

# 문자인식에서의 퍼지이론응용

연세대학교 전자공학과 김 재 희\*

## 목 차

I. 서 론	III. 퍼지기법을 사용한 특징추출 및 매칭
II. 필기체 문자인식의 개요	IV. 결 론

### I. 서 론

1940년대에 최초로 컴퓨터가 등장한 이래, 컴퓨터의 하드웨어와 소프트웨어는 실로 엄청난 발전을 보여 왔다. 특히 메모리소자나 중앙처리장치 등은 초기의 컴퓨터에서는 거의 상상을 못하던 수준에 이르렀다. 그러나 불행히도 사람과의 인터페이스(man-machine interface) 측면에서는 단말기에 의존하는 수준에 오랫동안 머무르고 있는데, 이에 대한 대책으로 인간의 음성이나 필체를 직접 입력으로 받아들이는 연구가 널리 진행되고 있으며, 또 일부는 실용적인 수준에까지 도달하였다. 이 논문은 컴퓨터의 문자인식중 특별히 필기체문자 인식과 관련하여, 최근에 널리 이용되고 있는 퍼지이론을 응용한 필기체문자 인식기법을 고찰한 것이다.

일반적으로, 필기체문자를 인식하는 경우에는 다양한 원인에 의하여 불확실성이 존재하게 된다. 즉, 다른 사람에 의한 필적은 물론이고, 같은 사람에 의한 필적도 상황에 따라서 달라지게 된다. 또한, 인식 대상으로 입력된 문자가 컴퓨터에 저장되어 있는 어떠한 내용과도 완전히 일치하지 않을 수도 있고, 혹은 입력 자체가 잡음 등에 의하여 훼손될 수도 있다. 따라서, 들어온 입력

문자에 대하여 가장 근사하다고 판단되는 것(혹은 이와 더불어, 두번째 등의 후보들)을 인식결과로 제시하기 위해서는 불확실성을 처리하여야 한다. 이러한 불확실성을 처리하기 위해서는, 휴리스틱한 방법으로서 경험적 수치를 이용하는 직관적인 방법도 있으나[1], 보다 수학적인 근거를 둔 방법으로는 Baye's Rule에 의한 통계적인 방법[2]과 퍼지이론을 이용한 방법[3]이 있다. 일반적으로 패턴인식문제에 있어서는 통계적 방법이 중요한 역할을 담당하지만, 필기체문자인식에 있어서는 퍼지기법이 유용하다고 판단되는데 그러한 이유는 다음과 같다.

- ① 퍼지집합이론은 Bayesian 통계방법보다 연산이 쉽고 간단하여 처리속도가 빠르다.
- ② 통계방법에 있어서는 다양한 내용의 통제를 미리 구하여야 하는데 실제로 이것이 어려운 경우가 많다. 특히 우리가 사용하는 특징들(예컨대, 획의 모양, 획과 획간의 방향 등)은 모양이 너무 다양하여 본질적으로, 완벽한 정의가 어려운데 이러한 것들에 대한 확률적 분포를 미리 통계적으로 구한다는 것이 불가능하다. 이런 경우에도 퍼지이론은 비교적 손쉽게 사용될 수 있다.
- ③ 비교되는 획이나 문자들간의 부합되는 정도를 나타내고 이러한 정보들을 관리하기에 퍼지값이 적합하다.

\* 통신회원

필기체문자인식에서 인식대상문자가 입력되는 시간과 인식되는 시간이 같으면 온라인(on-line)인식으로 간주되고, 다르면 오프라인(off-line)인식으로 구분된다. 이 두 가지는 인식방법에 있어서 유사부분도 있으나, 입력방법을 포함한 인식방법 및 결과의 인식률 등 많은 부분에 있어서 큰 차이를 보인다. II장에서는 이러한 온라인 및 오프라인 인식에 관한 일반적인 방법에 대하여 서술하며, III장에서는 퍼지이론의 응용가능 부분을 최근의 문자인식 연구들을 사례로 하여 어떤 부분에 어떤 방법을 적용하였는가에 대하여 서술하였고, IV장에서는 결론을 맺는다.

## II. 필기체문자인식의 개요

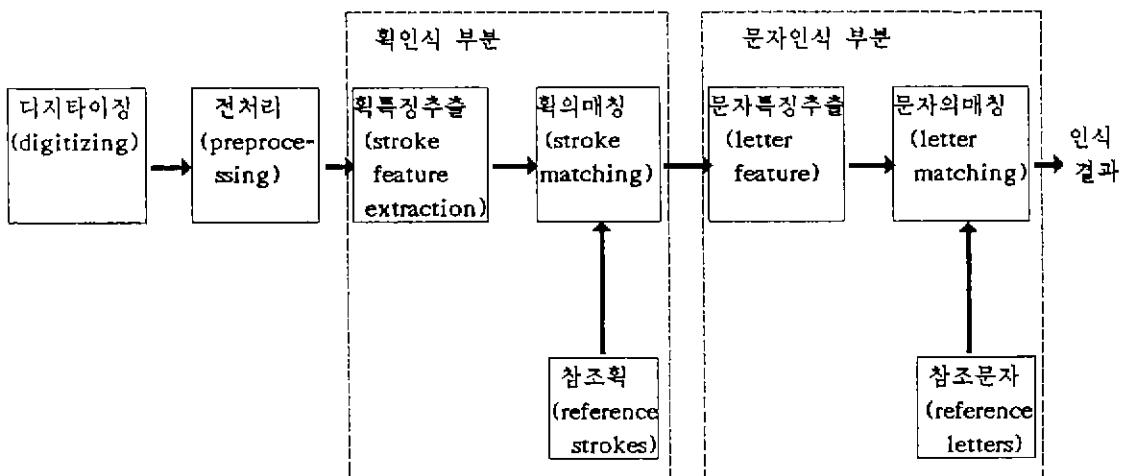
### 2.1 온라인 (on-line) 문자인식

온라인 문자인식은 카메라나 스캐너(scanner)와 같은 입력장치를 사용하는 오프라인 문자인식과는 달리 일상적으로 사용되는 필기구인 펜과 특성이 유사한 스타일러스(stylus)나 라이트펜(light pen) 등을 태블릿(tablet)이나 LCD 입·출력 장치에 필기토록 문자를 입력시키며, 문자가 쓰여지는 시간과 거의 동시에 입력 및 인식이 진행된다[4].

전형적인 온라인 문자인식을 단계별로 구분하면 (그림 1)과 같다.

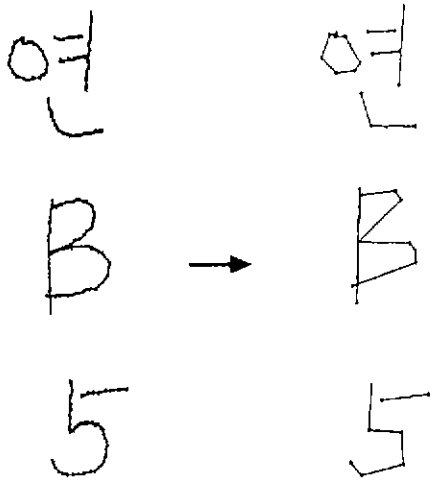
디지털라이징(digitizing)은 펜의 움직임에 나타내기 위하여 펜의 위치를 나타내는 일련의 x, y좌표로 얻게 되며, 이 데이터는 태블릿 등으로부터 실시간으로 입력된다. 하나의 문자는 하나 혹은 그 이상의 획(stroke)들로써 구성되는데, 하나의 획은 펜이 태블릿 면에 눌러진 때부터 떨어진 때까지의 펜의 이동경로로 정의된다. 따라서 디지털라이징 결과는 펜의 위치와 함께, 펜이 태블릿에 눌러지는지 여부도 포함되므로, 획 단위의 좌표값들로 표현된다.

전처리(preprocessing)과정은 디지털라이징 결과로 얻어진 데이터의 잡음을 제거하고, 인식에 유용한 부분만을 걸러내는 작업이다. 이 과정은 일반적으로는 연속된 문자들의 정보를 각 문자별로 구분하는 외부분리(external segmentation), 입력장치의 구조적결함이나 필기자의 손떨림 등에서 발생하는 잡음을 제거하거나(noise reduction), 스무딩(smoothing)하는 부분, 중복되거나 불필요한 자료를 제거하는 필터링(filtering), 입력문자의 크기 및 방향을 학습저장된 참조문자와 일치시키기 위한 정규화(normalization), 필기 시작이나 끝 부분에서 생기는 고리모양을 제거하기 위한 디후킹(dehooking) 등이 포함된다[5, 6]. 경우에 따라서, 원표(.)나 마침표(.)를 찾아내는 작업이 이 과정에서 행하여지기도 한다[7]. (그림 2)는 디지털라이징에 의하여 얻어진 3가지 문자에 대하여 전처리 과정을 행한 이후의 결과를 보이고 있다. 여기서 좌표점들간의 연결선은 보기에 좋도록 하기 위



(그림 1) 온라인 문자인식의 전형적인 단계

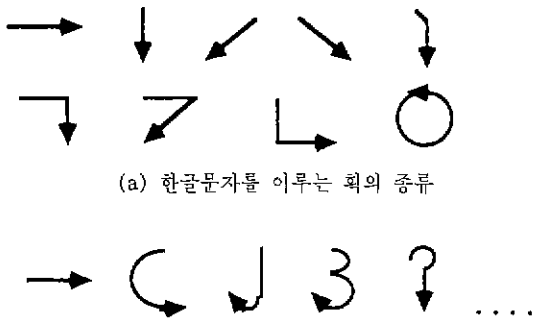
하여 추가적으로 부과된 것에 불과하고, 실제 유지되는 데이터들은 좌표점들뿐이다.



(그림 2) 전처리된 결과의 예

앞에서 언급한 바와 같이, 온라인 문자인식에서는 디지털아정 이후 얻어지는 문자에 대한 정보가 획 단위로 확실히 구분되어지는데, 이는 문자를 인식하는데 사용될 수 있는 대단히 유용한 정보가 된다. 따라서 문자인식은 입력되는 획에 대한 인식을 우선적으로 시도하고, 인식된 획들간의 조합으로써 문자를 인식하는 것이 보통이다. 혹은, 이와 달리 문자를 획 단위로 구분하지 않고 전체를 한번에 인식하는 시스템도 있다[8].

인식대상이 되는 획의 종류는 인식대상문자에 따라서 크게 달라지고, 부분적으로는 인식시스템에 따라서도 다소 다르다. (그림 3)에 몇가지 획에 대하여 나타내었다.



(a) 한글문자를 이루는 획의 종류

(b) 영문자, 숫자, 특수문자를 이루는 획의 종류

(그림 3) 획의 종류

일반적으로 한글을 이루는 획의 종류는 8개 내외가 보통이며, 영문자의 경우는 획의 종류가 20개 이상이다. 그러나 문자를 구성하는 획의 조합은 한글의 경우가 훨씬 다양하고, 영문자나 숫자의 경우는 매우 단순하다. 따라서 후자의 경우, 획에 대한 인식률이 전체 문자의 인식률에서 차지하는 비중이 대단히 크다. 반면, 한글문자의 경우, 획들의 조합으로 자소가 인식되며, 또 이들의 조합으로 하나의 문자가 인식되므로, 획의 인식도 중요하다, 획들간의 결합 및 자소간의 결합에 대한 인식방법도 매우 중요하다.

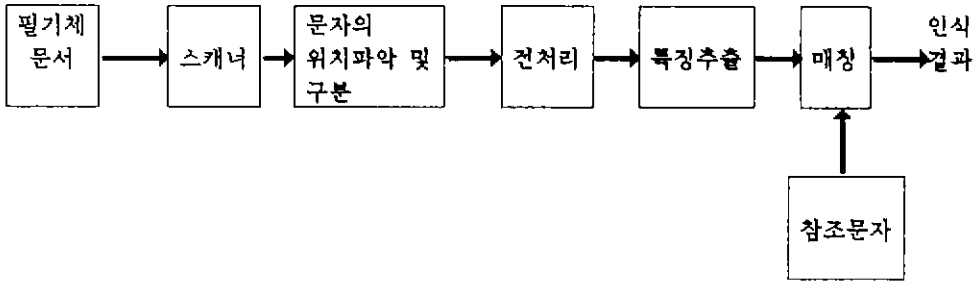
인식부분은 특징추출과정에서 인식에 사용되는 특징들을 얻어내어, 미리 학습과정에서 얻어진 참조패턴(reference pattern)들과 비교하여, 가장 적합한 패턴을 찾아내게 된다. 획 인식부분에서는 패턴이 획이 되고, 문자 인식부분에서는 패턴이 문자가 된다. 사용되는 특징 및 매칭과정은 인식시스템마다 독특한 방법을 쓰게 되는데, 이 논문에서는 특별히 퍼지기법을 응용한 방법들에 대하여 다음 장에서 면밀히 서술할 것이다.

## 2.2 오프라인(off-line) 문자인식

오프라인 문자인식은 여러 측면에서 온라인 인식과 차이가 나는데, 우선 데이터를 얻는 과정이 화상스캐너(image scanner)나 CCD 카메라를 통하는 것이 보통이며, 또 문자별로 데이터가 구분되어 얻어지기 보다는 문서영상으로부터 문자들을 구분하여 인식하게 되는 것이 보통이다. 또한 문서가 작성된 시간과 다른 시간에 인식이 수행된다. 일반적인 오프라인 문자인식과정을 단계별로 구분하여 보면 (그림 4)와 같다. 단계별 처리과정은 (그림 1)의 온라인의 경우와 유사하나(실제로 거의 모든 패턴인식과정이 이와 유사하다) 각 단계별 내용은 다소 다르다. 즉, 스캐너를 통하여 입력되는 정보는 여러 문자들이 혼합되어 있는 것이며, 여기에는 많은 잡음이 섞여 있는 것이 보통이다.

입력된 데이터에는 여러 문자들이 섞여 있으므로 이로부터 각 문자에 해당하는 영역을 파악하고 구분하게 된다. 각 문자정보는 온라인의 경우와는 달리 획에 대한 구분이나 좌표점들의 쓰여진 순서 등에 관한 것을 갖고 있지는 못하다.

전처리과정은 문자를 구성하는 선분에 나타난 구멍이나 끊어진 곳을 채우는 스무딩, 잡음의 제거, 크기의 정규화과정 등을 포함하게 되며, 경우에 따라서는 세션화



(그림 4) 오프라인 문자인식의 전형적인 단계

(thing)과정을 포함하기도 한다. 스무딩을 포함한 이런 과정들은 온라인의 경우와는 달리 보통 2차원 메트릭 영역으로 구성된 윈도우를 이용하는 것이 보통이다.

특징추출 및 매칭의 역할은 온라인의 경우와 같으나, 그 내용 및 방법은 온라인의 경우와 상이하며, 인식시스템에 따라 다르다. 온라인의 경우와 크게 다른 이유는 얻어진 데이터가 획 단위로 구분되어 있지 않고, 선분을 구성하는 데이터가 단순한 연속 좌표점이 아니며, 이들 간의 시간적 발생순위에 따른 정보도 제공되지 않는 데에서 기인한다. 이제, 우리는 III장에서 퍼지기법을 응용한 온라인 및 오프라인 문자인식에서 사용되는 특징과 매칭방법에 대하여 살펴볼 것이다.

### III. 퍼지기법을 사용한 특징추출 및 매칭

이 장에서는 특징추출 및 인식에서 퍼지기법을 사용한 여러 연구에 대하여 살펴보겠다. 이 중 3가지 연구는 온라인 인식에 해당되고 나머지는 오프라인 인식에 해당된다. 여기에서 언급된 연구가 결코 대표적인 연구는 아니며, 반드시 좋은 방법은 아닐 수도 있다.

#### 3.1 온라인 문자인식 1 : 최용엽[8]의 방법

이 연구는 한글인식을 대상으로 한 연구인데, 한글을 초·중·종성을 구성하는 자소로 구분하지 아니하고 전체 3,390자의 한글문자를 획 수에 따라 사전에 구분하여 놓은 뒤 입력되는 문자의 획 수를 구하여 같은 수의 획을 갖는 문자들 중에서 인식을 행하게 된다.

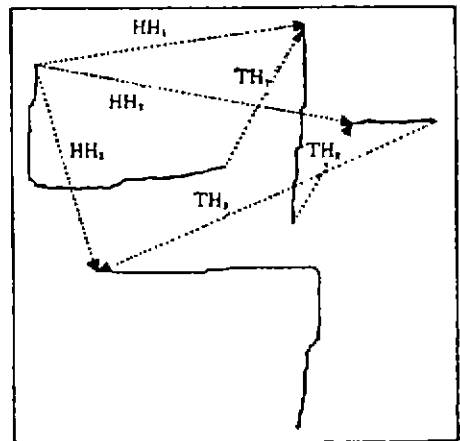
인식대상이 되는 기본 획은 0, ㄱ, ㄴ, /, |, \, -의 7가지이며, 입력된 획에 대하여 이의 길이를 S라 하고, 획의 시작점과 끝점을 잇는 직선의 길이를 L, 시작점과 끝점을 잇는 직선상에서 획까지의 최장의 법선에 대한

길이 h간의 관계를 파악하여 일차분류를 한 뒤에, 시작점과 끝점간의 방향을 파악하여 이차분류를 행하였다.

이 연구에서는 획의 인식에 대해 퍼지이론을 이용하지 않았는데, 이는 한글의 경우 획의 종류가 비교적 적고 구분이 명확하여 인식률이 높기(보통 98% 이상 가능) 때문이라고 판단된다. 그러나 다음에 논의될 김재희 등[9]의 연구에서 보듯이 획 인식에서도 퍼지기법의 응용은 상당히 유용하다.

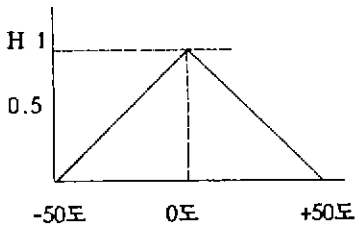
획을 인식한 뒤, 획들간의 상대적 위치정보로부터 문자를 인식한다. 이 연구에서는 상대적 위치정보로서, 한 문자를 이루는 N개의 획에 대하여 첫번째 획의 시작점과 (i+1)번째 획의 시작점을 연결하는 각도  $HH_i(i=1, 2, \dots, N-1)$ 와 i번째 획의 끝점과 (i+1)번째 획의 시작점을 연결한 각도  $TH_i(i=1, 2, \dots, N-1)$ 를 (그림 5)와 같이 이용한다.

방향코우드는 전체 180도를 18등분으로 나누어 코우



(그림 5) 상대위치에 따른 특징점

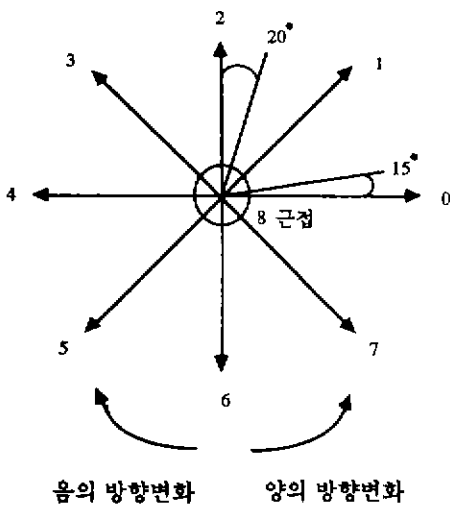
드 0부터 17까지를 각도 90도부터 -90도까지로 구분하였다. (이때, 예컨대 240도는 절대칭 값인 60도로써 간주한 듯하다.) 이때 각 코우드 값을 밀변이 100도가 되는 Yager 함수로써 나타내었다. 예컨대 코우드 8에 해당하는 방향코우드는 (그림 6)과 같은 멤버쉽 함수를 보인다. 학습에 의하여 생성되어 있는 참조문자는, 예컨대, "낙"의 경우,  $((HH_1=8) \wedge (HH_2=11) \wedge (HH_3=17)) \vee ((TH_1=4) \wedge (TH_2=1) \wedge (TH_3=7))$ 이다. 들어온 입력문자의 HH<sub>i</sub> 값과 TH<sub>i</sub> 값(i=1, 2, 3)을 구하여, 각 해당코우드의 멤버쉽 값을 구하고, "∧"는 최소값을 "∨"는 최대값을 구하면 입력문자가 각 참조문자와 어느 정도 유사한지가 결과의 멤버쉽 값으로 나오고, 인식결과를 가장 큰 멤버쉽 값을 지닌 문자로 처리한다.



(그림 6) 코우드 '8'의 멤버쉽 함수

3.2 온라인 문자인식 2 : 김재희[9]의 방법

이 연구에서는 한글문자에 대한 인식과 영어, 숫자 및 부호(?, !, ', ,, 등)에 대한 인식(이하 영어류로 칭함)의

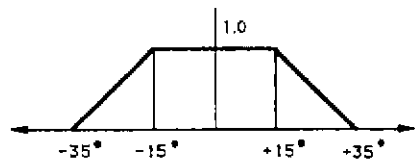


(그림 7) 방향 코우드

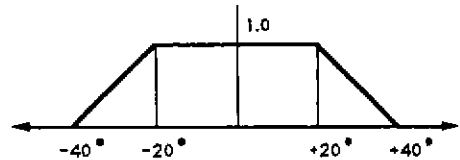
두 부류에 대한 것이 인식의 대상이다.

이 연구에서는 우선 비교적 철저한 전처리 과정을 통하여 각 획을 구성하는 필요 좌표점들만을 얻는다(그림 2 참조). 획의 인식을 위하여 사용된 특징들은 획의 시작 두점간의 방향, 끝 두점간의 방향, 시작과 끝점간의 방향 등의 방향코우드가 쓰인다.

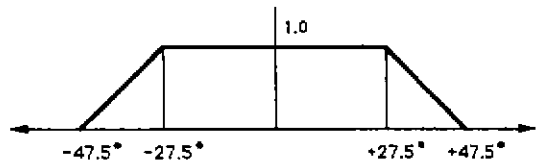
이 연구에서 사용된 (그림 7)의 방향코우드는 360도를 8등분한 것인데, 각 코우드에서 정의된 멤버쉽 함수의 값은 (그림 8)과 같다.



(a) 코드 0.4의 경우



(b) 코드 2.6의 경우



(c) 코드 1, 3, 5, 7의 경우

(그림 8) 방향코우드의 멤버쉽 함수

수평방향과 수직방향, 또 대각선방향의 멤버쉽 함수가 서로 다른 것은 일반적인 필기의 관습에 따라 대각선의 변화가 더욱 심한 것을 반영하여 준 것이다. 저장된 참조획은 한글의 경우 9개, 영어류의 경우 38개(후에 참조획의 수는 더욱 늘어남)로써, 참조획은 각 특성에 따른 코우드 값을 지니고 있으며, 입력문자에 대하여 각 특징에 대한 유사정도를 멤버쉽 함수로 주게 된다. 이 연구에서는 이 밖에도 획의 회전여부, 획의 방향변화, 획 전체의 x축 길이에 대한 시작점과 끝점간의 x축 길이의 비, 혹은 이러한 y축 길이의 비 등에 대한 특징도 멤버쉽 함수에 의한 퍼지집합으로 나타낸다. 앞의 최용엽[8]의 연

구와는 달리, 이러한 특징들은 노드로 하여 분류트리(획 인식용)를 구성하여 획의 인식을 시도하였고, 결과도 가장 멤버쉽 값이 큰 획이 제시되나, 후의 문자인식 과정을 위하여 몇개의 차선들도 유지한다.

문자의 인식을 위해서는 앞 획의 시작점과 뒷 획의 시작점과의 방향(HHD), 앞 획의 시작점과 뒷 획의 끝점과의 방향(HTD), 앞 획의 끝점과 뒷 획의 시작점과의 방향(THD), 앞 획의 끝점과 뒷 획의 끝점과의 방향(TTD) 및 획의 중심간의 방향 등이 쓰인다. 이 과정에서도 각 방향에 해당하는 퍼지집합을 정의하여 문자인식을 위한 분류트리(문자인식용)의 각 노드에서 입력문자에 대하여 지정된 특징을 구하여 저장된 참조특징들과 비교하여 가장 큰 멤버쉽 값을 따라간다. 이때, 경우에 따라서는 인정할 만한 멤버쉽을 지닌 경로를 찾지 못하는 수도 있는데, 이는 현재 인식한 획이 잘못되었을 때 종종 생겨나게 된다. 이런 경우에는 획인식과정에서 차선으로 주어진 획으로 재인식을 시도한다.

실과에 의하면, 획인식에서의 퍼지응용은 영어류의 경우 획인식률 자체는 95.3%에서 95.6%의 미소한 향상을 가져오나, 후보획의 제시에 의한 문자인식률의 향상은(물론 문자인식에서 사용한 또 다른 퍼지기법의 응용과 합쳐서) 81.5%에서 92.4%를 가져왔다.

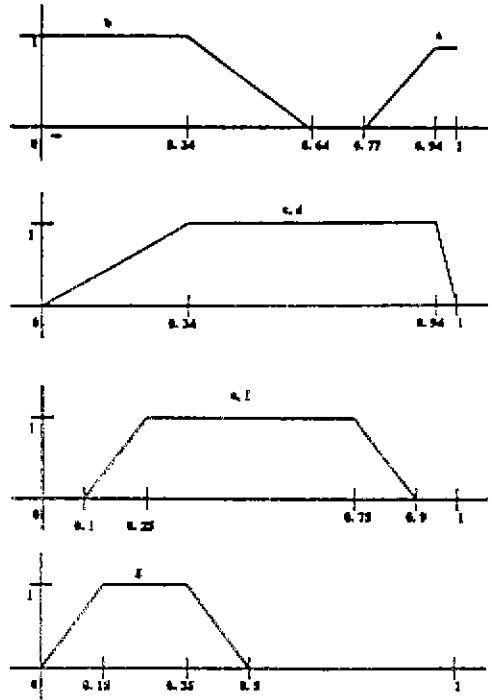
3.3 온라인 문자인식 3 : 오경환[10]의 방법

이 연구에는 획인식을 위하여 X좌표 길이의 비, Y좌표 길이의 비, 시작점과 끝점 사이의 길이의 비를 사용한 점이 다르다. 예컨대 X좌표 길이의 비는 한 획의 최소 X좌표와 최대 Y좌표 사이의 길이를 획 전체의 길이로 나눈 값을 가지고(그림 9)와 같은 퍼지집합중 최대의 것을 선택한다. 이 밖에도 시작점과 끝점 사이의 길이의 비는 한 획의 시작점과 끝점 사이의 길이를 획 전체로 나눈 값이며 이를 위한 퍼지함수도 따로 정의하였다. 이 밖의 사항은 김재희[9]와 유사하며, 인식대상은 한글문자이다.

3.4 오프라인 문자인식 1 : SIY[11]의 방법

이 연구는 필기체 숫자인식을 위한 것으로 오프라인 인식을 행한 연구이지만, 사용된 특징 및 인식방법은 온라인 인식에서도 적용가능하다.

이 연구에서는 각 숫자를 노드(node)와 가지(branch)로써 나타낸다. 노드는 끝점(이웃점이 하나), 코너(이웃



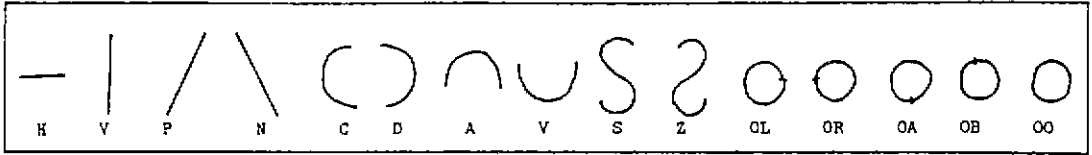
(그림 9) X좌표 길이의 비에 대한 퍼지소속함수

점이 두개이나 급격한 변화를 이룸). 할류점(셋 이상의 이웃점을 갖는 점) 중의 하나로 되고, 가지는 노드를 연결하는 선분으로서, (그림 10)과 같이 정의된다. 따라서(그림 11)에서 보이는 "5"의 경우,

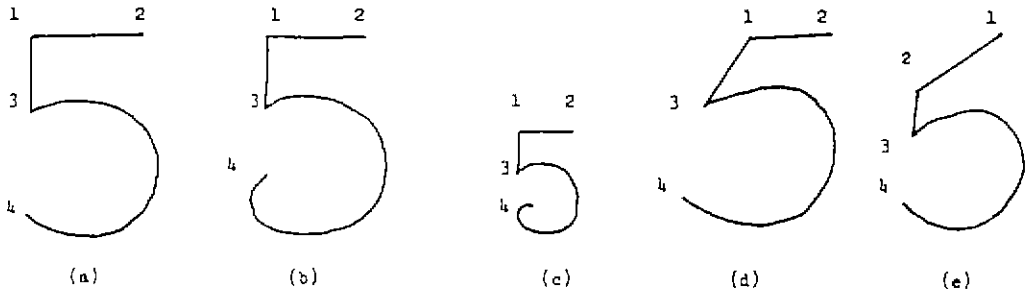
$$\begin{aligned}
 F(5) &= F(5a) + F(5b) + F(5c) + F(5d) + F(5e) \\
 &= H(1, 2) \cdot V(1, 3) \cdot D(3, 4) \\
 &\quad + H(1, 2) \cdot P(1, 3) \cdot D(3, 4) \\
 &\quad + P(1, 2) \cdot V(1, 3) \cdot D(3, 4)
 \end{aligned}$$

와 같이 주어진다. 이러한 표시가 모든 숫자에 대하여 얻어져서 참조패턴으로 사용된다. 후에 퍼지연산시에는 +는 최대값을, ·은 최소값을 구하게 된다.

인식을 행하기 위해서는 입력된 숫자에 대한 노드 및 이들간의 가지특징을 구하여야 하는데, 가지특징의 유사 여부를 판단하기 위하여 참조패턴을 퍼지집합으로 모사한다. 즉, 우선적으로 얻어진 선분에 대하여 직선어부(H, V, P, N)를 판단하기 위하여 최소제곱오차 S<sub>r</sub>를 갖는 curve-fitting을 행한 뒤, 전체선분의 길이 S와 비교를 한다.



(그림 10) 가지특징 집합



(그림 11) 문자 '5'의 전형적인 예

$$f_{SL} = 1 - S_T / S \quad (\text{if } S > S_T)$$

$$= 0 \quad (\text{if } S \leq S_T)$$

을 구하여  $f_{SL}$ 이 0.5 이상이면 직선으로 판정하고, 0.5 이하이면 다른 것으로 판정한다(여기에서도 퍼지집합을 이용할 수 있을 것이다). 입력된 선분이 직선이라고 판정되면, 이것이 H, V, P, N 중의 어느 것에 속하는가를 판단하기 위하여 퍼지집합을 힘수로서 정의한다. 여기에서  $\theta$ 는 curve-fitting에서 얻어진 선분의 기울기이다.

$$FH(\theta) = 1 - \min[\min[|\theta|, |180 - \theta|, |360 - \theta|], 45, 1]$$

$$FV(\theta) = 1 - \min[\min[|90 - \theta|, |270 - \theta|], 45, 1]$$

$$FP(\theta) = 1 - \min[\min[|45 - \theta|, |270 - \theta|], 45, 1]$$

$$FN(\theta) = 1 - \min[\min[|135 - \theta|, |315 - \theta|], 45, 1]$$

만일 선분이 직선이 아닌 반원이라면 일단 수직형태(FVC: C와 D 포함)인지 수평형태(FHC: A와 V 포함)인지를 판단하기 위하여 가지의 시작점과 끝점으로부터  $\theta = \tan^{-1}(Y_n - Y_1) / (X_n - X_1)$ 을 구한다. 이를 이용하여,

$$FH(\theta) = 1 - \min[\min[|\theta|, |180 - \theta|, |360 - \theta|], 90, 1]$$

$$FV(\theta) = 1 - \min[\min[|90 - \theta|, |270 - \theta|], 90, 1]$$

를 구한다. 이런 식으로 세부적으로 구하여 가면, 최종적으로 각 가지특징에 대한 유사성 여부를 퍼지 멤버십 값으로써 얻을 수 있게 된다.

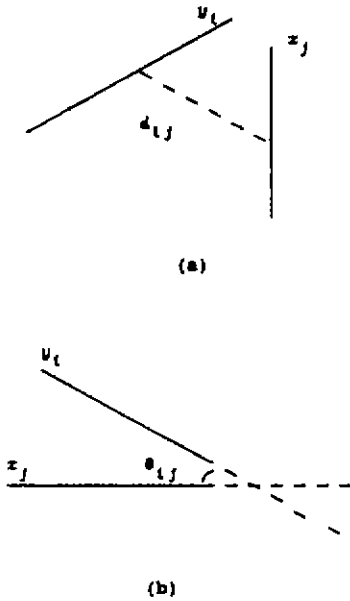
이 방법은 앞에서 서술한 연구들에서 사용한 지역적 특징(local feature)을 배제하고 전역적 특징(global feature)을 사용함으로써 변형이나 잡음에 강한 인식 특성을 지니게 되나, 처리 속도가 늦게 되는 단점이 있다.

### 3.5 오프라인 문자인식 2 : CHENG[12]의 방법

이 연구는 필기체 한자인식을 대상으로 오프라인 인식을 위한 연구이지만 사용된 방법은 온라인 방식에서도 공히 적용될 수 있다. 한자의 경우 한글이나 영문자류와는 달리 문자를 구성하는 획이 모두 직선이거나 이의 단순결합이며 원이든 원의 일부와 같은 커브모양은 없다.

이 연구에서는 두 문자들간의 유사성을 위치 유사성(location similarity) 및 타입 유사성(type similarity)으로 구분하여 고려한다. 위치 유사성은 각각 서로 다른 문자의 두 획  $y_1$ 의  $z_2$ 에 대하여 크기를 정규화한 뒤, (그림 12(a))와 같이  $d_n = \|M_i - M_j\|$ 로써 각 문자에서의 획  $y_n$ ,  $z_n$ 의 상대적 위치를 비교하고, 이에 의하여 멤버십 값  $\mu_{10} = \exp(-K_1 \cdot d_n)$  ( $K_1$ : 적당한 상수)를 얻는다.

한편, 타입의 유사성은 비교되는 두 획간의 각도  $\theta_n$ 를 (그림 12(b))와 같이 구한 뒤에  $\mu_{10} = \exp(-K_2 \cdot d_n)$ 에 의하여 구한다. 두 획간의 유사성에 대한 종합 멤버십은  $\mu_{10} = \min[\mu_{10}, \mu_{10}]$ 로써 구한다.

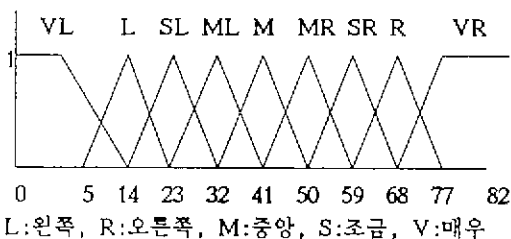


(그림 12) (a) 거리  $d_{ij}$  및 (b) 각  $\theta_{ij}$ 의 정의

이 방법은 한자와 같이 획이 단순하고, 획 수에 의한 문자 분류가 용이하며, 또 크기와 방향에 대한 정규화가 정확하게 이루어지는 경우에 효과적이다. 이 방법은 연산이 비교적 단순하며 처리속도가 빠른 것이 장점이다.

3.6 오프라인 문자인식 4 : 송정영[13]의 방법

이는 필기체 한글문자에 대한 오프라인 인식의 연구로서, 입력문자를 세로 160, 가로 82좌표의 정보로 압축한 화상에 대하여 인식을 행한다. 사용된 특징은, 문자를 구성하는 세그먼트의 수(세그먼트는 획보다 작은 단위로서, 이 연구의 경우에는 /, \, |, -, o의 5가지가 있음)와 각 세그먼트의 시점좌표( $X_0, Y_0$ ), 종점좌표( $X_1, Y_1$ ), 시점과 종점간의 직선길이(L), 이의 각도( $\theta$ ) 등이



(그림 13) 멤버십 함수( $X_0$ )

다. 여기에서 세그먼트의 수를 제외하고는 모든 것이 퍼지집합으로 정의되어 있다. 예컨대, 시점좌표  $X_0$ 에 대한 멤버쉽 함수는 (그림 13)과 같다.

인식방법은 전 한글문자를 세그먼트 수에 따라 나누어 각 문자에 대하여 문자를 구성하는 각 세그먼트에 대하여 퍼지규칙을 (그림 14)와 같이 구성하여 참조문자로 저장한다. 입력된 문자에 대하여 세그먼트를 구한 뒤, 그 수에 맞추어 해당되는 문자들에 대한 퍼지값을 구하여 제일 큰 것으로 간주한다.

<p>제 1 세그먼트에 대하여</p> <p>시점좌표 <math>X_0</math>가 중앙에 위치하여 있고,                  시점좌표 <math>Y_0</math>가 위에 위치하여 있고,                  길이 L이 매우 짧으며,                  각도 <math>\theta</math>가 360도이다.</p> <p>제 2 세그먼트에 대하여</p> <p>시점좌표 <math>X_0</math>가 왼쪽에 위치하여 있고,                  시점좌표 <math>Y_0</math>가 중앙에 위치하여 있고,                  길이 L이 길며,                  각도 <math>\theta</math>가 180도이다.</p> <p>제 3 세그먼트에 대하여</p> <p>시점좌표 <math>X_0</math>가 중앙에 위치하여 있고,                  시점좌표 <math>Y_0</math>가 중앙보다 위에 위치하여 있고,                  길이 L이 짧으며,                  각도 <math>\theta</math>가 92도이다.</p>
---

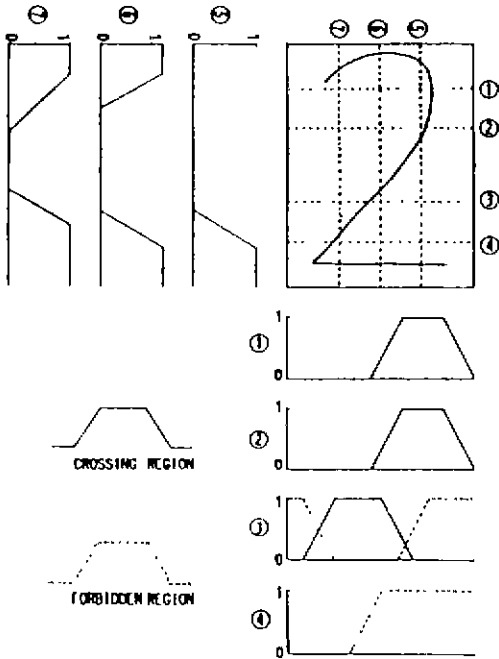
(그림 14) 입력문자 '우'에 대한 퍼지화

이 방법은 입력문자의 크기 및 방향에 대한 정확한 정규화가 필요하고, 속도가 느리며, 조합형 문자에는 적합한 방식이 아닌 것으로 판단된다.

3.7 오프라인 문자인식 4 : Yamakawa[14]의 방법

이 방법은 필기체 숫자인식을 위한 것으로서, 각 숫자에 대하여 (그림 15)와 같이 그리드를 ①~⑦까지 구성하여, 각 그리드에서 교차점이 꼭 나타나야 할 곳과 꼭 나타나지 말아야 할 곳에 대한 퍼지집합을 구성하여, 입력문자와의 유사성을 얻는다. 이 방법은 사용되는 특징이 매우 단순하고, 쓰여진 획의 위치가 다소 변하여도 인식률의 현저한 차이를 가져오며, 복잡한 패턴의 경우 처리 방법이 복잡해지는 단점이 있다.





(그림 15) 숫자 '2'에 대한 crossing 영역과 forbidden 영역의 멤버십 함수의 할당

#### IV. 결 론

이상에 걸쳐 필기체 문자인식에서 퍼지기법을 사용한 연구에 대하여 오프라인과 온라인으로 구분하여 살펴 보았다. 오프라인에 비하여 온라인은 필기방향이나 획간의 구분에 대한 정보가 훨씬 정확하기 때문에, 비교적 인식률이 높으며 실용성이 클 것으로 판단된다. 여기에서 검토한 연구들에 대하여 인식률이 대체로 알려져 있으나, 인식자의 협조여부 및 참조문자와의 유사관계 등에 의하여 의존되는 바가 크므로, 인식률의 비교는 행하지 않았다. 살펴본 바와 같이, 필기체 문자처럼 획 형태나 위치관계의 변동이 상황에 따라 변화가 큰 경우에는 이 변동량을 흡수하는 퍼지기법이 유용하게 사용될 수 있다. 또한, 획이나 문자의 부합되는 정도에 따라 차선택을 관리함으로써 인식률의 향상과 후보문자의 제시를 가능하게 한다.

#### 참 고 문 헌

1. 김재희, 인공지능의 기법과 응용, 교학사, 1988.

2. I. Guttman, S. S. Wilks, and J. S. Hunter, *Introductory Engineering Statistic*, John Wiley & Sons, Inc., 1971.

3. G. J. Klir and T. A. Floger, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, 1988.

4. J. R. Ward and M. J. Phillips, "Digitizer Technology: Performance Characteristics and the Effects on the User Interface," *IEEE CG & A*, pp. 31~44, April 1987.

5. 구본석, 김성훈, 김재희, "On-line 문자인식에서의 Preprocessing 알고리즘", 한국통신학회 추계종합 학술발표회 논문집, pp. 285~288, 1990. 11.

6. E. Mandler, "Advanced Preprocessing Technique for On-Line Recognition of Hand-printed Symbols," *Computer Recognition and Human Production of Handwriting*, pp. 19~36, 1989.

7. 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. 2, No. 8, pp. 787~808. Aug. 1990.

8. 최용영, 최갑석, "Fuzzy 추론을 이용한 온라인 필기체 한글문자 인식에 관한 연구", 대한전자공학회 논문지, 제27권 제11호, pp. 103~110, 1990. 11.

9. 전영환, 구본석, 김성훈, 김재희, "퍼지이론을 응용한 펜 컴퓨터에서의 On-Line 획인식 기법", 한국퍼지시스템연구회 춘계학술대회 학술발표논문집, 제1권 제1호, pp. 168~176, 1991. 3.

10. 심영철, 오경환, "퍼지소속함수를 이용한 온라인 한글 획 인식", 한국정보과학회 가을학술발표논문집, 제18권 제2호, pp. 301~304, 1991.

11. P. SIY and C. S. CHEN, "Fuzzy Logic for Handwritten Numeral Character Recognition," *IEEE Trans. on Syst., Man and Cyber.*, Nov. 1974, pp. 570~575.

12. F. H. CHENG, W. H. HSU, and C. A. CHEN, "Fuzzy Approach to Solve the Recognition Problem of Handwritten Chinese Characters," *Pattern Recognition*, Vol. 22, No. 2, pp. 133~141, 1989.

- 13. 송정영, 이희열, 아끼즈끼 카게오, "세그먼트의 퍼지화에 의한 수서 한글 인식", 대한전자공학회 추계종합학술대회 논문집, 제14권 제2호, pp. 717~719, 1991.
- 14. T. Yamakawa, "A Fuzzy Neuron and its Application to a Hand-written Character Recognition System," '91 신경망 및 퍼지시스템 응용 워크숍, pp. 245~261, 1991.



김 재 회

1979년 연세대학교 전자공학과 졸업  
 1980년~1984년 미국 Case Western Reserve Univ. Electrical Engineering 석사 및 박사  
 1978년~1980년 금성사 중앙

연구소

1984년~현재 연세대학교 전자공학과 부교수  
 관심분야 : 패턴인식, 지식베이스 관리, 전문가시스템, 펜 컴퓨터 등