

## 퍼지 결정 트리를 이용한 온라인 필기 문자의 계층적 인식

### (An Application of Fuzzy Decision Trees for Hierarchical Recognition of Handwriting Symbols)

全炳煥\*, 金成勳\*\*, 金在熹\*\*

(Byung Hwan Jun, Seong Hoon Kim and Jai Hie Kim)

#### 要約

온라인 필기 문자 인식 시스템 'SCRIPT (Symbol/Character Recognition In Pen-based Technology)'는 조합 가능한 모든 한글과 영어 대문자, 숫자, 그리고 키보드 부호 등 자연스럽게 필기되는 정자체 문자를 인식하기 위한 알고리즘이다. 필기 문자는 동일인이 쓰더라도 형태의 변화가 다양해서 정보의 불확실성을 지니게 된다. 그런데 기존의 결정 트리(decision tree)를 이용한 특징 분석 방법(feature analysis approach)은 효율적이지만 필기의 변형에 약하여 잘못된 선택을 하기 쉽기 때문에, 이러한 단점을 보완할 수 있는 방법이 필요하다. 이 논문에서는 패턴의 계층적(hierarchical) 특성에 맞추어 획 자체의 형태와 획간의 위치 관계를 파악하기 위한 두 단계의 퍼지 결정 트리(fuzzy decision trees)를 사용하여 문자 패턴의 특징을 분석하는 방법을 제안한다. 이러한 방법은, 다양한 가능성을 저장함으로써 형태의 변형에 강하고 이전의 잘못된 선택을 수정하기 쉬우며, 특히 하위 후보 패턴들에 의한 상위 패턴의 인식을 상승 효과가 매우 크다. 실험 결과, 한글은 약 91%의 인식률과 약 0.33초의 인식 속도를 나타냈으며, 영어 및 기타 문자는 약 95%의 인식률과 약 0.08초의 인식 속도를 보였다. 이는 퍼지 결정 트리를 적용하지 않은 경우에 비하여 인식률이 8~18% 정도 향상된 것이다.

#### Abstract

SCRIPT (Symbol/Character Recognition In Pen-based Technology) is an algorithm for on-line recognition of handwriting Hangeul, English uppercase letters, decimal digits, and some keyboard symbols. The shape of handwriting symbols has a large variation even when written by the same person. Though the feature analysis approach using a conventional decision tree is efficient, it is not robust under shape variations and prone to misclassification. Thus, a new method to overcome this shortcoming is necessary. In this paper, a feature analysis algorithm using two fuzzy decision trees which utilize the hierarchical property of the pattern is proposed. The first tree is used to represent the stroke shape, and the other tree is used to represent the relation between the strokes. Since this method stores various possibilities, it is robust to shape variations and can readily modify false selections. In addition, there is a large increase in the recognition rate of high-level patterns due to low-level candidates. Experimental results show 91% recognition rate for Hangeul at the recognition speed of 0.33 second per character, and the recognition rate of alphanumerics and some keyboard symbols is 95% at 0.08 second per symbol. This is 8 ~ 18% increase in the recognition rate over the method not applying fuzzy decision trees.

#### 1. 서론

\* 準會員, \*\* 正會員, 延世大學校 電子工學科  
(Dept. of Elec. Eng., Yonsei Univ.)  
接受日字: 1993年 7月 2日

최근 펜 컴퓨터의 등장으로 온라인(on-line) 필기  
문자 인식의 필요성이 부각되고 있으며, 원형 매칭

방법(prototype matching approach), 구문 해석 방법(syntactic analysis approach), 신경망 방법(neural network approach), 통계적 방법(statistical approach), 퍼지 추론 방법(fuzzy inference approach), 그리고 특징 분석 방법(feature analysis approach) 등 다각적인 접근 방법으로 연구가 활발히 진행되고 있다.

원형 매칭에서는 주로 동적 패턴 매칭(dynamic pattern matching)을 사용하여 홀림체와 같이 형태의 변형이 심한 경우에도 유연하게 매칭하지만, 매칭 전에 패턴의 구분(segmentation)이 선행되어야 하고 많은 저장 공간과 처리 시간을 필요로 한다.<sup>1,2</sup> 구문 해석은 정형화된 문법(grammar)으로 패턴을 표현하고 추론하는 방법인데, 패턴의 변형이 심할 경우에는 생성 규칙의 수가 크게 증가하여 문법의 설계가 복잡해지고 인식률도 떨어지는 단점이 있다.<sup>3,4</sup> 또한, 신경망 방법은 학습(learning)에 전적으로 의존하기 때문에 불충분하거나 잘못된 샘플로 인한 에러의 위험이 있으며, 뉴럴 컴퓨터가 아니면 처리 시간도 많이 걸린다.<sup>5,6</sup> 통계적 정보를 사용하는 은닉 마르코프 모델링(hidden Markov modelling) 방법은 변형이 매우 심하여 구조적 정보가 미흡한 대상을 인식하는데 적합하지만, 처리 시간이 매우 오래 걸린다.<sup>7,8</sup>

이와 달리, 특징 분석은 각 패턴들이 지니는 고유 특성을 나타내고 다른 패턴과의 구분에 적합한 특징(feature)을 추출하여 입력 패턴을 분류해내는 방법이다.<sup>9,10</sup> 주로 결정 트리(decision tree)를 사용하여 복잡하고 전역적인(global) 결정을 간단하고 지역적인(local) 연속된 부분 결정들로 구성하는데, 이는 각 단계에서의 결정 영역을 간단하게 하고 빠른 결정을 내릴 수 있게 한다. 그러나 잘못된 선택시 치명적인 에러를 유발하며 이전의 선택을 변경하기도 어렵다. 이를 보완하기 위해 패턴의 특징을 퍼지화 데이터로 표현하고 IF-THEN의 생성 규칙으로 추론하는 퍼지 추론 방법의 장점을 활용한다.<sup>11,12</sup> 즉, 패턴의 수에 비례하여 증가하는 생성 규칙의 사용은 지양하고, 특징을 퍼지화하여 변형을 흡수하는 장점만을 취하는 것이 바람직하다.

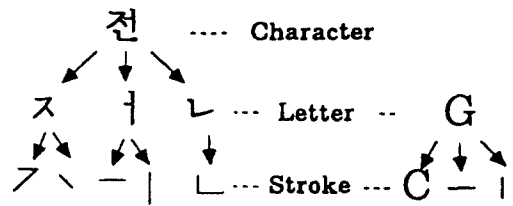
이 논문에서 제안하는 필기 문자 인식 시스템 'SCRIPT(Symbol/Character Recognition In Pen-based Technology)'는 퍼지 결정 트리(fuzzy decision trees)를 계층적(hierarchical)으로 구성하여, 조합 가능한 모든 한글과 영어 대문자, 숫자, 그리고 키보드 부호 등 자연스럽게 필기되는 정자체를 대상으로 실용적인 인식 성능을 갖는 알고리즘이다. 즉, 결정 트리에 의한 특징 분석 방법을 기본으로 다

양한 특징을 효율적으로 운영하되, 변형에 약하고 잘못된 선택으로 인한 에러를 줄이기 위해 퍼지 멤버쉽 함수(fuzzy membership function)를 도입하여 사용 특징을 퍼지화한다. 이렇게 구성된 퍼지 결정 트리는 형태의 변형에 강할 뿐 아니라 다양한 가능성을 저장함으로써 이전의 잘못된 선택을 수정하기 쉽다. 또한 문자 패턴의 계층적 구조(hierarchical structure)에 적합하게 획 자체의 인식과 획간의 위치 관계 파악을 위한 두 단계의 퍼지 결정 트리를 계층적으로 구성함으로써, 하위 후보 패턴에 의한 상위 패턴의 인식을 향상 효과가 매우 크게 된다.

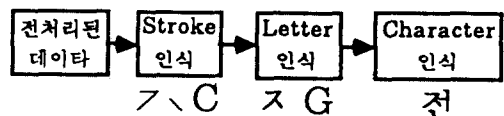
II. 인식 대상의 계층적 구조

문자를 구성하는 최소 기본 단위는 '획(stroke)'으로 펜이 바닥에 닿아서(pen-down) 이로부터 떨어질 때(pen-up)까지의 연속된 좌표로 정의한다.<sup>13</sup> 문자가 지니는 구조적인 특성에 따라 한글과 영어 및 기타 문자로 나누어 인식하는데, 이는 영어 및 기타 문자가 4개 이내의 획으로 한 문자가 완성되는 반면에, 한글은 4개 이내의 획을 갖는 자소들이 문법체계에 따라 한번 더 조합되어야 비로소 한 글자가 구성되기 때문이다. 편의상 영어 및 기타 문자나 한글 자소를 'letter'로 정의하고 한글 문자는 'character'로 구분하여 정의한다.

그림 1의 (a)는 한글과 영어 및 기타 문자의 계층적 구조를 나타내고 있으며, 그림 (b)는 이에 따른 인식의 계층적 과정을 나타내고 있다.



(a) 문자의 계층적 구조

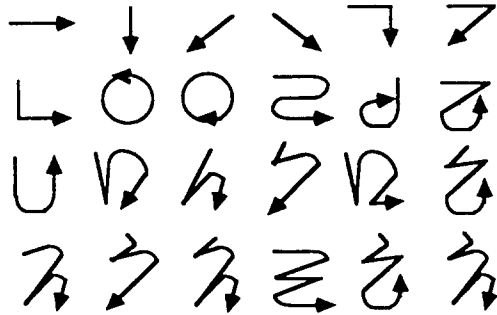


(b) 인식의 계층적 과정

그림 1. 문자의 계층적 구조 및 이의 인식 과정  
Fig. 1. Hierarchical structure and recognition process of characters.

1. 획의 종류

다음의 그림 2와 같이 한글 문자는 정자체 문자의 기본 획과 자소내에서 흔히 발생하는 흘림 획을 포함하여 총 24종류의 획으로 구성되고, 영어 대문자, 숫자, 그리고 키보드 부호등은 총 42종류의 획으로 구성된다.



(a) 한글 문자를 이루는 획의 종류



(b) 영어 및 기타 문자를 이루는 획의 종류의 예

그림 2. 획의 종류  
Fig. 2. Stroke types.

2. 문자의 종류

한글의 자소와 영어 및 기타 문자는 표 1과 같다.

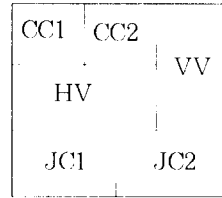
표 1. 한글 자소와 영어 대문자, 숫자, 키보드 부호의 종류

Table 1. Hangeul letters, English uppercase letters, digits, and keyboard symbols.

한글 자소	자음	ㄱ ㅋ ㆁ ㄷ ㅌ ㄴ ㄹ ㅁ ㅂ ㅃ ㅅ ㅆ ㅈ ㅊ ㅊ ㅌ ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ
영어 대문자	수평모음	h k t p h
숫자	수평모음	h k t p h
키보드 부호	수직모음	h k t p h

영어 대문자	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
숫자	1 2 3 4 5 6 7 8 9 0
키보드 부호	( ) + - * / = # % : ; ? ! , .

한글 문자는 그림 3과 같이 상하좌우의 배치로 몇 개의 자소가 문법 구조에 맞추어 조합되어야 한다.



여기서,

CC1 = {기본자음 14개}

CC1 · CC2 = {ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㄴ, ㄹ, ㅁ, ㅂ, ㅃ, ㅅ, ㅆ, ㅈ, ㅊ, ㅊ, ㅌ, ㅍ}

HV = {수평모음 5개}

VV = {수직모음 9개}

HV · VV = {과, 고, 개, 구, 귀, 괴, 기}

JC1 = {기본자음 14개}

JC1 · JC2 = {ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㄴ, ㄹ, ㅁ, ㅂ, ㅃ, ㅅ, ㅆ, ㅈ, ㅊ, ㅊ, ㅌ, ㅍ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ}

그림 3. 한글 문자의 구조

Fig. 3. The structure of Hangeul.

III. 인식을 위한 특징 및 문법 정보

필기 문자는 문자의 크기는 물론, 획 자체의 부분적인 크기나 획간의 거리등에 있어서 변화가 심하기 때문에 거리성 특징보다는 이러한 변형에 덜 민감한 방향성 특징을 주로 사용하는 것이 효과적이다.<sup>14</sup> 즉, 그림 4에서 알 수 있듯이, 필기 방향에 주목할 경우 '2'의 부분적 크기 차이나 '기'의 획간의 거리에 크게 영향을 받지않고 인식할 수 있다.



그림 4. 변형 패턴의 예

Fig. 4. Examples of variant patterns.

인식전에 행하는 전처리 알고리즘은 입력 장치로부터 들어온 획 단위의 데이터로부터 잡음이나 인식에 불필요한 중복점 등을 제거하고 인식에 필요한 시작, 끝, 굴곡점 등 주요 특징점만을 추출하는데 적합하도록 구현된다.<sup>15</sup>

1. 방향 코드와 멤버십 함수

인식에 주로 사용되는 방향성 특징을 위해 그림 5와 같이 0에서 7까지 8방향의 방향 코드를 정의하되, 일반적인 필기 형태에 적합하도록 수직, 수평보다는

사선 방향에 대해 보다 많은 각을 할당한다.<sup>16)</sup> 또한, 두 점이 너무 가까워서 방향이 무의미한 경우를 위하여 근접 코드 8을 정의하며, 방향 부호가 증가하는 방향 변화를 양의 방향 변화, 방향 부호가 감소하는 방향 변화를 음의 방향 변화로 정의한다.

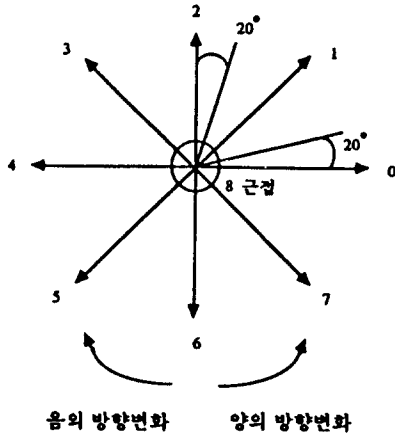


그림 5. 방향 코드와 방향 변화  
Fig. 5. Directional codes and directional changes.

각 코드에 대한 멤버십 함수(memberhip function)는 다음과 같다. 즉, 그림 6의 (a)는 수평 방향과 수직 방향의 코드인 경우이며, (b)는 사선 방향의 코드인 경우이다. 여기서 멤버십 값이 1.0인 각도의 범위가 다른 것은 그림 2에서 각 코드의 적용구간의 범위가 다른 것과 그 이유가 같다.

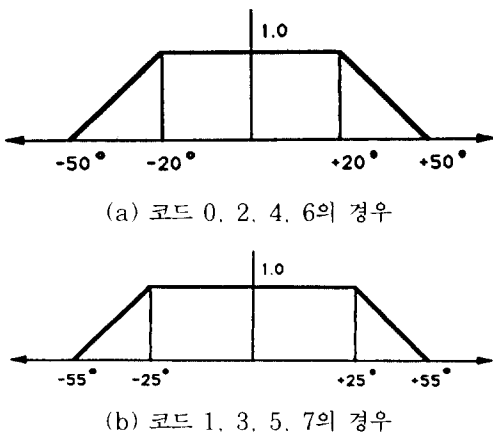


그림 6. 방향 코드의 멤버십 함수  
Fig. 6. Membership functions for directional codes.

또한, 코드 8에 대한 근접 정도를 나타내는 멤버십 함수는 그림 7과 같이 정의된다.

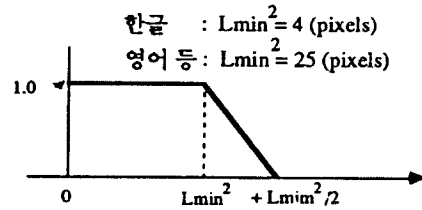


그림 7. 근접 코드 '8'의 멤버십 함수  
Fig. 7. Membership function for approximation code '8'.

2. Stroke 단계에 사용되는 특징  
획 인식에 사용되는 특징들은 다음과 같이 주로 필기 방향에 주목하여 정한다.

- 1) 획의 방향 변화 (CD)
  - 2) 획의 회전 정도 (CIRC)
  - 3) 획의 시작점에서 끝점을 잇는 방향 코드 (DSE)
  - 4) 획의 마지막 방향 코드 (DE)
  - 5) 획의 처음 방향 코드 (DS)
  - 6) y축으로 증감이 변하는 점간의 길이의 비
- 예를 들어, 그림 8의 (a)에서 DSE = 0 이고, (b)에서 DE = 5, DS = 0 이다. 또한, (c)는 양의 방향 변화였다가 다시 음의 방향 변화로 바뀐 경우(CD = +- )이며, (d)는 반 회전(CIRC = 1)을 인정하고, (e)는 완전 회전(CIRC = 2)으로 간주한다. 특히, (g)는 '0'과 '6'의 구별을 위해 사용한 거리의 비를 나타낸다.

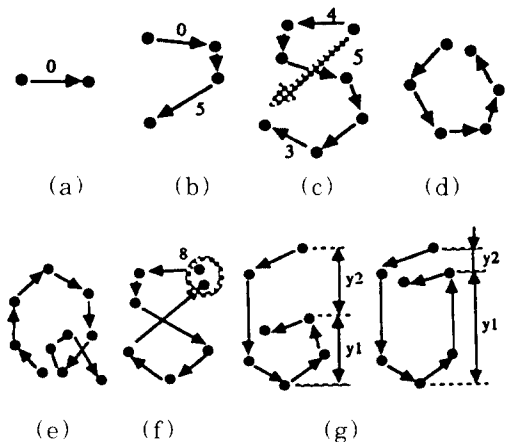


그림 8. 획 인식에 사용한 특징의 예  
Fig. 8. Examples of features for stroke recognition.

3. Letter 단계에 사용되는 특징

획간의 위치 관계도 거리에 무관하게 상하좌우 배치를 파악할 수 있는 방향성 특징을 사용하여 구하는 것이 바람직하다. 다음의 특징 1) 에서 5)는 한글 자소나 영어 및 기타 문자에서 획 사이의 위치 관계를 규정하기 위해 사용한다.

- 1) 이전 획의 시작점에서 현재 획의 시작점을 잇는 방향코드 (HHD)
  - 2) 이전 획의 시작점에서 현재 획의 끝점을 잇는 방향코드 (HTD)
  - 3) 이전 획의 끝점에서 현재 획의 시작점을 잇는 방향코드 (THD)
  - 4) 이전 획의 끝점에서 현재 획의 끝점을 잇는 방향코드 (TTD)
  - 5) letter를 구성하는 획들의 시작점이나 끝점들 사이의 x축 혹은 y축에 대한 길이의 비
- 예로써, 그림 9의 (a)에서 키보드 부호 '4'는 수평 획 다음에 수직 획이 그려진 경우인데 HHD=1 이고, THD=3 이다. 또한 (b)는 영문자 'J'의 경우로써 HTD=6 이고 TTD=5 이다. 특히, (c)는 'A'와 'H'를 구분하기 위해 적용한 거리 특징의 예이다.

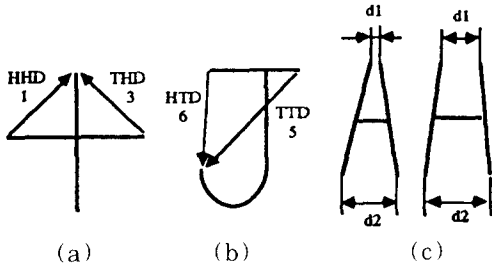


그림 9. Letter 인식에 사용되는 특징의 예  
Fig. 9. Examples of features for letter recognition.

4. Character 단계에 사용되는 문법 정보 및 특징 이 단계는 한글 자소들의 조합을 위해 그림 7의 위치 관계과 문법을 파악하는데 필요하며, 이에 따른 자소 조합의 흐름은 그림 10과 같다.

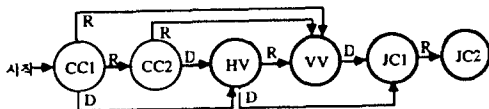


그림 10. 한글 자소 조합의 흐름도  
Fig. 10. Flow for the combination of Hangeul letters.

여기서 우측 R(right)과 하측 D(down)은 자소간의 위치 관계를 의미하는데, 이전 자소 박스의 중심점(CP)에서 현재 자소 박스의 좌상점(LUP)으로의 방향 코드가 0, 1, 2이면 우측 R이고, 방향 코드가 4, 5, 6 이면 하측 D이다. 그림 11에 자소 박스의 각 위치점이 나타나 있다.

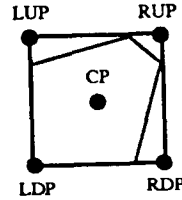


그림 11. 자소 박스의 위치점  
Fig. 11. Positions of a letter box.

IV. 퍼지 결정 트리를 이용한 계층적 인식

인식 대상의 계층적 특성에 맞추어 먼저 stroke의 가능한 후보 패턴을 구하고, 획간의 위치 관계를 파악하여 가능한 letter 후보들을 추출하며, 특히 한글의 경우는 연속된 자소들의 여러 가능한 조합 중에서 문법 구조에 따라 판단하여 최종 결과를 내게 된다. 다음에서 stroke 단계와 letter 단계에서의 퍼지 결정 트리에 대해 상세히 설명한다.

1. Stroke 인식을 위한 퍼지 결정 트리

획 인식을 위한 퍼지 결정 트리는 한글과 영어 및 기타 문자의 두 그룹에 대해 각각 따로 구성된다. 이때, 주어진 샘플 패턴들을 획 분류 알고리즘으로 분류하여 인식 트리의 기분을 구축한 다음, 데이터의 부족으로 생기는 일부 미비한 부분을 실험을 통하여 보완한다. 획 분류 알고리즘은 혼합된 획 클래스의 수가 적고 완전히 구분된 클래스의 수가 많은 특징을 선택하는데, 자세한 내용은 [17]을 참조하기 바란다.

그림 12는 이렇게 구성된 획 인식 트리의 형태이다. 노드 n은 적용하는 특징(feature)이고, 아크 a는 특징값(feature value)이며, 종단 노드 t는 획 코드(stroke code)이다. 이때 같은 획 코드가 하나 이상 존재할 수 있다.

획 인식을 위해 시작 노드에서 부터 한계치 이상의 가능한 모든 경로(path)를 추적하며, 모든 후보 획을 다음 단계에 넘겨준다.

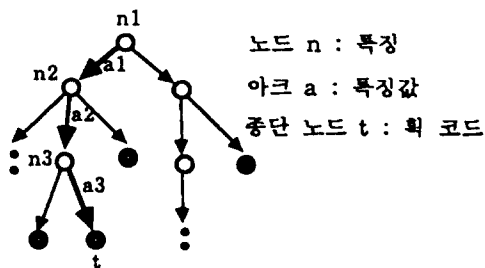


그림 12. 획 인식을 위한 퍼지 결정 트리의 형태  
Fig. 12. Fuzzy decision tree for stroke recognition.

따라서 임의의 획 코드에 대한 가능성은 동일한 코드를 갖는 종단 노드들의 OR(maximum)로 구하며, 각 종단 노드는 시작 노드부터의 경로(path)에 나타나는 아크의 AND(minimum)으로 구한다. 즉,

$$\mu(C_s) = \text{MAX}_i \{ \mu(t_{s_i}) \}$$

$$\mu(t_{s_i}) = \text{MIN}_j \{ \mu(a_j) \}$$

이때,

$C_s$  : 임의의 stroke S의 코드

$t_{s_i}$  : stroke 코드가  $C_s$ 인 종단 노드들,  $i = 1, \dots, p$

$a_j$  : 시작 노드에서 해당 종단 노드  $t_{s_i}$ 까지의 경로에서 각 단계  $j$ 의 아크,  $j = 1, \dots, q$

## 2. Letter 인식을 위한 퍼지 결정 트리

영어 및 기타 문자는 문자를 구성하는 획의 수를 미리 알 수 있기 때문에 획수에 따라 인식 트리를 따로 구성하는 것이 인식 성능 향상에 도움이 된다. 반면, 한글 문자는 자소를 이루는 획수를 미리 알 수 없으므로 획수에 따른 분류가 불가능하다. 또한, 자음, 수평모음, 수직모음에 따라 별개의 인식 트리를 구현할 수도 있으나, 인식을 위한 자소단계(CC1, CC2, HV, VV, JC1, JC2)를 미리 결정해야 하기 때문에 이를 결정하기 위한 부담과 잘못된 선택시 에러가 증가하는 위험 부담이 크다. 따라서 이러한 문제를 해결하고 인식의 흐름에 관한 정보를 총괄할 수 있도록 하기 위해, 자음, 수평모음, 수직모음을 포함하는 공통된 하나의 인식 트리를 구성한다.<sup>[10]</sup> 그림 13은 분류 알고리즘에 의해 구현된 letter 인식 트리의 형태이다. Letter 분류 알고리즘에서 하나의 letter만 존재하는 경우에는 특징값이 분산되지 않고 일관된 값을 나타내는 특징을 선택하고, 여러 letter가 혼합된 경우에는 letter간의 차이를 가장 크게 하

는 특징을 선택하는데, 자세한 내용은 [17] 을 참조하기 바란다.

노드 n에는 매칭되어야 하는 획 코드와 이전 획과의 위치 관계를 파악하기 위한 특징(feature)이 포함되어 있고, 아크 a는 특징값(feature value)에 해당한다. 획 인식 트리와의 중요한 차이는, 획 순에 따른 각 단계에서 획 코드가 매칭되는 노드를 먼저 선택한 후에 획 간의 특징을 적용하는 점이다.

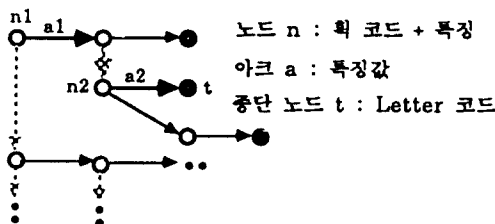


그림 13. Letter 인식을 위한 퍼지 결정 트리의 형태  
Fig. 13. Fuzzy decision tree for letter recognition.

임의의 letter 코드에 대한 가능성은 동일한 코드를 갖는 종단 노드들의 OR(maximum)로 구한다. 각 종단 노드의 가능성은 시작 노드부터의 경로(path)에서 각 단계별 가능성의 AND(minimum)로 구하며, 각 단계에서의 가능성은 매칭된 노드의 획 코드에 대한 가능성과 해당 아크의 최소값( $\wedge$ )으로 구한다. 즉,

$$\mu(C_L) = \text{MAX}_i \{ \mu(t_{L_i}) \}$$

$$\mu(t_{L_i}) = \text{MIN}_j \{ \mu(C_{s_j}) \wedge \mu(a_j) \}$$

이때,

$C_L$  : 임의의 letter L의 코드

$t_{L_i}$  : letter 코드가  $C_L$ 인 종단 노드들,  $i = 1, \dots, p$

$C_{s_j}$  :  $t_{L_i}$ 까지의 경로에서, 획 순에 따른 각 단계  $j$ 의 노드 n와 매칭된 획 코드,  $j = 1, \dots, q$

$a_j$  : 노드 n에서의 해당 아크,  $j = 1, \dots, q$

## V. 실험 및 고찰

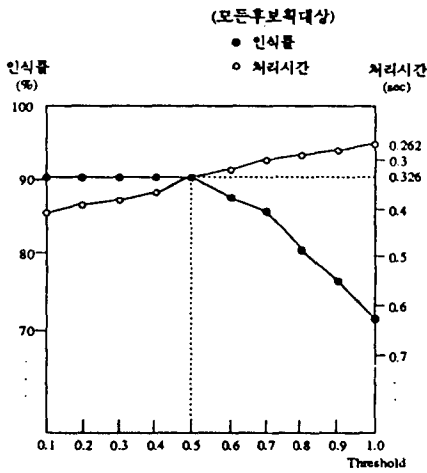
실험에 사용한 컴퓨터는 IBM 호환기종 PC 386-SX (21MHz) 이며, 알고리즘은 C 프로그래밍 언어를 사용하여 구현하였다. 입력장치는 Wacom사의 타블렛 디지털타이저를 사용하는데, 약 170 points/sec 정도의 sampling rate와 약 510 points/inch 정도의 해상도(resolution)를 갖는다.

입력 데이터는 타블렛(tablet)상의 펜끝(pen-tip)의 움직임에 따른 (x, y) 좌표값으로 입력되는데, 한글은 64 X 64 pixels의 박스를 사용하고 영어 및 기타 문자는 48 X 48 pixels의 박스를 사용하여 문자 단위의 입력을 한다. 필기 문자는 자연스러운 정자체로 일반적인 필순에 맞추어 써야 한다.

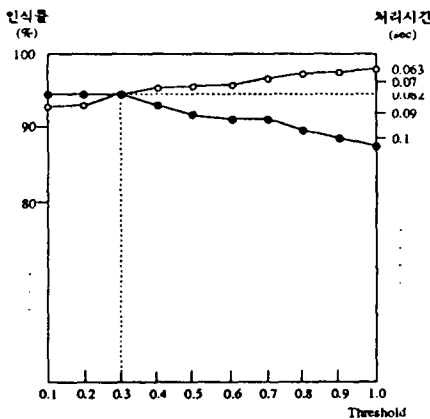
그림 14는 3사람으로 부터 다양한 조합을 고려한 한글 150자와 영어 및 기타 문자 153자의 데이터를 받아, 가능성의 한계치를 0에서 1까지 0.1씩 변화시켜 가면서 인식률과 처리 시간을 측정 한 것이다. 이때, 처리 시간은 전처리를 제외한 인식 모듈의 시간인데, 전

처리는 보통 자당 0.05 ~ 0.15초 정도 걸린다.

위 그림에서 알 수 있듯이, 한계치를 줄일수록 처리 속도는 다소 느려지지만 인식률은 증가하는데, 특히 한계치가 1.0이면 퍼지를 사용하지 않은 경우를 의미한다. 이때, 한계치를 너무 작게 설정하면 스택(stack)에 저장하는 노드의 수가 많아져 인식시간이 많이 걸리게 되고, 높게 잡을 경우 인식률이 떨어지기 때문에 한계치의 설정은 중요하다. 표 2는 효율적인 한계치(한글: 0.5, 영어 및 기타 문자: 0.3)에서의 제한 후보 획의 수에 따른 인식률과 처리 시간을 나타낸다. 여기서, 하위 단계의 후보 획이 상위 단계의 문자 인식률을 상승시키는데 크게 기여함을 알 수 있다.



(a) 한글의 경우



(b) 영어 및 기타 문자의 경우

그림 14. 한계치에 따른 인식률과 처리 시간의 변화  
Fig. 14. Results of recognition rate and processing time with respect to threshold.

표 2. 후보 획의 수에 따른 인식률과 처리 시간의 변화

Table 2. Results of recognition rate and processing time with respect to the number of stroke candidates.

사용 후보 획수	한글 (한계치 = 0.5)		영어 및 기타 (한계치 = 0.3)	
	인식률 (%)	인식시간 (sec)	인식률 (%)	인식시간 (sec)
1	72.7	0.272	91.5	0.068
2	89.3	0.321	93.5	0.078
3	89.3	0.325	94.8	0.081
all	90.7	0.326	94.8	0.082

결과적으로, 한글 문자는 한계치 0.5에서 인식률은 약 91%이고 인식 시간은 평균 0.33초이며, 영어 및 기타 문자는 한계치 0.3에서 인식률은 약 95%가 되며 인식시간은 평균 0.08초이다. 이는 퍼지를 적용하지 않은 경우(한계치 = 1.0, 후보획 수 = 1개)에 비해서, 한글은 18%, 영어 및 기타 문자는 8%의 인식률 상승을 나타낸다.

### VI. 결론

이 논문은 펜 컴퓨터의 핵심이 되는 온라인 필기 문자 인식에 관한 연구이며, 인식 대상은 한글과 영어 대문자, 숫자, 그리고 키보드 부호등이다.

필기와 같이 동일인이 쓰더라도 항상 같은 모양의 글자가 얻어지는 것이 아니라 정신적 또는 물리적 상황에 따라 데이터의 변동이 발생하는 경우, 이러한 애매한 정보를 흡수할 수 있는 방안이 필요하다. 따

라서 이 논문에서는 퍼지 이론을 응용하여 계층적인 구조를 갖는 문자 패턴에서 획의 형태와 획간의 위치 관계에 단계적으로 불확실성(uncertainty)을 흡수하고 가능한 조합을 놓치지 않음으로써 필기문자의 인식률을 높이는 방안을 제안하였다. 이 방법은 퍼지 이론을 적용하지 않은 경우에 비하여 인식률을 8~18% 정도 향상시켰다.

향후, 퍼지 결정 트리를 이용하여 문자를 인식하는 연구는 연속으로 필기되는 문장에서 내부 분할(internal segmentation)에 의해 가능한 후보 단어들까지 관리하는 문자 인식 방법에도 응용할 수 있다. 또한, 흘림체와 같이 보다 변형이 심한 필기 문자를 인식하기 위해서는 퍼지 결정 트리의 형성에 관한 조직적인 분류(classification)와 학습(learning)에 대한 연구도 필요하다.

#### 參 考 文 獻

- [1] 이의동, 김태균 외 2. "확장된 DP 매칭법에 의한 흘림체 한글 온라인 인식." 대한전자공학회 논문지, 제26권 제1호, pp.29-37, 1989. 1.
- [2] K. Yoshida, H. Sakoe. "Online Handwritten Character Recognition for a Personal Computer System." *IEEE Trans. Consumer Electronics*, vol.CE-28, no.3, pp.202-209, Aug. 1982.
- [3] 정복만, 권오석, 김태균. "온라인 입력 한글의 적응학습과 인식에 관한 연구." 한국정보과학회 논문지, 제16권 제5호, pp.487-498, 1989. 9.
- [4] P. S. Wang. "An Application of Array Grammars to Clustering Analysis for Syntactic Patterns." *Pattern Recognition*, vol.17, no.4, pp.441-451, 1984.
- [5] 최정훈, 권희용, 김춘석, 황희용. "신경망 모델을 이용한 한글 필기체 온라인 인식." 한국정보과학회 논문지, 제17권 제5호, pp.540-549, 1990. 9.
- [6] 이일병. "동적 문자인식 모형 사례연구." 한국정보과학회지, 제9권 제1호, pp.64-78, 1991. 2.
- [7] J. Y. Ha, J. H. Kim. "Extended Run-on English Recognition Using Hidden Markov Models." *2nd Int. Conf. on Automation, Robotics and Computer Vision*, pp.CV20.8.1-5, Sep. 1992.
- [8] 신봉기, 김진형. "통계적 방법에 의한 온라인 한글 필기 인식." 제 4 회 한글 및 한국어 정보처리 학술발표 논문집, pp.533-542, 1992. 10.
- [9] 김태균, 이은주. "한글에 적합한 획 해석에 의한 연속 필기 한글의 On-line 인식에 관한 연구." 한국정보과학회 논문지, 제15권 제3호, pp.171-181, 1988. 6.
- [10] 권오성, 권영빈. "온라인 한글의 획 해석방법에 대한 고찰." 한국정보과학회 인공지능연구회 인공지능소식, 제6권 제3호, pp.5-12, 1991.
- [11] 김재희. "문자인식에서의 퍼지이론응용." 한국정보과학회지 제10권 제1호, pp.38-47, 1992. 2.
- [12] 최용엽, 최갑석. "Fuzzy 추론을 이용한 온라인 필기체 한글문자 인식에 관한 연구." 대한전자공학회 논문지, 제27권 제11호, pp.103-110, 1990. 11.
- [13] C. C. Tappert, C. Y. Suen, and T. Wakahara. "The State of the Art in On-Line Handwriting Recognition." *IEEE Trans. Pattern Anal. Machine Intell.*, vol.12, no.8, pp.787-808, Aug. 1990.
- [14] J. R. Ward, T. Kuklinski. "A Model for Variability Effects in Handprinting with Implications for the Design of Handwriting Character Recognition Systems." *IEEE Trans. Systems, Man, Cybernetics*, vol.18, no.3, pp.438-451, May/June, 1988.
- [15] 구본석. "온라인 문자 인식을 위한 전처리 기법에 관한 연구." 연세대학교 전자공학과, 석사학위논문, 1991. 12.
- [16] M. R. Ito and T. L. Chui. "On-Line Computer Recognition of Proposed Standard ANSI(USASI) Handprinted Characters." *Pattern Recognition*, vol. 10, pp.341-349, 1978.
- [17] 전병환. "자동패턴분류와 퍼지집합이론에 의한 On-line 필기 문자 인식." 연세대학교 전자공학과, 석사학위논문, 1991. 6.
- [18] 전병환, 정진영, 김재희. "온라인 필기문자인식에서의 퍼지기법 응용에 관한 연구." 대한전자공학회 하계종합학술대회 논문집, 제15권 제1호, pp.735-738, 1992. 6.



## 著者紹介



全炳煥(準會員)

1989年 연세대학교 전자공학과 학사. 1991년 연세대학교 전자공학과 석사. 1992년~ 현재 연세대학교 전자공학과 박사과정. 주관심 분야는 인공지능, 패턴 인식, 문자 인식 등임.



金成勳(正會員)

1988년 서강대학교 전자공학과 학사. 1990년 연세대학교 전자공학과 석사. 1990년 ~ 현재 연세대학교 전자공학과 박사과정. 주관심 분야는 인공지능, 패턴 인식, 서명 검증 등임.



金在熹(正會員)

1979년 연세대학교 전자공학과 학사. 1982년 미국 Case Western Reserve University 전기공학과 석사. 1984년 미국 Case Western Reserve University 전자공학과 박사. 1984년~ 현재 연세대학교 전자공학과 교수. 주관심 분야는 전문가 시스템등의 인공지능과 문자 인식등의 패턴 인식 등임.