

구조 정보의 DP 정합에 의한 흘려 쓴 한글의 온라인 인식

(On-line Recognition of the Cursive Hangeul by DP Matching with Structural Information)

李恩主*, 朴晋烈**, 朴宰成***, 金太均****

(Eun Joo Rhee, Jeon Youl Park, Zae Sung Park and Tae Kyun Kim)

要約

본 논문에서는 DP 정합(Dynamic Programming Matching)을 이용한 흘려 쓴 한글의 온라인 인식에 관하여 논하였다. 제안한 방법에서는 입력문자와 표준문자 사이의 정합도를 나타내는 벌점(Penalty)을 한글의 구조정보인 정합하는 획의 종류, 획의 방향, 조각획의 길이비에 의하여 산정하였다. 또 이 벌점은 3×3 윈도우를 사용하여 경로탐색 과정에서 동적으로 부과하므로 최적 경로의 탐색이 가능하게 되었다. 그 결과 변형이 많이 포함된 흘려 쓴 한글 까지도 표준문자에서 가장 유사한 획의 쌍을 갖는 자모를 찾는 방향으로 정합이 유도되었다.

제안한 방법에 대한 모의실험 결과 문자단위로 입력된 정자체 한글 뿐 만 아니라 흘려 쓴 한글의 인식에서도 그 유용성을 확인할 수 있었다.

Abstract

This paper presents the algorithm of an on-line cursive Hangeul recognition by means of the DP matching with structural information. In the proposed method, the penalty represents the similarity between input character and reference character. The penalty is dynamically computed with types of strokes, directions of strokes, and the length ratios of fragment during the process that searches for an optimal path using 3×3 matrix. As the result, this method can be exactly matched for even greatly deformed characters.

The computer simulation shows that the proposed algorithm can be utilized in recognizing cursive Hangeul as well as correct Hangeul.

* 正會員, 大田産業大學校 電子計算學科
(Dept. of Computer Science, Taejon Nat'l Univ. of Technology)

** 準會員, 三元 테크(株)
(Samwon Tech Co., LTD.)

*** 準會員, 金星計電(株)
(Goldstar Instrument & Electric Co., LTD.)

**** 正會員, 忠南大學校 컴퓨터工學科
(Dept. of Computer Engineering, Chungnam Nat. Univ.)

※ 이 논문의 일부는 1991년도 교육부지원 한국학술진흥재단의 자유공모(지방대학육성)과제 학술연구조성비에 의하여 연구되었음.

接受日字 : 1993年 8月 3日

1. 서론

인간에게서 말이나 연필로 글을 쓰는 것은 매우 자연스럽고 익숙한 정보 표현 및 전달 수단이다. 컴퓨터 사용자들은 현재와 같이 키보드를 통하여 컴퓨터를 사용하는 것보다 더욱 일상적이며 익숙한 방법으로 컴퓨터를 쉽게 이용하고자 하는 욕구를 가졌고, 이를 실현하기 위한 한 방안으로 온라인 문자인식에 관한 연구가 꾸준히 진행되어 왔다.

그리하여 미국과 일본을 중심으로 선진 각국들은 많은 연구를 수행하여 필기에 제약을 갖기는 하나 다양한 상업용 시스템을 발표하였고, 획순, 획수와 필기 위치에 대한 제약을 완화하여 자유롭게 필기한 문자 인식에 관한 연구를 활발히 진행하고 있다.¹⁻³⁾

한글 인식에 관한 연구는 한글이 주로 “간단한 직선성의 획들이 모여서 한글의 기본 자모를 구성하고, 이 자모들이 2차원적으로 조합하여 문자를 이루는 계층적인 구조”를 갖는 특성을 가지고 있으므로, 그간 이 특성을 이용하여 한글을 인식하고자 하는 접근 방식인 구조해석적 방법에 의한 연구가 많이 되었고 좋은 결과를 얻고 있다.⁴⁻⁵⁾ 그러나 이 방식에 의하여 흘러 쓴 한글을 인식하기에는 흘러쓰므로 발생한 패턴의 변형과 필기자의 개인차에 의하여 나타나는 패턴의 변형을 모두 문법적으로 기술하여야 하는 어려움이 있다.

이러한 패턴의 변형을 극복하기 위한 방안으로 필기자에 의존적으로 다양한 표준 패턴을 생성하거나⁶⁾ “서로 다른 문자간의 차이가 같은 문자에서 다른 필기 형태간의 차이보다 크다”는 가설하에 입력패턴에 모든 표준패턴을 정합하는 방법론으로 DP 정합을 한글 인식에 이용하였다.⁷⁻⁹⁾

그리하여 문헌 [6]에서는 적응학습법을 도입하고 개인차에 의해 나타나는 변형을 극복하기 위하여 필기자에 의존적으로 다양한 표준문자를 생성하나, 개인차에 의해 나타나는 패턴의 변형과 흘러쓰므로 발생하는 패턴의 변형을 모두 흡수하기는 어렵다.

문헌 [7]에서는 DP 정합을 자소 추출이 가능하도록 확장하고, 한글 인식에 적용하여 좋은 결과를 보고한 바 있다. 이 방법은 입력문자와 표준문자와의 정합을 한글의 구조정보 중에서 주로 획의 방향에 의존하므로, 방향차가 큰 획의 삽입이 많은 흘러 쓴 한글의 인식에는 문제가 있다. 또한 정합에서 벌점을 경계치에 의하여 두 개의 값으로 한정하였다.

문헌 [8]에서는 획방향 및 길ियो소 가중에 의한 정합방법¹⁰⁾을 자모의 세그먼트 길이비에 적용하고, 또 자소간의 위치관계를 이용하여 한글을 인식하였

다. 한글의 자모 표현에서 자모를 구성하는 세그먼트의 상대적 길이비는 의미있는 요소이다. 그러나 방향 변환을 포함한 획과 첨가 형태의 획에서는 길이보다 획의 존재 유무가 더 의미가 있으며, 또 흘러 쓴 한글에서는 다양하게 획의 길이가 변하므로 세그먼트의 길이비를 예측할 수 없다는 문제가 있다. 그리고 경로탐색 단계에서 벌점은 획의 방향에 의하여 계산하고, 탐색이 끝난 다음 탐색된 경로에 대하여 세그먼트의 길이비를 고려해서 다시 산정하여 부과하므로, 길이비에 의한 벌점 부과가 경로 탐색에 영향을 전혀 주지 못한다.

또 이 연구들에서는¹¹⁻¹⁶⁾ 정합행렬에서 문자인식을 위한 경로 탐색이 단순히 탐색점에서 최소의 벌점을 갖는 경로를 찾으므로 국부적 최소의 문제가 발생하며, 학습능력이 없어 개인에 적용할 수 없고, 표준자모의 생성과 시스템 확장이 어렵다.

본 연구에서는 이와 같은 문제점을 해결하고 흘러 쓴 한글을 효율적으로 인식하기 위하여 한글이 갖는 구조적인 특성중 인식에 가장 유효한 정보인 획의 방향, 획내의 조각획의 길이비와 획 및 자모의 상대위치 관계를 벡터형태의 자료구조로 표현하고, DP 정합에 의하여 이들의 정합정도를 판단하여 인식하는 방법을 제안하였다. 정합과정에서 벌점의 부과는, 흘러 쓴 문자에는 획형태의 변화가 있어 표준문자에 비하여 하나의 획에 다수의 대표점이 더 추출되어 한 획이 다수의 조각획의 조합으로 나타나므로, 획형태의 변형으로 인한 문자의 변형도를 측정하여 벌점을 부과하기 위하여 입력문자와 표준문자 사이의 획의 방향 뿐 만 아니라 획을 이루고 있는 조각획들의 길이비를 이용하였다. 또 정합과정에서, 가상획은 획들의 위치관계이고 실획은 문자를 이루는 획에 관한 정보이므로 획의 종류에 따라서 상이한 벌점을 부과하여 입력문자와 표준문자의 가상획은 가상획과 실획은 실획과 자연스럽게 정합하도록 유도하였다. 이 벌점의 산정과 부과는 기존 연구에서와 같이 경로탐색 후에 하지 않고, 탐색과정에서 동적으로 하므로써 획의 방향에 의한 정합도의 보정효과 뿐 만 아니라 최적 경로탐색의 가능성을 크게 했다. 그리고 경로탐색에서 국부적 정보에 의한 탐색으로 국부적 최소에 빠지므로 발생하는 오인식을 막기 위하여 3 × 3 윈도우 내에서 문자의 형태를 대국적으로 예측하며 경로를 탐색하는 방법을 제안하였다.

이하 본 논문에서는 II장에서 전처리, III장에서 DP 정합, IV장에서 한글인식에 대하여 기술하고, V장에서 인식실험을 통하여 제안한 방법의 유용성을 검토하고, VI장에서 본 연구의 결론을 기술한다.

II. 문자의 입력 및 전처리

태블릿 위에 펜을 이용하여 기록한 문자는 펜과 태블릿의 접촉상태 정보를 포함하여 펜의 태블릿 상의 위치정보인 x, y 좌표열이 등시간 간격으로 샘플링되어 입력된다. 태블릿을 통해서 입력된 문자에는 필기자가 느끼는 태블릿과 일상 필기와의 감의 차이로 인하여 펜의 떨림 등에 의한 변형과 불필요한 정보가 많이 포함된다. 전처리에서는 인식과정 전에 입력자료에 포함된 불필요한 잡음을 제거하고 변형을 보정하는 등, 인식단에서 불필요한 정보는 줄이고 필요한 정보는 강화하므로써 인식을 향상을 도모하였다.^[6]

또 잡음이 제거된 입력 데이터에 대하여는 처리량을 줄이므로 인식 속도를 향상시키기 위하여, 많은 입력 데이터 중에서 이들을 대표하여 인식에 이용될 수 있도록 문자의 형태를 보존하는데 필수적인 좌표 데이터를 대표점이라 하고, 대표점의 추출은 최소각에 의한 대표점 추출방법을^[5] 이용하였다. 사용한 대표점은 획의 시작점, 끝점과 굴곡점이고 대표점이 찾아진 입력문자는 입력순으로 인접한 대표점을 직선으로 연결하며, 이는 시계열의 벡터가 된다.

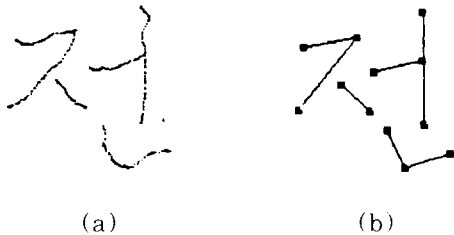


그림 1. 전처리 예
(a) 입력문자
(b) 처리 결과(■은 대표점을 표시)

Fig. 1. An example of preprocessing.
(a) input character.
(b) processed result.

III. 한글 인식을 위한 DP 정합 기법

DP 정합에 의한 문자인식은 서로 다른 문자간의 차이가 같은 문자에서 다른 필기 형태간의 차이보다 크다는 가설에 근원을 두고 있으며, 입력문자를 모든 표준문자에 정합하여 가장 작은 벌점을 갖는 표준문자를 택하는 방법론으로, 다양한 변화를 갖는 흘려 쓴 문자 인식에 매우 유효하다.^[7]

본 장에서는 한글인식에서 정합에 사용할 정합행렬의 기준정보인 입력정합벡터와 표준정합벡터의 구성과 정합행렬에 관하여 기술한다. 또 정합행렬에서 벌점의 부과방법과, 정합과정으로 정합행렬의 첫번째 요소에서 마지막 요소에 이르는 가장 작은 벌점을 갖는 가능한 최적 경로를 탐색하는 기법 등, 흘려 쓴 한글 인식을 위하여 제안한 DP 정합 기법 전반에 대하여 논한다.

정의 1은 정합과정에서 사용되는 획에 관련된 용어들이고, 그림 2는 정의 1에 대한 예로써, (a)에서 a, b, c, d, e, f, g는 대표점을 나타내고, (b)의 “ㄱ”에는 정자로 필기했을 때 외에 a1, b1 등의 대표점이 더 추출되었다. (a)에서 ab, bc, de, fg는 획임과 동시에 실획이고, cd, ef의 점선은 가상획이다. (b)에서 aal, alb는 획 ab의 조각획이고, bbl, bic는 획 bc의 조각획이다.

- [정의 1] 정합과정에서 사용되는 획에 관련된 용어
- ① 획 : 정자로 필기된 자모에서 대표점 사이의 펜의 궤적을 획이라 한다.
 - ② 조각획 : 흘림체 한글에서 획의 변형은 획내에 다수의 대표점을 발생시켜 하나의 획이 다수의 조각들로 나누어 진 경우, 대표점 사이의 각 부분획을 조각획이라 한다.
 - ③ 실획 : 펜이 태블릿에 닿아서 움직인 펜의 궤적
 - ④ 가상획(dark-stroke) : 필기시 펜이 태블릿에 닿지 않고 이동한 문자의 부분을 가상획이라 하며, 두 획 사이의 위치관계를 나타낸다.

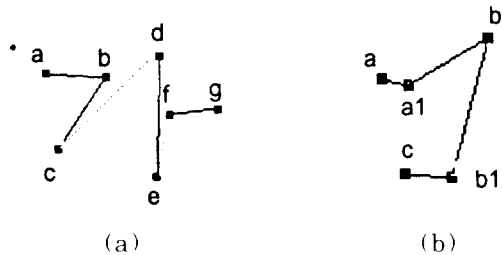


그림 2. 획의 예
(a) 정자 “가” (b) 흘려 쓴 “ㄱ”
Fig. 2. Examples of strokes.
(a) correct character “가”,
(b) cursive phoneme “ㄱ”.

1. 입력정합벡터와 표준정합벡터의 구성

입력정합벡터와 표준정합벡터는 입력된 문자와 표준문자에 대하여 대표점을 구하고 대표점들을 잇는

각각의 선분에 방향코드를 부여한 다음, 여기에 정합에 필요한 정보를 첨가한 벡터형태의 자료구조로, 각각 2차원 정합행렬의 행과 열의 기준정보가 된다.

1) 입력정합벡터

입력정합벡터는 입력문자에 대하여 대표점을 추출하고 대표점 사이를 직선으로 연결하여 16방향 체인코드를 부여한 방향벡터에 펜의 상태, 이 선분의 길이와 대표점의 위치 좌표가 첨가되어 구성된다. 또 필기시 획이 분리된 곳도 필기 순서에 따라서 펜업에서 다음의 펜다운까지가 가상획에 의하여 연결되어 있다고 가정하고, 이 두 점 사이를 잇는 가상획에 대하여도 같은 형태로 입력정합벡터에 포함하며, 입력정합벡터는 정의 2와 같다.

[정의 2] 입력정합벡터는 정합행렬인 MATRIX [INPUT, REFERENCE] 의 열의 기준정보로, 입력 정합벡터 INPUT은 아래와 같다.

INPUT = { I_ST₁, I_ST₂ ... I_ST_i ... I_ST_n }

I_ST_i = { STA, DIR, DIS, POS }

STA : 펜의 상태로, 1은 펜다운을 0은 펜업 상태를 표현

DIR : 각 대표점 사이를 직선으로 연결하고, 방향에 따라 부여된 16방향 체인코드

DIS : 두 대표점 사이의 거리

POS : 대표점의 x, y 좌표

i = 1, 2, ... n

n : 입력문자에 포함된 대표점의 총수

2) 표준정합벡터

한글은 기본자모가 2차원적으로 조합하여 문자를 이루는 모아 쓰기를 한다. 이미 한글은 모아 쓰는 형태에 따라서 한글의 유형과 자모의 위치관계가 잘 정형화되어 있다. 한글인식은 조합가능한 모든 문자를 인식대상으로 하지 않고, 이 정보를 이용하여 문자를 구성하는 자모단위로 하는 것이 인식시간과 시스템 구성면에서 매우 유리하다. 그리하여 인식을 위한 표준문자는 초성, 중성, 종성 각각에 자모단위로 가지고 있으며, 또 초성과 중성 사이와 중성과 종성 사이의 가상획을 포함하고 있다. 표준문자는 그림 3과 같이 초성이 19, 중성 21, 종성 27개를 기본으로 하여, 필기시 현격한 필순 및 형태의 변형을 갖는 자모들을 여기에 첨가하였다.

표준정합벡터는 표준문자에 포함된 자모들을 정자체로 필기하여 전처리와 대표점 추출 단계를 거친 후 입력정합벡터의 구성에서와 같이 필요한 정보를 획득하여 구성하며 정의 3과 같다.

초성	단자용	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㆁ ㅅㅇ	14
중성	중자용	ㄱ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㆁ ㅅㅇ	5
중성	단모음	ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ	10
중성	중모음	ㅘ ㅙ ㅚ ㅜ ㅝ ㅞ ㅟ ㅠ ㅡ ㅢ	11
중성	단자용	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㆁ ㅅㅇ	14
중성	중자용	ㄱ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㆁ ㅅㅇ	13

그림 3. 표준문자

Fig. 3. Reference characters.

[정의 3] 표준정합벡터

REFERENCE = { FIR_i, DRK_{ij}, MID_j, DRK_{jk}, LST_k }

FIR_i = { R_ST_{i1}, R_ST_{i2} ... R_ST_{ij} ... R_ST_{in} }

DRK_{ij} = { R_ST_{ij} }

MID_j = { R_ST_{j1}, R_ST_{j2} ... R_ST_{jk} ... R_ST_{jp} }

DRK_{jk} = { R_ST_{jk} }

LST_k = { R_ST_{k1}, R_ST_{k2} ... R_ST_{ki} ... R_ST_{kp} }

R_ST_{ij} = { STA, DIR, DIS, POS }

여기서, FIR_i는 i번째 초성 표준문자, MID_j는 j번째 중성 표준문자, LST_k는 k번째 종성 표준문자, DRK_{ij}는 i번째 초성 표준문자와 j번째 중성 표준문자를 잇는 가상획이고, DRK_{jk}는 j번째 중성 표준문자와 k번째 종성 표준문자를 잇는 가상획이다. R_ST_{xy}에서 STA, DIR, DIS, POS는 정의 2와 같고, 첨자 x는 i, j, k중의 하나이며, y는 해당 표준문자에서 임의의 대표점 번호이다. 또 n, o, p는 정합대상 문자의 초성, 중성, 종성의 각 표준문자에 포함된 대표점의 수이다.

3) 정합행렬

정합행렬은 입력정합벡터를 행의 기준정보로 표준정합벡터를 열의 기준정보로 하여 2차원 배열로 구성

표준정합벡터		
인식대상	실획	가상획
	실획	$P'_{i,j} = P_{i,j}$
가상획	$P'_{i,j} = (P_{i,j} + 1) \times 5$	$P'_{i,j} = P_{i,j} \times 1.2$

$P_{ij} = [\text{MIN}\{(I_ST_i(\text{DIR}) - R_ST_j(\text{DIR})), (16 - (I_ST_i(\text{DIR}) - R_ST_j(\text{DIR})))\}]^2 + 1$ 로, 입력정합벡터와 표준정합벡터 사이의 방향차에 의한 벌점.

P'_{ij} : P_{ij} 가 정합 대상 획의 종류에 따라서 변화된 벌점.

그림 4. 벌점 부과

Fig. 4. Penalty computation.

되고, 각각의 교차점에는 입력문자와 표준문자의 상이도인 벌점을 가진다. 또 벌점은 입력문자와 표준문자의 정합에서 가상획은 가상획, 그리고 실획은 실획과 자연스럽게 정합하도록 유도하기 위하여, 획의 종류에 따라서 그림 4와 같이 방향차에 의한 벌점에 실험에 의하여 얻은 각각 다른 크기의 가중치를 주어 부과하였다. 그림 5는 정합행렬이고, 행렬의 요소인 "P_{i,jk}"는 입력정합벡터 I-ST_i(입력문자에서 i번째의 획)와 표준정합벡터 R-ST_k(중성의 j번째 표준문자에서 x번째 획) 사이의 벌점이다.

입력정합벡터 \ 표준정합벡터	...	Dar _{1j}	R-ST _{1j}	...	R-ST _{jk}	...
I-ST _i	...	P _{i,j1}	P _{i,j1}	...	P _{i,jk}	...
...
I-ST _i	...	P _{i,j1}	P _{i,j1}	...	P _{i,jk}	...
...

그림 5. 정합행렬
Fig. 5. Matching matrix.

3. 경로 찾기

경로 찾기란 정합행렬의 처음 요소에서 행렬의 마지막 요소에 도달하는 벌점의 합을 최소화 하는 경로를 찾는 것이다. 이 과정은 비교하는 두 문자에서 가장 유사한 획의 쌍들을 찾는 것이 목적으로, 비교대상 문자 사이에서 가장 유사한 획의 쌍들을 찾은 경우가 벌점의 합이 최소가 된다.

정합행렬에서 정합 경로를 찾기 위하여 현재의 위치인 MATRIX [i,j] 에서 가능한 이동은 정의 4와 같이 3가지로 정의한다. 여기에서 삽입 이동은 표준문자의 하나의 획에 다수개의 입력획이 대응된 것으로, 흘려쓰므로 획의 형태가 변하여 하나의 획에 다수의 대표점이 생긴 경우이다. 삭제 이동은 다수의 표준문자의 획들이 하나의 입력획에 대응된 것으로, 이 경우는 필기의 편이를 위하여 획들을 붙여 쓰거나 생략한 경우이다. 정합 이동은 입력문자와 표준문자의 획이 1대 1로 대응된 경우이다.

[정의 4] 정합행렬에서의 이동

- ① 삽입이동 : MATRIX [i,j] → MATRIX [i+1,j]
- ② 삭제이동 : MATRIX [i,j] → MATRIX [i,j+1]
- ③ 정합이동 : MATRIX [i,j] → MATRIX [i+1,j+1]

경로 찾기에서 현재의 위치에서 다음 상태로의 이

동은 그림 6에서 보인 3 × 3 윈도우를 사용하였다. 이 윈도우에서 정의 5를 이용하여 이동방향에 따라서 예상되는 벌점을 계산하고, 최소의 벌점을 갖는 경로를 택하여 정의 4와 같이 이동한다. 경로탐색에서 기존의 연구와 [7]과 같이 단순히 현재의 지점에서의 두 획 사이의 상이도를 최소화하는 경로를 택하면 국부적 최소에 빠져 오인식의 위험이 있고, 윈도우를 크게 설정하면 최적경로는 선택되나 탐색시간이 많이 걸린다.

그리하여 본 논문에서는, 한글을 흘려쓰므로 생기는 패턴의 변형이 한 획에 한 개 내지 두 개의 대표점이 더 추출되는 경우와, 두 획 내지 세 획이 한 획으로 표현된 경우가 대부분이므로 계산량도 많지 않고 변형된 한글에 적합한 3 × 3의 윈도우를 이용하여 경로 탐색을 하므로 이 문제들을 해결하였다. 그림 7에서는 "한"에 대한 정합행렬에서 탐색된 정합경로를 보였고, 기본적인 탐색과정은 문헌 [7]을 참조하기 바란다.

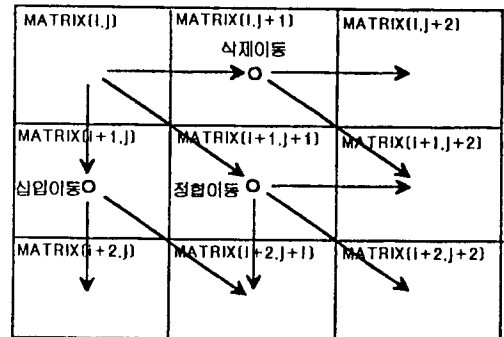


그림 6. 최적 경로탐색을 위한 3 × 3 윈도우
Fig. 6. 3×3 window for optimal path search.

[정의 5] 최적 경로탐색을 위한 벌점 계산

- ① 삽입이동 벌점 = 1.4 × (Pel(MATRIX [i+1, j]) + Min(Pel(MATRIX [i+2, j]), Pel(MATRIX [i+2,j+1])))
- ② 삭제이동 벌점 = 1.4 × (Pel(MATRIX [i, j+1]) + Min(Pel(MATRIX [i, j+2]), Pel(MATRIX [i+1, j+2])))
- ③ 정합이동 벌점 = Pel(MATRIX [i+1, j+1]) + Min(Pel(MATRIX [i+1, j+2]), Pel(MATRIX [i+2, j+2]), Pel(MATRIX [i+2, j+1]))

Pel(MATRIX [i,j]) : 정합행렬에서 i번째 행과 j번째 열의 벌점.

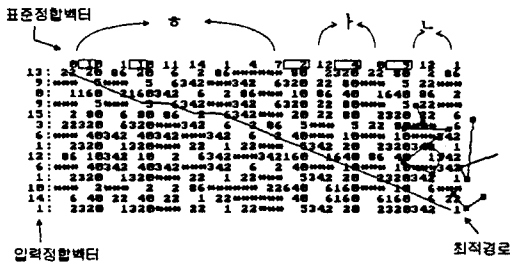


그림 7. "한"의 정합 경로
Fig. 7. Matching path of "한".

4. 획의 길이에 의한 벌점

한글은 흘려 쓴 때에는 종종 획의 시작 부분과 끝 부분을 위주로하여 획내에 길이가 짧으면서도 표준문자와 방향차가 큰 획들이 첨가되고, 또 직선성의 획들이 곡선으로 표현되어 획내에 다수의 대표점이 나타난다. 이와 같이 변형된 문자에서의 벌점은 다수의 입력획과 하나의 표준획 사이에서 방향차의 합이 되므로 입력획들이 조합하여 하나의 표준획이 될 수 없음을 나타내는 벌점이 큰 값이 되고 입력획에 방향차가 큰 획이 포함된 경우는 더욱 그러하며, 이것은 오인식의 요인이 된다. 그러나 실제로는 길이가 짧고 방향차가 큰 획이 획의 시작과 끝 부위에 삽입된 경우는, 이 삽입획은 방향차는 크나 길이가 짧기 때문에 획의 전체적인 모양이 삽입으로 인하여 큰 영향을 받지 않는다. 또 획이 곡선적으로 필기되어 한 획에서 다수의 대표점이 추출되어 유사한 방향을 갖는 획의 조합으로 나타난 경우도 각각의 획들은 자신의 길이 만큼 획의 모양을 변화시키므로, 이로 인하여 입력문자가 오인식 되어서는 안된다는 것이다.

그러하여 본 연구에서는 정합과정에서 벌점 부과에 획의 방향 뿐 만 아니라 입력획을 구성하는 조각획의 길이비 개념을 도입하였다. 이것은 하나의 획이 다수의 조각획 형태로 입력될 때, 이들은 획의 요소들로서 자신의 길이비에 비례하여 자모구성에 기여한다는 것이다. 그러므로 표준획과 입력획 사이의 벌점은 표준획과 조각획들과의 방향차에 의하여 계산하고, 계산된 벌점은 조각획의 길이비에 따라서 정의 6에 의하여 다시 산정하여 부과된다.

제한한 획의 방향과 길이비에 의한 벌점 부과는 하나의 획이 다수의 조각획으로 표현된 경우와, 또 표준문자와 방향차가 큰 획의 삽입이 문자 인식에 미치는 영향을 줄이는 효과가 있다. 또한 같은 이유로 기존의 연구에서 획의 시작과 끝부분에서 발생하는 빼침획(Hook)을 문자의 전체적인 정보 없이 삭제하던

무모함을 극복할 수 있게 되었다.

그림 8에서는 자모 "ㄱ"의 변형예를 보였다. 여기서 $L_{11} \sim L_{22}$ 는 입력획에서 조각획들의 길이이고, $D_{11} \sim D_{22}$ 는 조각획의 방향코드, D_1, D_2 는 표준자모에서 획들의 방향코드이다. 식 (1)에서 P_D 는 입력자모와 표준자모와의 획의 방향차에 의하여 계산된 벌점이고, P_L 는 획의 방향과 거리에 의하여 계산된 벌점이다.

$$L_1 = L_{11} + L_{12}, \quad L_2 = L_{21} + L_{22}$$

$$P_D = \sum_{i=1}^2 \sum_{j=1}^2 ((D_{ij} - D_i)^2 + 1)$$

$$P_L = \sum_{i=1}^2 \sum_{j=1}^2 (((D_{ij} - D_i)^2 + 1) \cdot (L_{ij}/L_i)^{1/2}) \quad (1)$$

[정의 6] 획의 길이비에 의한 벌점

$$P = (P_1 \times (L_1/L)^{1/2}) + (P_2 \times (L_2/L)^{1/2}) + \dots + (P_n \times (L_n/L)^{1/2})$$

P : 획의 방향과 길이비에 의하여 계산된 벌점

$L_1, L_2 \dots L_n$: 입력획을 구성하는 조각획의 길이

$P_1, P_2 \dots P_n$: 하나의 표준획과 정합된 조각획 사이의 방향차에 의한 벌점

$$L = L_1 + L_2 + \dots + L_n$$

n : 하나의 표준획에 정합된 조각획의 수

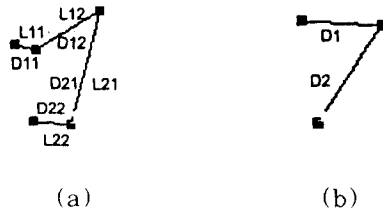


그림 8. "ㄱ"의 변형

(a) 변형된 자모 "ㄱ" (b) 정자체 "ㄱ"

Fig. 8. Deformation of "ㄱ".

(a) deformed "ㄱ", (b) correct "ㄱ".

IV. 한글 인식

본 장에서는 DP 정합을 이용한 한글인식 과정을 설명한다. 한글의 인식은 그림 9와 같이, 입력문자에 대하여 잡음을 제거하고 대표점을 찾은 다음 앞장에서 언급한 방법에 의하여 초성, 중성, 종성순으로 수행된다.

한글의 초성 인식 과정에서는 입력문자와 초성의

표준문자에 의하여 구성된 정합행렬에서 벌점이 작은 초성 N개를 후보로 선택한다. 중성 인식에서는 입력 정합행렬에서 이미 사용한 초성을 제외한 부분과 중성의 표준자모에 의하여 구성된 정합행렬에서, 초성 후보로 분류된 N개에 대하여 각각 M개의 중성을 후보로 선택한다. 중성 인식은 이미 분류된 (N × M) 개의 초·중성 후보문자에 대하여 입력정합행렬에서 사용한 초성 중성을 제외한 부분과 중성의 표준문자에 의하여 구성된 정합행렬에서 이루어 진다. 최종 인식 결과는 정합과정을 통하여 정합경로가 정합행렬의 마지막에 도달하였을 때, 부과된 벌점이 가장 작은 문자이며, 또 작기 순으로 k개는 후보문자가 된다.

인식과정에서 가상획이 갖는 위치정보 만으로 구분되지 않는 문자들은 기존의 연구에서 정의한^{4,5} 자모 사이의 위치관계를 이용하였다. 인식과정 후 인식된 문자에 대하여, 문자를 구성하는 획수에 비하여 상대적으로 큰 벌점을 갖는 문자는 기울어진 문자로 판정하고, 바로 세우기를 하여 다시 정합을 하였다. 바로세우기를 한 문자에 대해서는 바로세우기 전후의 벌점을 비교하여 작은 값을 갖는 편을 최종 인식결과로 택하였다. 바로세우기는 문자를 구성하는 획의 길이를 16방향별로 누적하고 누적된 값에 따라 기울어짐을 판정하는 방법으로 문헌 [10]를 통하여 이미 보고하였다.

초성과 중성인식 과정에서 정합행렬의 구성은 정합과정에서 삽입이동이 생기면 하나의 입력획을 정합행렬에 추가하고, 삭제이동이 일어나면 하나의 입력획을 정합행렬에서 제외시키므로 입력획과 삭제획의 수에 따라서 동적으로 구성된다.⁹

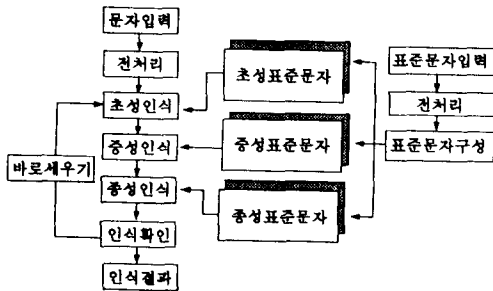


그림 9. 인식 과정
Fig. 9. Recognition procedure.

V. 실험 및 고찰

제안한 방법들의 유용성을 검토하고 시스템 보완을

위하여 인식실험을 하였다. 제안한 방법들에 대한 실험은 PC-386상에서 C언어로 하였으며, 입력 장치는 정밀도 0.5mm, 초당 100점의 표본화 속도를 갖는 WACOM SD-510C 태블릿을 사용하였다. 실험 데이타는 3인으로 부터 각각 300자씩 900자를 입력받아 실험하였다. 필기는 필기자에게 노트에 필기하듯 자연스럽게 하도록 말하고, 몇 차례의 필기 연습을 한 후 사각형 내에 한 문자씩 구분하여 연속적으로 필기하게 하였다.

표준정합행렬 구성을 위한 표준문자는 초성 25개, 중성 21개, 중성이 45개이고, 여기에는 정자체로 필기한 자모외에 필순이나 형태가 현격하게 변형된 자모를 포함하고 있다. 또 본 시스템은 사용자 적응성을 위하여 자신의 필체에 따라서 시스템을 훈련시킬 수 있는 학습 기능이 있다.

본 시스템에서는 정합과정을 통하여 초성후보는 7개, 중성은 각 초성에 대해 4개씩을 선택하였으며, 최종 10개의 문자를 선택하였다. 인식실험 결과 문자당 1초의 인식 속도를 보였고, 자모내 흘러 쓴 한글에서 90%, 자모간 흘러 쓴 문자에서 86%의 정인식률을, 후보문자를 포함한 경우는 98%의 인식률을 얻었다.

오인식의 원인은 대표점 추출 단계에서 필요한 대표점을 많이 소실한 경우이며, 이 문제를 해결하기 위하여는 흘러 쓴 한글에 적합한 전처리 방법이 필요하다. 또 다른 원인은 표준자모에 없는 자모의 입력이다. 이 경우는 표준문자의 확장이 필요하나, 표준문자의 확장은 기존의 인식 시스템에 나쁜 영향을 미칠 수 있기 때문에 표준문자를 신중하게 선택하여야 한다. 오인식의 가장 큰 원인은 획방향의 애매모호함

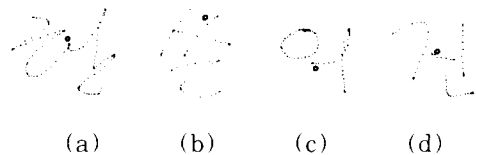


그림 10. 오인식 문자의 예
(a) 대표점 소실 (b) 표준문자에 정의되지 않은 자모 (c) 획 위치의 애매모호 (d) 획 방향의 애매모호

Fig. 10. Examples of wrong recognition.
(a) loss of feature, (b) not yet defined phoneme, (c) ambiguous stroke position, (d) ambiguous stroke direction.

과, 자모내 획 및 문자내 자모위치의 애매모호함이며 육안으로도 식별하기 힘든 경우가 많다. 그림 10의 (a)에서는 “ㅇ”부분의 대표점이 전처리단에서 소실되어 “란”으로 오인식되고, (b)는 “쓰”에서 “ㅇ”을 표시한 획이 가장 늦게 쓰인 경우로 필기순서에 의한 오류이다. (c)에서는 “ㅇ”을 표시한 획의 위치문제로 인한 오인식으로, “유”와 “의”등에서도 같은 문제가 발생하고, (d)에서는 “ㅇ” 표시를 한 획의 방향에 관한 문제로 “그”과 “즈”과 종방향의 모음이 결합될 때와 “유”와 “웃”등에서 생기는 문제이다. 그림 11에서는 정인식된 문자의 예를 보였고, 그림 12는 문자 인식의 예이다.

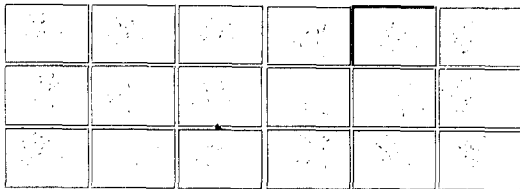


그림 11. 정인식된 문자의 예
Fig. 11. Examples of correct recognition.

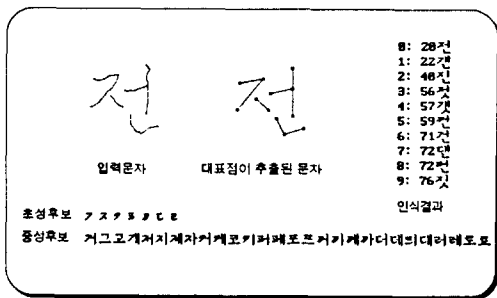


그림 12. 문자인식 예
Fig. 12. Example of character recognition.

Ⅶ. 결론

본 논문에서는 DP 정합을 이용한 흘려 쓴 한글의 온라인 인식 알고리즘에 관하여 논하였다. 정합과정에서 입력문자와 표준문자 사이의 상이도를 나타내는 별점은 획의 방향 뿐만 아니라 획내의 조각획의 길이비에 의하여 부과하였다. 그러므로 흘려 쓴 한글에서 흔히 발생하는 획의 시작부분과 끝부분의 빠침이

나, 획의 곡선적 표현에 의한 획의 변형등을 유연하게 수용할 수 있었다. 또 정합과정에서 실패와 가상 획 등 정합되는 획의 종류에 따라서 상이한 별점을 부과하므로 입력문자와 표준문자 사이에서 각 획들이 그들의 형태와 문자 내에서의 역할이 가장 유사한 획의 쌍을 찾는 방향으로 자연스럽게 유도되었다. 특히 별점의 부과는 경로탐색 과정에서 하고, 경로탐색에서 국부적 최소에 빠지는 오류를 막기 위하여 3 × 3 윈도우에 의한 탐색기법을 제안하므로 최적 경로의 탐색률을 높여 오인식의 요인을 줄였다. 본 알고리즘에 의한 인식실험 결과 문자단위로 입력된 정자체 한글 뿐 만 아니라, 자모내 및 자모간 흘림이 허용된 흘려 쓴 한글의 인식에서도 그 유용성을 확인할 수 있었다.

앞으로 남은 과제는 오인식의 가장 큰 원인인 위치나 방향이 애매모호한 획을 가진 문자에 대한 처리 방법과 속도 개선이 과제로 남아 있고, 문자단위가 아닌 연속필기 문자 인식에 관한 연구와, 한글 뿐 만 아니라 한자 영문자를 통합적으로 인식하려는 연구를 계획하고 있다.

參考文獻

[1] C. C. Tappert, C. Y. Suen, and T. Wakahara "The State of the Art in On-line Recognition of Handwriting Recognition", *IEEE Transaction on PAMI*, Vol. 12, No. 8, pp. 787-808, 1990. 8.

[2] C. Comoda, M. Nakagawa and N. Takahashi, "Grammatical Representation of Hierarchical Structure of Kanji pattern and Its Advantage for On-Line Recognition of Simplified Distorted and Wrong Stroke Order Pattern", *日本電子情報通信學會論文誌*, Vol. J70-D, No. 4, pp. 777-784, 1987. 4.

[3] J. C. Simon and O. Baret "Cursive Word Recognition", *International Workshop on Frontiers in Handwriting Recognition*, Chateau de Bonas, France, pp. 1-20, 1991. 9.

[4] T. K. Kim, T. Agui and M. Nakajima, "On-line Recognition of Cursive Korean Characters using Attributed Grammar", *日本電子情報通信學會論文誌*, Vol. J68-D,

No. 4. pp. 798-804, 1985. 4.

[5] 이 은주, 김 태균, "한글에 적합한 획 해석에 의한 필기 한글의 On-line 인식에 관한 연구", 정보과학회논문지, B권, 제 3호 pp. 171-181, 1988.

[6] B. M. Jung, T. K. Kim, T. Agui and M. Nakajima, "An Adaptive Learning and Recognition of On-line Input Characters Using Grammartical Inference", 日本 電子情報通 信學會論文誌, Vol. J75-DII, No. 1, pp. 48-57, 1992. 1.

[7] 이회동, 김태균, T. Agui and M. Nakajima, "확장된 DP 매칭법에 의한 흘림체 한글의 온라인 인식", 전자공학회논문지, 제26권, 제1호, pp. 103-111, 1989. 1.

[8] 심동규, 함영국, 박래홍, "DP 매칭과 퍼지 이론을 이용한 흘림체 온라인 한글의 인식", 전자공학회논문지, 제30권, B편, 제4호, pp. 116-129, 1993. 4.

[9] 박재성, 이은주, 김태균외 3인, "획방향과 길 이요소 가중 DP 매칭에 의한 흘려 쓴 한글의 인식", 인공지능, 신경망 및 퍼지시스템 종합 학술대회 발표논문집, 서울, pp. 170-179, 1992. 10.

[10] 이향미, 이은주, 김태균외 3인, "온라인 한글 인식을 위한 전처리 기법에 관한 연구", 인공지능, 신경망 및 퍼지시스템 종합학술대회 발표논문집, 서울, pp. 180-189, 1992. 10.

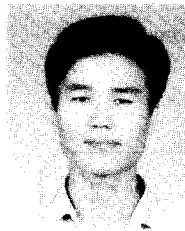
著 者 紹 介



李 恩 主(正會員)

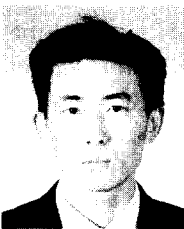
1980年 10月 충남대학교 전자공학과 졸업(공학사). 1984年 2月 충남대학교 대학원 석사과정 졸업(공학석사). 1989年 2月 동대학원 박사과정 졸업(공학박사). 1989年 8月 ~ 대전산업대학교 전자계산

학과 부교수. 주관심 분야는 패턴인식, 영상처리, 인공지능 등임.



朴 晉 烈(準會員)

1994年 2月 대전산업대학교 전자계산학과 졸업(공학사). 1993年 1月 ~ 삼원 테크(주) 근무. 주관심 분야는 문자인식, 폰트 디자인, 컴퓨터 그래픽스 등임.



朴 宰 成(準會員)

1992年 2月 충남대학교 컴퓨터공학과 졸업(공학사). 1994年 2月 충남대학교 대학원 석사과정 졸업(공학석사). 1994年 2月 ~ 금성계전(주) 근무. 주관심 분야는 문자인식, 멀티미디어, 영상처리 등

임.

金 太 均(正會員) 第 26卷 第 12號 參照

현재 충남대학교 컴퓨터공학과 교수이며, 충남대학교 공과대학 학장