타내고 있으나, 필기체에 있어서 습성에 의한 곡선 구조가 많이 발생되고 있다. 한편으로, 숫자 패턴은 대부분 직선 구조 보다는 곡선 구조를 취한다.
 - 3) 각 패턴간에 있어서 동일한 획 성분들이 많다.
 - 4) 각 패턴의 자소 성분들은 1개부터 최대 4개의 획으로 구성되며, 획 단위 처리를 할 수 있다.
 - 5) 한글은 조합문자 구조인 목 구조의 계층구조를 취하고 있는 반면에 숫자 및 영문자 패턴은 풀어쓰기 형태를 취하고 있다.
 - 6) 입력 도구인 테블렛을 사용하였을 때, 단일 체인 코드 발생이 매우 유사함을 볼 수 있다.
 - 7) 각 패턴들은 획 단위로 분석하였을 때, 대부분의 획들은 매우 단순한 구조를 취하게 된다.
 - 8) 각 패턴에 있어서, 위치 연산자가 매우 중요한 역할을 하게 된다.
- 그러나 5)에 있어서 7)의 원리를 이용하여 분석하

였을 경우 한글 패턴이나 숫자 및 영문자 패턴에서의 모든 문자는 그림1에서와 같이 풀어쓰기 형태로 나열한 목 (tree) 구조의 획 조합 구조를 취하고 있으며, 4)와 같은 획 단위 처리의 원리가 된다. 필기자 습성에 의한 1)과 2)의 성질과 단일 체인 코드 사용시 6)의 성질은 3)에서와 같은 성질과 가상 획을 도입한 2중 체인 코드의 사용으로 해결할 수 있다. 즉, 다수 발생하는 획으로서 표준 패턴 사전을 구성하고 또한, 2중 체인 코드의 사용으로 각 획들 상호간의 위치 정보를 부여함으로써 맵핑(mapping) 통한 자소 분리를 행할 수 있다.

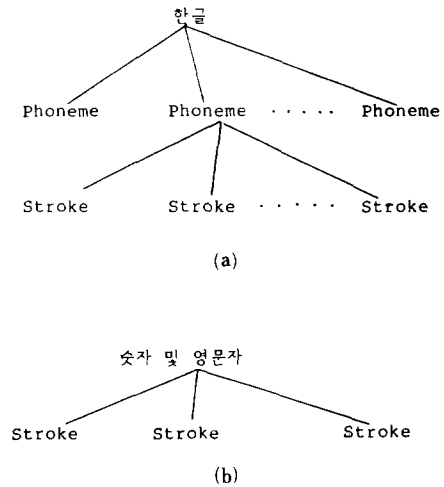


그림 1. 문자의 결합 구조

- (a) 한글 패턴
- (b) 숫자 및 영문자 패턴

Fig. 1. Combination structure of character.

- (a) hangeul pattern,
- (b) alphanumeric pattern.

III. 문자의 기술

문자는 획들의 조합원리에 의하여 구성되며, 일련의 획들은 테블렛상의 펜(pen)의 움직임에 따라 X, Y 좌표값들의 집합으로써 변환된다. 이와 같이 얻어진 좌표점 중에서, 직전의 좌표점과 같은 위치의 점은 무시하고, 전후 두 좌표점이 서로 상이한 점만을 샘플점으로 등록하며, 펜 다운(pen-down)으로부터 펜 업(Pen-Up)까지를 하나의 획으로 정의한다. 입력문자를 I, 그 문자를 구성하는 각 획(stroke)을 S_i라고 한다면, 입력문자 I는 다음과 같이 나타낼 수 있다.

$I = (S_1, S_2, S_3, \dots, S_i, \dots, S_N)$
 $S_i = (P_{i,1}, P_{i,2}, \dots, P_{i,j}, \dots, P_{i,m})$
 $P_{i,j} = (X_{i,j}, Y_{i,j}); S_i$ 의 j 번째 샘플점의 좌표
 $P_{i,j} \neq P_{i,j+1}$
 $i = 1, 2, 3, \dots, N;$
 $j = 1, 2, 3, \dots, m-1;$
 N ; 획의 총수
 m ; S_i 의 샘플점 수

또한, 전 획의 종점에서 다음 획의 시작점까지를 가상 획 성분으로 정의하면, 문자를 구성하는 획들의 위치 정보를 갖게 되며, 한편으로 문자의 필순 정보가 된다. 그러므로, 현재 위치 좌표점을 $P_{i,j}$ 라 놓고 $\Delta x, \Delta y$ 의 변위량을 갖는다면 8방향으로 가상 직선 성분 하나가 발생되며, 다음과 같이 표현될 수 있다.

한 문자를 구성하기 위한 가상획의 집합을 I_m 이라 하고, 가상 획을 PS라 놓으면,

$I_m = (PS_1, PS_2, \dots, PS_k, \dots, PS_N)$
 $PS_k = (P_{i,j}, P_{i+\Delta x, j+\Delta y});$ 가상 획의 좌표점
 $P_{i,j} \neq P_{i+\Delta x, j+\Delta y}$
 $\Delta x, \Delta y;$ 변위량
 $N;$ 가상 획의 총수

필순 정보인 가상 획들의 발생 횟수에 의하여 한글, 숫자 및 영문자 패턴의 문자 사전 분류를 행한다. 즉, 한글, 숫자 및 영문자 패턴들은 그들 문자를 구성하기 위하여 일정량의 획을 취한다. 여기에서, 영문자 및 숫자 패턴은 4개 이하의 획으로서 문자를 구성하며, 한글은 한개의 문자를 구성하기 위하여 2-7개의 자소가 필요하고, 각 자소는 1-4개의 획으로 이루어진다. 한편으로, 각각의 획들은 8방향 프리먼스 체인코드(8-directional freeman's chain codes)를 사용하여 코드열 패턴으로서 표현된다. 이때, 사용되는 프리먼스 체인코드는 2중 체인 코드로 변형된 체인코드로 사용된다. 그림2는 변형된 2중 체인코드와 그 사용 예를 보인다.

IV. 획 인식과 자소 및 문자의 분류

1. 획 인식 알고리즘

획 인식은 DP 매칭법에 의하여 모든 표준 패턴과 임의의 입력 패턴간의 거리가 가장 가까운 패턴을 검색하여 인식하며, 두 패턴간의 거리값 계산은 WLD 알고리즘을 이용하여 계산한다.

[정의] 임의의 코드열 패턴 X의 i 번째 코드 x_i 와 표준 코드열 R의 j 번째 코드 r_j 를 비교하여 서로 다를 경우, 두 코드열을 일치시키기 위하여 x_i 에 부과하는 변환의 종류는 다음과 같다.

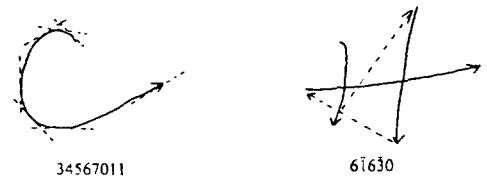
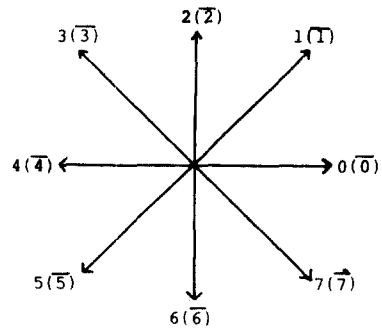


그림 2. 변형된 2중 체인코드와 그 사용 예
Fig. 2. Double 8-directional freeman's chain code & example.

(1) 삭제 변환 (Deletion) ; $D(x_i)$

$$\alpha x_i \beta \xrightarrow{D(x_i)} \alpha \beta$$

(2) 삽입 변환 (Insertion) ; $I(x_i, r_j)$

$$\alpha x_i \beta \xrightarrow{I(x_i, r_j)} \alpha r_j x_i \beta$$

(3) 대체 변환 (Substitution) ; $S(x_i, r_j)$

$$\alpha x_i \beta \xrightarrow{S(x_i, r_j)} \alpha r_j \beta$$

이 때, α, β 는 유한 길이를 갖는 코드열이다. 위에서 정의된 변환 방법들을 이용하여 최소한의 횟수로 X를 R에 일치하도록 하며, 두 코드열 패턴 X와 R을 일치시킬 때 발생하는 각 변환에 서로 다른 가중치를 부여하는 경우가 WLD $d^w(X, R)$ 이다. 그림 3의 각 노드(Node)는 표준 코드열 패턴의 r_j 와 입력 코드열 패턴의 x_i 의 비교 및 각 변환에 해당하는 경로를 갖는다. 거리값은 코드 비교후 선택한 각 변환 경로를 지날 때, 그 변환 가중치 값이 누적되어 계산된다. 코드 비교 및 거리값 계산은 B에서 시작하여 최종적으로 E지점에서 가중치 합이 거리값이 된다. 그러나 각 노드에서는 3가지의 변환 경로가 존재하며, 이들 경로중 최적의 경로를 선택하여야 함으로 이를 위하여 DP기법을 도입한다. 최적 변환에 의한 부분 거리값 $P(i, j)$ 의 계산은 다음과 같다.

$$P(i, j) = \text{Min} \begin{cases} P(i-1, j) + W_d \\ P(i, j-1) + W_i \\ P(i-1, j-1) + W_s \end{cases}$$

[알고리즘] WLD 알고리즘

- 0) $P(0, 0) = 0$
- 1) for $i=1, |X|$
do $P(i, 0) = P(i-1, 0) + W_d$;
- 2) for $j=1, |R|$
do $P(0, j) = P(0, j-1) + W_i$;
- 3) for $i=1, |X|$
for $j=1, |R|$
do
 $F_1 : \text{if}(x_i = r_j) \text{ then } T_1 = P(i-1, j-1);$
 $\text{else } T_1 = P(i-1, j-1) + W_s;$
 $T_2 = P(i-1, j) + W_d;$
 $T_3 = P(i, j-1) + W_i;$
 $F_2 : P(i, j) = \text{Min}(T_1, T_2, T_3);$
 }
- 4) $d^w(X, R) = P(|X|, |R|);$

알고리즘의 병렬성은 계산과정의 동시성을 살펴봄으로써 구할 수 있다. 이때, 동시성은 각 과정별 계산상의 데이터 의존성에 따라 결정되며, 데이터 의존성이 낮을 수록 동시성이 높다. WLD 알고리즘의 경우, 과정3)에서 최적 변환에 의한 거리값 계산은 $|R| \times |X|$ 회의 코드 비교(F_1 ;) 및 부분 거리값 $P(i, j)$ 의 계산(F_2 ;)이 반복되며, 데이터 의존성을 살펴보면 $P(i, j)$ 는 $P(i-1, j)$, $P(i, j-1)$, $P(i-1, j-1)$ 의 결과에 영향을 받는다. 그림4는 코드 비교와 $P(i, j)$ 의 계산 부분을 단위 모듈로 하는 각 노드들 사이의 데이터 의존 관계에 따라 이들을 1차원 어레이로 매핑한 것이며, k 방향으로 진행되는 사선 선상의 각 노드들에서 동시 처리가 이루어지는 것을 볼 수 있다. 이때, 병렬성에 따라 각 노드들을 k -인덱스를 도입하여 표시하면, $P(i, j)$ 는 $P(k)$, $P(i-1, j)$ 와 $P(i, j-1)$ 은 $P(k-1)$ 로, $P(i-1, j-1)$ 은 $P(k-2)$ 로 나타낼 수 있다. 그러나 1차원 어레이로 매핑된 노드들 사이의 데이터 교환이 파이프라인 방식으로 이루어지기 때문에 $P(k)$ 는 $P(k-2)$ 를 직접 참조할 수 없다. 따라서 $P(i, j)$ 의 계산의 k 벡터 방향으로 각 노드에서 동시 처리되며, k -순서에 따른 병렬 WLD 알고리즘은 다음과 같다.

[병렬 WLD 알고리즘]

- for $k=1, (|X|+|R|-1)$
 $F_1 : R(k) = R(k-1); X(n) = Xn;$
 if $R(k) = X(n)$ then $T_1 = S(k);$
 else $T_1 = S(k) + W_s;$
 $T_2 = I(k) + W_i;$
 $T_3 = D(k) + W_d;$

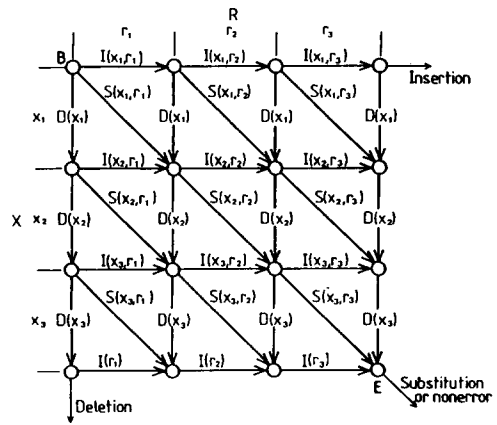


그림 3. 변환에 의한 코드열 매칭 과정

Fig. 3. Graph interpretation of matching metric.

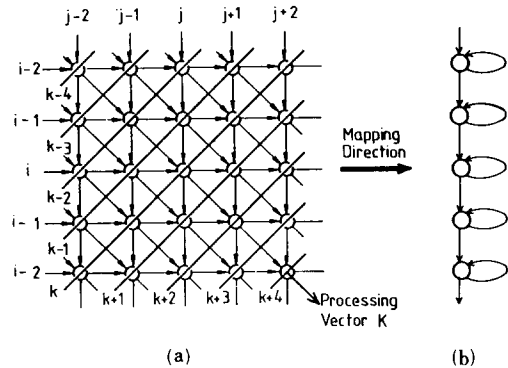


그림 4. 데이터 의존 관계에 의한 1차원 매핑

Fig. 4. Mapping of matching metric to linear arrays.

(a) Data dependence and concurrence between nodes,

(b) Linear mapped arrays and data graph.

$$\begin{aligned}
 F_2 : D(k) &= P(k-1); \\
 P(k) &= \text{Min}(T_1, T_2, T_3); \\
 I(k) &= P(k); \\
 S(k) &= D(k);
 \end{aligned}$$

병렬 처리할 경우 코드비교 및 부분 거리값의 계산은 k 인덱스에 의하여 $|X|+|R|-1$ 의 반복으로 수행된다. 그러므로 그림 4(b)와 같이 1차원 시스템 어레이로 실현할 수 있다.

2. 자소 및 문자의 분류

표1에서와 같이 전체 49개로서 표준획의 사전을

구성하였다. 문자를 인식하기 위해서는 자소의 분리가 되어야 하며 자소를 이루는 획들의 결합상태를 통하여 자소를 분리한다. 한글은 1-4개로 구성된 획들의 조합에 의하여 자소를 형성하며, 이 자소들은 2-7개의 조합으로서 하나의 문자를 발생시킨다. 또한, 영문자 및 숫자 패턴들은 1-4개의 획들로서 하나의 문자를 구성한다. 따라서, 한글은 자소단위로 분리하여 인식처리하며, 영문자 및 숫자 패턴은 조합된 문자단위로 인식처리 한다. 한편, 획과 획 사이에서 발생된 가상획 성분은 필기시 필순의 위치정보를 표시하게 된다. 그러므로 필기시 출현하는 획의 순서와 가상획에 따른 정보를 이용하여 상태전이 그래프가 구성된다. 그림5는 상태전이 그래프의 한 예를 보인다. 상태 전이 그래프에서 숫자는 DP 매칭에서 인식된 획의 번호이며, 영문자(A-H)는 가상획 성분을 나타내고, Q_i는 중간 전이 상태, F_i는 종단상태를

표 1. 전체 표준 획 사전의 구성
Table 1. Reference of strokes.

No.	구분	Stroke	No.	구분	Stroke	No.	구분	Stroke	No.	구분	Stroke
1	한	ㄷ	14	공	/	27	수	3	40	영	ㄹ
2	한	ㄴ	15	공	\	28	수	5	41	영	ㅅ
3	한	ㄹ	16	공	7	29	수	6	42	영	ㅈ
4	한	ㄱ	17	공	7	30	수	7	43	영	ㅊ
5	한	ㅁ	18	공	0	31	수	8	44	영	ㅊ
6	한	ㅂ	19	공	0	32	수	8	45	영	ㅊ
7	한	ㄷ	20	공	L	33	수	9	46	영	W
8	한	ㅊ	21	수	<	34	수	9	47	영	Y
9	한	ㅊ	22	수	1	35	수	9	48	영	Z
10	한	ㅊ	23	수	2	36	영	C	49	영	V
11	한	ㅊ	24	수	Z	37	영	C			
12	공		25	수	Z	38	영	J			
13	공	—	26	수	3	39	영	<			

표 2. 자소 및 문자의 표준패턴
Table 2. Reference of phonemes & characters.

F - No.	자소 및 문자	F - No.	자소 및 문자	F - No.	자소 및 문자	F - No.	자소 및 문자
1	7	17	ㄱ	33	5	49	K
2	L	18	ㅋ	34	6	50	L
3	C	19	ㄴ	35	7	51	M
4	ㄹ	20	ㄴ	36	8	52	N
5	ㅁ	21	ㅍ	37	9	53	O
6	H	22	ㅍ	38	∅	54	P
7	ㅂ	23	—	39	A	55	Q
8	O	24		40	B	56	R
9	Z	25	H	41	C	57	S
10	ㅊ	26	H	42	D	58	T
11	7	27	ㅋ	43	E	59	U
12	E	28	ㅋ	44	F	60	V
13	ㅈ	29	1	45	G	61	W
14	ㅊ	30	Z	46	H	62	X
15	ㅊ	31	3	47	I	63	Y
16	F	32	4	48	J	64	Z

나타낸다. 상태 전이 그래프의 종단상태로 표시되는 것이 한글의 자소 또는 영문자 및 숫자 패턴의 문자가 되며, 표2는 자소 및 문자의 표준패턴을 나타낸다. 그림5의 상태 전이 그래프에 의하여 유한상태 오토 메타를 구성할 수 있으며, 다음과 같이 정의된다.

- A = (Σ, Q, M, Q₀, F)
- Σ : 입력기호의 유한 집합
- Q : 상태의 유한 집합
- M : 맵핑 함수
- Q₀ : 초기 상태, Q₀ ∈ Q
- F : 종단 상태의 유한 집합

상태 Q_i에서 입력 데이터를 받으면 맵핑함수에 의하여 다음상태 Q_j로 상태전이가 이루어지며, 맵핑함수 M은 다음과 같이 정의된다.

$$M(Q_i, S_s, S_n, P_{ps}, N_{ps}) = Q_j$$

$$(Q_i, Q_j \in Q, S_s, S_n, P_{ps}, N_{ps} \in \Sigma)$$

여기에서, S_s는 획의 시작정보이며, S_n은 획의 번호

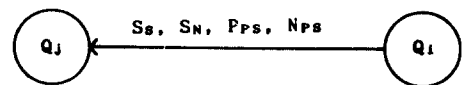


그림 6. 맵핑 함수 M(Q_i, S_s, S_n, P_{ps}, N_{ps}) = Q_j의 그래프

Fig. 6. Graphical representation of M(Q_i, S_s, S_n, P_{ps}, N_{ps}) = Q_j.

그림 5. 자소 및 문자의 상태전이 그래프의 예
Fig. 5. Example of state transition graph of phonemes & characters.

이고, P_{PS} 는 현재의 가상화이며, N_{PS} 는 다음 가상화의 정보이다. 그림6은 매핑함수 M 을 나타낸다. 기술된 자소 및 문자의 상태 천이 그래프에서 오토메톤을 구성하면 다음과 같다.

$$A = (\Sigma, Q, M, Q_0, F)$$

$$\Sigma = \{S_s, A, B, \dots, H\} \quad | 1, 2, \dots, 49\}$$

$$Q = \{Q_0, Q_1, \dots, Q_{60}\}$$

$$M(Q_i, \Sigma) = \{Q_j, F_k\}$$

$$Q_0 = \emptyset$$

$$F = \{F_1, F_2, \dots, F_{64}\}$$

V. 실험 및 고찰

실험은 IBM PC/AT (i80286)에서 C언어를 사용하여 시뮬레이션을 수행하였다. 표준 획은 피실험자 5명으로부터 한글 자음으로부터 시작되는 14개의 형태, 영문자 26개, 숫자10개를 각각 40개씩 필기하도록 하여 발생빈도가 높은 데이터를 기준으로 감독자 추천 방식에 의해 선정하였고, 실험에 사용된 입력 데이터는 표준 획 선정에 참여한 피실험자를 제외한 임의의 피실험자들로부터 실재하는 문장들을 테블렛에 필기하도록 하여 사용하였으며, 총 문자의 수는 10,000여자를 필기하여 인식 실험을 진행하였다. 그리고, WLD 알고리즘에 있어서, 삭제 변환 가중치 $W_d = 2$, 삽입 변환 가중치 $W_i = 3$, 대체 변환 가중치 $W_s = 5$ 로 설정하였으며, 이 가중치는 실험치이다. 또한, 변환이 발생되지 않았을 경우의 가중치는 "0"이다.

한편, WLD 알고리즘을 기능적으로 수행할 수 있도록 프로그램하여 각 코드 비교시, WLD 알고리즘에 대한 병렬성과 동시성을 볼 수 있었다. 즉, 전용 프로세서로 구성할 경우, 입력 패턴과 표준 패턴의 비교는 각 노드에서 일어나며, 각 변환에 대하여 동시성과 병렬성을 갖는다. 전체 시스템은 하드웨어에 의한 시스템 구성시, 파이프라인 처리가 될 수 있도록 하였고, 병렬처리가 가능하도록 알고리즘을 전개하였다.

본 연구에 있어서, 획의 인식률은 96.57% 이었고, 자소 및 문자의 분리율은 95.41%이었으며, 실험의 범위와 결과를 그림7과 표3에서 보인다. 오인식 또는 Reject의 주원인은 테블렛의 양자화 잡음에 기인하고 있으며, 필기자의 습관에 의한 독특한 획의 출현등도 인식 실패의 원인이 되고 있다. Reject된 문자는 분리가 불가능한 문자들이었고, 오분리된 문자는 다른 문자로 분리된 경우이다. 오분리된 문자는 "위↔유", "D↔P" 등과 같은 문자들로서, 학습성에 따른 애매성 해소가 요구되는 문제로서, "위↔유"에 있

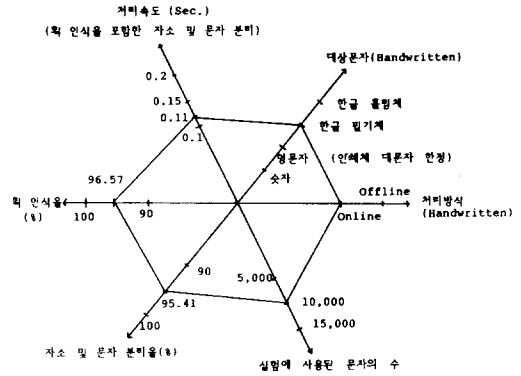


그림 7. 본 연구의 범위 및 결과
Fig. 7. The region and result of the study.

표 3. (a) 획의 인식률과
(b) 자소 및 문자의 분리율

Table 3. (a) Recognition rates of stroke, and
(b) Separation of phoneme & character.

Stroke No.	발생 빈도	정인식율	오인식율	Stroke No.	발생 빈도	정인식율	오인식율	Stroke No.	발생 빈도	정인식율	오인식율
1 (ㄷ)	228	92.42	6.56	20(L)	359	93.82	4.18	29(ㄱ)	200	86.54	1.46
2 (ㄴ)	328	94.51	4.46	21(L)	342	95.04	4.96	40(ㅋ)	212	97.21	2.79
3 (ㄹ)	218	95.41	4.59	22(ㅌ)	237	96.64	3.36	41(ㆁ)	182	90.62	9.37
4 (ㄷ)	102	94.12	3.88	23(ㄷ)	263	96.07	4.93	42(ㄴ)	429	97.42	2.57
5 (ㄹ)	144	95.83	4.17	24(ㄹ)	294	95.94	4.06	43(ㅇ)	289	97.94	2.08
6 (ㄱ)	257	95.22	4.67	25(ㄷ)	244	97.09	2.91	44(ㅅ)	449	96.22	4.88
7 (ㅅ)	250	95.6	4.4	26(ㅅ)	333	95.19	4.81	45(ㅈ)	373	94.10	5.9
8 (ㅈ)	120	95.26	4.52	27(ㅈ)	220	95.94	4.08	46(ㅊ)	291	98.14	3.84
9 (ㅊ)	105	92.22	4.87	28(ㅊ)	222	93.78	4.22	47(ㅊ)	286	100	0
10(ㅊ)	329	96.25	3.65	29(ㅊ)	202	92.71	6.29	48(ㅊ)	375	95.2	4.6
11(ㅊ)	253	95.43	4.25	30(ㅊ)	226	94.92	5.08	49(ㅊ)	103	89.22	10.88
12(ㅊ)	1414	100	0	31(ㅊ)	139	99.21	10.79	Total	15626	96.57	3.42
13(ㅊ)	1242	100	0	32(ㅊ)	209	90.42	9.57				
14(ㅊ)	242	100	0	33(ㅊ)	230	93.6	6.4				
15(ㅊ)	242	100	0	34(ㅊ)	122	92.62	7.38				
16(ㅊ)	247	100	0	35(ㅊ)	107	94.29	5.61				
17(ㅊ)	228	100	0	36(ㅊ)	208	96.57	1.42				
18(ㅊ)	230	91.21	6.79	37(ㅊ)	428	96.71	3.29				
19(ㅊ)	271	94.46	5.54	38(ㅊ)	301	96.68	3.22				

(a)

Total	Separation		Reject & Failure	
	Count	Ratio	Count	Ratio
10000	9541	95.41	459	4.59

(b)

어서, 오분리의 원인은 근본적인 형태에 있어서 가상 획의 방향이 달라 생생되나, 필기자 습관에 주로 기인하여 발생하는 형태이며, 또한 크기에 대하여 고려하여야 할 문제이고, "D↔P"의 형태에서는 끝점의 상대적 위치에 대한 고려와 크기에 대한 고려가 필요하며, 주로 필기자 습성에 따른 학습성 문제이고, 위에서 언급한 학습성에 대한 문제가 한글 및 외래

경	희	재	학	교
전	자	공	학	과

4	0	0	9	2
8	3	2	2	9

그림 8. 입력문자의 예
Fig. 8. Example of input characters.

어를 포함한 문자를 처리할 때 매우 중요하다. 따라서, 한글 및 외래어를 포함한 문자를 인식처리 할 경우, 상호 각 문자들의 유사성을 분석하여, 이에 따른 해결 방안으로 학습성의 강도를 높여 애매성을 해소하고, 인식성을 높여야 한다.

Ⅶ. 결 론

본 논문에서는 온라인 필기체 문자인식 알고리즘을 제안하였고, 전용 시스템 구성시, 알고리즘 기술이 적합한가를 고찰하였으며, 전용 시스템의 구현 가능성을 볼 수 있었다. 자소 및 문자의 구성 상태를 천이 상태 그래프를 통하여 분리 인식하여 분류하였으며, 획의 인식율은 96.57%, 자소 및 문자 분리율은 95.41%이었으며, 본 알고리즘의 유용성을 볼 수 있었다. 오인식의 주 원인은 태블렛의 잡음과 필기자의 습관에 따른 것으로, 알고리즘의 개선에 의하여 인식율은 더 향상될 수 있다. 또한 일본문자들을 포함한 기타 문자인식에 있어서는 새로운 획의 선정, 매핑 함수의 추가와 함께 문자 구조상의 중심점의 이용, 한글에 있어서 모아쓰기 알고리즘과 영문자, 숫자등의 풀어쓰기 형태의 기타 문자들과의 알고리즘 병행등에 따른 알고리즘 개선에 의하여 보다 많은 문자들을 인식할 수 있을 것이다. 또한 전용 시스템 구현시에는 DP매칭의 병렬처리와 동시 처리를 위하여 시스템용 어레이의 구조로 구성하여야 할 것이며, 유한 상태 오토메타에서 룩업 테이블을 이용하고, 동시성과 병렬성을 갖도록 구성을 하면 실시간 처리에 보다 효율성을 높일 수 있다.

한편, 본 연구를 바탕으로 현재 그림4에서 고찰한 문자열 인식을 위한 어레이 프로세서를 설계 제작중에 있다.

參 考 文 獻

- [1] K. Odaka, T. Wakahara & S. Hashimoto, "On-line handwrittern character recognizer," *J. IECE*, vol. 65-D, no. 8, pp. 951-958, Aug. 1982.
- [2] T. Wakahara, K. Odaka & 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.
- [3] S. Hanki, T. Temma, H. Yoshida, T. Arakawa, M. Suzuki, T. Seki, Y. Kikuchi, "On-line real-time character recognition system," *J. IECE*, vol. 64-D, No. 5, pp. 411-418, May. 1981.
- [5] Tzay Y. Young, King-sun Fu, *Handbook of pattern recognition and image processing*, Academic Press, London, 1986.
- [6] King-sun Fu, *Syntactic pattern recognition & applications*, Prentice-Hall, 1982.
- [6] 이희동 외3, "확장된 DP 매칭법에 의한 홀림체 한글 온라인 인식," 대한전자공학회 논문지, Vol. 26, No. 1, pp. 29-37. 1989.
- [7] 이희동, 김태균, "보강 문맥 자유 문법을 이용한 필기체 한글 온라인 인식," 대한전자공학회 논문지, Vol. 24, No. 5, pp 37-44. 1987.
- [8] 홍성민, "구조해석에 의한 필기체 한글의 온라인 인식," 경희대학교 대학원 석사학위논문, 1985, 8.
- [9] 과학기술처, CAD Software의 개발에 관한 연구 최종보고서, 과학기술처, pp. 124-146, 1987, 7.
- [10] K.W. Shin, B.Y. Chai, M.K. Lee, "A VLSI architecture of systolic array for FFT computation," 대한전자공학회 논문지, Vol. 25, No. 9, Sep, 1988.
- [11] Y.P. Chaing and K.S. Fu, "VLSI arrays for syntactic pattern recognition," in *VLSI for Pattern Recognition and Image Processing* Ed. by King-sun Fu, Springer-Verlag, 1984.
- [12] Monica S. Lam and J. Mostow, "A transformational model of VLSI systolic design," *IEEE Comput.*, Feb., 1985.

 著 者 紹 介

**金 銀 源 (正會員)**

1962年 1月 20日生. 1985年 경희대학교 전자공학과 졸업 (공학 학사). 1990年 경희대학교 대학원 전자공학과 졸업 (공학석사). 1990年~현재 경희대학교 대학원 전자공학과 박사과정 재학중. 주

관심분야는 패턴인식, VLSI설계, 영상처리분야 등으로서 실시간 처리용 프로세서 설계 분야임.

**趙 源 敬 (正會員)**

1948年 3月 19日生. 1972年 경희대학교 전자공학과 졸업 (공학 학사). 1974年 한양대학교 대학원 전자공학과 졸업 (공학석사). 1986年 한양대학교 대학원 전자공학과 졸업 (공학박사). 현재 경

희대학교 전자공학과 부교수. 주관심분야는 고속 디지털 신호처리용 VLSI설계로서 실시간 영상처리, 실시간 인식을 위한 패턴매칭, 고속 디지털 신경회로망을 위한 알고리즘과 이를 VLSI로 실현하는 연구를 하고 있다.