

《主 題》

# 온라인 문자인식 기술

이 헌 주

(금성중앙연구소 지능정보 1실)

■ 차 례 ■

- I. 서론
- II. 문자인식 시스템 개발 개요
- III. 인식기 설계

- IV. 인식기 구현
- V. 결론 및 연구방향

## 요 약

펜입력 기능을 장착한 시스템의 문자입력 기능에서 사용자들의 다양한 형태의 필기문자를 인식하여 사용자의 만족을 극대화 시킬 수 있는 고성능의 인식기를 만들기 위해서는 상황과 장소에 따라서 변하는 필기형태를 학습할 수 있는 다이나믹한 인식기 개발이 절대적으로 필요하다고 판단된다. 또한 국제 경쟁력을 갖기 위해서는 사용자의 국적(nationality)에 따라 다양한 자종을 인식할 수 있는 기능이 있어야한다. 따라서 본 논문에서는 다양한 사용자의 요구를 만족시키기 위하여 인식기 설계시 필요한 design issues와 philosophy를 제시하고, 이를 기초로 학습기능, 국제화, 소형화의 특징을 갖는 학습기능 다중언어(multilingual) 인식기 설계 방법을 제안하고자 한다.

### I. 서 론

1980년대초 미국의 컴팩사가 휴대용 컴퓨터를 발표한 이래 세계의 컴퓨터 업체들은 보다 가볍고 작은 휴대하기 간편한 휴대용 컴퓨터를 개발하고자 노력해 왔다. 1990년대에는 부품과 디바이스의 경량화, 소형화 기술과 이들의 결합기술이 급격히 발전되어 휴대용 컴퓨터는 더욱 작고 가볍게 만들어질 수 있었다. 그러나 사용자는 이에 만족하지 않고 기존의 노트북 컴퓨터, 팜톱PC와 같은 휴대용 컴퓨터로는 대체할 수 없는 빠른 데이터 처리능력, 펜입력장치 및 통신기능을 보유한 보다 휴대하기 편리한 새로운 휴대용 정보기기를 요구했다. 이러한 사용자의 요구에 부합하는 휴대용 정보기기를 일반적으로 PDA(Personal Digital

Assistant)라 부른다[1]. PDA는 펜, 통신, 운영체제 및 특정 응용 소프트웨어를 통한 첨단 기술의 통합체로서 기술적 문제를 상호 보완하기 위해 필요한 기술을 가진 회사와 기술 제휴를 맺고 공동 개발하는 것이 특징이다.

그동안 컴퓨터의 소형화를 위한 발전에는 키보드 입력장치의 장벽으로 한계를 맞고 있었다. 즉 컴퓨터의 문자입력 수단인 키보드를 더 이상 소형으로 만들 수 없다는 점이다. 물론 현재의 기술로 이보다 작은 키보드를 만드는 것은 아무런 문제가 없지만 현재의 노트북에 사용하는 키보드도 불편을 느끼고 있는데 더 작은 키보드는 사용자가 사용할 수 없기 때문이다. 키보드를 사용하는 경우 항상 컴퓨터를 고정된 상태에서 사용해야 하기 때문에 이동하는 작업자는 키보

드로 컴퓨터를 사용할 수 없다. 이러한 사용자의 욕구를 만족시키기 위해 새로운 입력방법을 생각하게 되었다[2].

키보드를 이용하지 않고 문자를 입력하는 방법으로 온라인 문자인식이나 음성인식을 사용하고 있다. 그중에서 본 논문에서는 온라인 문자인식을 이용한 입력 방법에 대해 다루고자 한다. 이 방식은 연필로 종이 위에 글씨를 쓰듯 입력도구인 전자펜으로 액정 화면위에 글씨를 쓰면 그 글자들을 시스템이 인식해 액정화면위에 나타내 주므로 사용자의 컴퓨터에 대한 심리적인 거부감을 제기시켜 줄 수 있어 그동안 컴퓨터 사용에 제외되었던 일반 사용자 계층을 고객 화함으로써, 컴퓨터의 보급 촉진을 이룰 수 있을 것이다. 인간은 이미 어릴적부터 펜의 사용을 배워왔기 때문에 이것이 가장 친숙하고, 자연스런 입력 수단이다. 펜은 키보드나 마우스와는 달리 편편한 정지된 판이 필요없고, 앉아서도 사용할 수 있고, 서서도 사용할 수 있다. 특히 펜만의 유일한 장점인 글씨의 입력 뿐만 아니라, 그림의 입력이 자유스러워 약도나, 싸인등의 입력이 가능하다. 또, 펜의 장점을 살린 새로운 소프트웨어 개발에 영향을 주어 사용자에게 새로운 컴퓨팅 환경을 만들어 주게 되고, 더 많은 컴퓨터 사용자를 만들어낼 수 있게 된다.

그러나 펜입력 기능을 장착한 시스템의 문자입력 기능에서 사용자들은 인쇄체와 흘림체에 대해 높은 인식율을 요구하고 있기 때문에, 개발시 미리 정의된 필기체의 Style만을 인식하는 사용자 독립형(user independent) 시스템으로는 다양한 필기자의 다양한 필기 형태를 수용하는데 한계가 있다고 보고, 다양한 형태의 흘림을 인식하여 사용자의 만족을 극대화 시킬 수 있는 고성능의 인식기를 만들기 위해서는 상황과 장소에 따라서 변하는 필기형태를 학습할 수 있는 다이나믹한 인식기 개발이 절대적으로 필요하다고 판단된다.

또한 국제 경쟁력을 갖기 위해서는 사용자의 국적(nationality)에 따라 다양한 자종을 인식할 수 있는 기능이 있어야한다. 따라서 본 논문에서는 다양한 고객의 요구를 만족시키기 위하여 인식기 설계시 필요한 design issues와 philosophy를 제시하고, 이를 기초로 학습기능, 국제화, 소형화의 특징을 갖는 학습기능 다중언어(multilingual) 인식기 설계 방법을 제안하고자한다.

## II. 문자인식 시스템 개발 개요

### 1. Design Issues & Philosophy

시장조사 결과 사용자들은 필기시 왜곡된 문자나 다양한 필기 습관을 흡수할 수 있는 문자 인식을 요구하고 있다. 그러나 현재의 문자 인식기술은 필기시 제약을 가하거나, 조심스럽게 필기하는 경우 높은 인식율을 얻을 수 있으나, 그렇지 않은 경우 기대 이하의 성능을 발휘한다. 인식기 개발에서 실질적인 어려움은 필기 형태가 사람마다 다르고, 심지어 동일 사람도 그때그때 상태에 따라 필기 형태가 변한다는 사실이다. 따라서 인식기 설계시 가능한 강인하고(robust), 다양한 필기형태(writing styles)를 허용하며, 가능한 최대로 필기변형(writing distortion)을 허용할 수 있는 설계철학을 가져야 할 것이다. 즉 사용자 입장에서 인식기를 설계해야 한다는 것이다.

이러한 위해 인식 시스템 설계시 고려해야 할 몇가지 사항을 기술하였다[3].

#### 가) Recognition Consistency

일반적으로, 사용자가 새로운 제품을 구입할 경우 그 제품의 동작방법에 익숙하기 위해서는 얼마간의 시간이 요구된다. 이러한 적응 훈련은 그 시스템이 논리적이고, 일관성(consistency) 있게 동작할 때 쉽게 빨라진다. 특히 문자인식 시스템에서 인식 결과의 일관성은 매우 중요한 사항이다. 외관상 거의 같게 필기된 문자가 서로 다른 문자로 인식할 경우 사용자는 시스템을 불신하게 되고 결국 사용을 포기할 것이다. 예로써, 한 시점에서 사용자가 문자 "a"를 필기하고, 인식 결과도 "a"가 되었으나, 다른 한 시점에서 외관상 거의 같은 모양으로 "a"를 필기하였을 때, 그 인식 결과가 "d"인 경우 사용자는 시스템을 불신하고 사용을 기피할 것이다. 따라서 인식기 설계시 인식의 일관성 문제를 중요하게 고려해야 할 것이다.

#### 나) Non-recognition vs Mis-recognition

미인식(non-recognition)이란 인식기가 판단을 할 수 없을 때 발생되며, 필기한 문자를 인식하지 못함을 알려주는 것이다. 오인식(misrecognition)은 시스템이 실수하여 마치 정확히 인식한 것처럼 결과를 출력하는 것이다. 예를 들면 문자 "a"를 필기하였는데 "d"로 인식되는 경우이다. 여기서 오인식의 경우 사용자에게 경고없이 부정확한 인식결과를 제공함으로써 미인식보다 문제가 훨씬 심각하다. 반면, 미인식이 발생할 경우는 시스템이 문제가 있다는 것을 사용자에게 알려줌으로써 심각한 상황을 사전에 방지할 수 있다. 따라

서 인식기 설계시 가능한 미인식 기능을 충분히 고려해야 할 것이다.

**다) Trainability**

시스템이 사용자에게 자동으로 적응할 수 있고, 사용함에 따라 점점 성능이 향상되는 학습의 개념은 매우 매력적인 기능이다. 그러나 이 기능은 상당히 복잡하며, 현재 용어도 표준화되지 않았다. 일반적으로 인식기 구현은 사용자 학습형(user trainable)과 사용자 독립형(suer independent)의 두가지 방식으로 분류된다. 전자의 단점은 학습이 정확하지 않으면, 시스템의 인식성능이 나빠지고, 또다른 문제는 사용자의 필기 형태가 시간이 지나면서 변하게되어 계속적인 추가 학습이 요구되는 것이다. 후자의 단점은 인식기가 인식할 수 있는 필기형태가 고정되어 변경이 불가능하여, 사용자가 이 시스템 사용에 어려움이 있더라도 시스템에 사용자가 적응하여야 한다는 것이다. 가장 바람직한 방식은 각 방식의 장점을 조합하여, 개인의 독특한 필체의 적응기능을 살리면서 최소의 학습을 요구하는 사용자 적응형(user adaptive) 방식으로 생각된다. 즉 대다수의 사용자가 별도의 추가 학습없이 사용할 수 있는 수준의 인식 시스템에, 인식하기 어려운 문자를 적용시키기 위한 학습기능이 제공되는 것이 가장 이상적인 시스템이라 생각한다. 또한 학습 방법에는 비지도(unsupervised) 학습과 지도(supervised) 학습이 있는데 전자의 경우 사용자 입장에서 가장 편리한 방식이나 오학습의 위험이 높기 때문에, 보다 안전하고 안정된 후자가 바람직하다.

**라) Linguistic Post Processors**

현재 다수의 펜 운영체제들이 인식기의 단어단위의 인식률을 향상하기 위하여 spell checkers를 제공하고 있다. 그러나 spell checkers는 일반적으로 많은 메모리를 요구하고, 추가 처리로 인해 인식시간이 늦어지는 단점이 있다. 뿐만 아니라 인식된 단어가 사전에 없는 경우 spell checker는 가장 유사한 단어를 출력함으로써 오히려 오인식을 유발할 수도 있다. 따라서 spell checker는 성능 좋은 문자 인식기의 대체 차원이 아니라, 메모리가 문제되지 않고 사전 사용이 적합한 응용분야에 보조적으로 채택하는 것이 바람직하다.

**마) International Solutions**

펜 인터페이스는 기존의 문자입력과 포인팅 기능을 위해 필요한 키보드와 마우스를 하나의 장치로 대

치하는 장점은 물론, 마우스로는 어려운 그림이나 그래픽의 자연스러운 작성을 할 수 있는 기능을 제공한다. 이러한 장점들은 문자 입력에 있어 영어, 유럽 언어들, 일본어, 한자, 그리고 한글등 모든 종류의 언어에 적용이 가능하며, 따라서 international solution 제공을 위해 다양한 언어를 인식할 수 있는 다중언어(multilingual) 인식기 개발이 필요하다.

**2. 인식기 개발 동향**

링크정보를 문자로 변환시키는 온라인 문자인식 기술은 펜 관련업체에서는 중요한 문제로 되어 왔다. 특히 사용자들은 정사체뿐만 아니라 흘림체에 대한 높은 인식률을 기대하고 있기 때문에, 현재 개발된 많은 필기체 인식 시스템들이 사용자들의 기대대로 제대로 동작될 것인지 의문을 제기하는 사람도 있다. 일반적으로 필기체 인식은 패턴의 변화가 크기불문항 상을하기가 어렵고, 많은 펜 관련업체의 개발자들은 인식률 향상을 은하여 각자 다양한 방법을 사용하고 있다. 일반적으로 문자인식 기술은 neural network, HMM (Hidden Markov Modeling), 단순한 template matching 등이 있다. 상용화된 제품으로 CIC의 Handwriter, Lexicus의 Longhand, Nestor의 NestorWriter, 그리고 Apple사의 Newton에 사용된 Paragraph(U.S./Russia joint venture)의 Calligrapher 등이 있다. 다음은 대표적인 인식기의 특징들을 기술하겠다[4].

**가) Lexicus's Longhand**

Lexicus는 자사의 인식기술에 대해 언급을 꺼리고 있으며, 이 제품을 면기위 여석한 전문가에 의하면 제한없는 필체를 인식하기 위하여 Harabach 기초한 모델을 사용하는 것으로 판명되었다. Horabach 모델은 curve에 기초한 손이 움직이는 모양의 집합을 정의한 것으로, 특히 흘림체 인식에 강한 반면, 정사체에는 약한 단점이 있다. 현재 Lexicus는 Motorola에서 인수한 것으로 알려지고 있다.

**나) Paragraph's Calligrapher**

Paragraph 역시 인식기술의 노출을 꺼리고 있어, 사실이 알 수 없으나 보고에 의하면 이 제품은 패턴의 모양을 이용하는 것보다 직관적인 방법을 이용하고 있다. 각 획을 8가지 정의된 기본 패턴에 matching 시키고, 이들을 차례로 20~30개 정도의 subalphabet으로 분류한 후 최종적으로 특정언어에 속하는 문자에 mapping시킨다.

다) IBM Handwriting Systems

IBM은 서명 검증(signature verification)과 무제한 discrete 필기체를 처리하기 위한 인식기를 개발하고 있으며, 이를 위하여 elastic matching과 neural network 기술을 결합한 방식을 채택하고 있다. 인식과정에서 상호보완 특성을 가진 두 방법을 병렬로 수행시켜, 하나의 방법으로 구현된 것보다 정확도에서 우수하다고 주장한다. 그리고 이 혼합 방식의 인식기는 인식율 향상을 위해 사용자가 학습시킬 수 있도록 구현하였다. 즉 IBM의 기본 전략은 학습을 통하여 인식 성능을 향상시키는 것이다. 각 제품들의 자세한 사양에 대해서는 <표1>에 자세히 기술하였다.

Ⅲ. 인식기 설계

1. 인식기 기본 사양

본 논문에서 제안하고자 하는 인식기의 각 언어의 특수성을 고려하여 다중언어 인식이 가능한 인식 시스템이다. 이러한 기능의 구현을 위해, 사용하는 코드는 완성형 코드(KSC-5601-1987)를 기준으로 하였다. 따라서, 인식기에서 처리하는 문자는 완성형 table에 있는 모든 문자를 인식하는 것으로 하였다. Table내에는 숫자, 기호, 한글 자소, 영어, 한글, 한자, 일본어가 포함되어 있다. 여기서 인식의 단위가 되는 의미단위를 정의한다. 의미단위는 문자 인식의 최소단위로 의

표 1. 인식기 비교

상품명	Handwriter	Longhand	WfPC Recognizer	Nestror Writer	Calligrapher
회사명	CIC	Lexicus	Microsoft	Nestor	Paragraph
필기인식					
인쇄체	O	X	O	O	O
혼합체	X	O	?	?	O
인식문자					
영어	O	O	O	O	O
일본어	O	X	X	X	X
유럽문자	O	X	O	O	X
인식특징					
사진사용	O	O	O	O	O
획순무관	O	O	O	O	O
위치무관	O	?	O	O	O
후보문자	X	?	?	?	O
학습가능	O	?	O	O	O
인식속도 (386/20 기준)	6 cps	4 cps	10 cps	20 ~ 25 cps	3 ~ 4 cps
RAM required	알수없음	200 KB	300 KB	300 KB	130 KB
인식률					
초기 문자	92%	n/a	92%	92%	n/a
초기 단어	n/a	85%	n/a	n/a	88 ~ 90%
학습후 문자	>98%	n/a	98%	>97%	n/a
학습후 단어	n/a	n/a	n/a	n/a	>90%

미 부여가 가능한 획들의 집합으로 정의한다. 따라서 글자인식은 먼저 의미단위를 인식하고, 인식된 의미단위를 조합하여 글자를 완성하게 된다. 영어, 일본어, 숫자, 기호는 의미단위가 곧 글자를 의미한다. 그러나 한글의 경우, 조합 방식으로 글자를 완성하게 되므로 자소를 의미단위로 정의하게 된다. 한자도 부수를 의미단위로 나눌 수 있지만 본 연구에서는 편의상 한글자 자체를 의미단위로 정의하였다. 글자내의 홀림을 수용하는 경우, 의미단위를 사용하는 것이 보다 일반적인 시스템을 구현할 수 있다. 즉 한글의 경우 의미단위가 일반적으로 자소가 되지만, 홀림이 있는 경우, 여러 개의 자소나 하나의 글자가 될 수 있다. 기타 언어의 경우, 조합형 글자가 아니기 때문에 글자가 의미단위가 되지만, 경우에 따라 여러글자를 흘려쓰는 경우, 이 문자열을 하나의 의미단위로 생각할 수 있다. 본 시스템의 데이터베이스는 한글의 자소들에 대한 데이터와, 한자, 영어, 일본어, 기호의 글자에 대한 데이터를 갖게 된다. 결과적으로 한글의 경우, 하나의 자소를 학습하면, 그 자소를 포함한 모든 글자에 영향을 미치게 된다. 전체적으로 볼 때 구현하는 시스템은 다중언어에 관한 정보를 단일 데이터베이스에 갖고 있으면서 의미 단위에 따라 다중언어 인식 및 다중언어 및 후보 문자 출력이 가능하다.

## 2. 학습형 다중 언어 인식기 구조

그림1은 학습형 다중 언어 인식기의 전체적인 구성을 보인 것이다. 먼저 인식의 단위가 되는 Cell에 대해 살펴보면, Cell은 1개 이상의 획으로 이루어진 입력획의 집합으로, 특징 추출과 인식은 이 Cell을 기준으로 이루어진다. 구성은 입력된 Cell의 데이터 중에 의미 없는 점을 없애기 위한 거리 필터링 과정을 거치게 되고, 필터링 이후의 데이터를 가지고 획의 모양 변화가 시작되는 점을 찾아 영역을 분리하는 영역 분리과정과 분리된 영역에서 특징값을 만드는 특징 추출을 하게 된다. 여기까지의 과정이 Cell을 인식하기 위한 처리 단계이고 이 과정들을 입력 처리부라 부른다. 또 이렇게 만들어진 특징값은 의미단위 인식 과정에서 이미 데이터 베이스에 저장되어 있는 의미단위의 특징값과 비교하여 Cell을 인식하고 유사도에 따라 인식된 의미단위를 결과로 출력한다. 그후 의미단위 조합 과정을 통해 의미단위 인식에서 인식된 의미단위를 각 의미단위의 위치에 따라 조합, 인식하여 결과적으로 인식된 문자와 후보문자를 출력하게 된다.

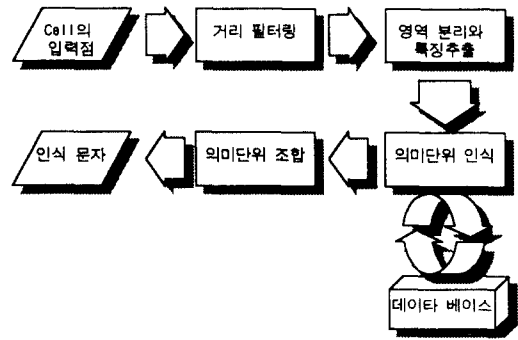


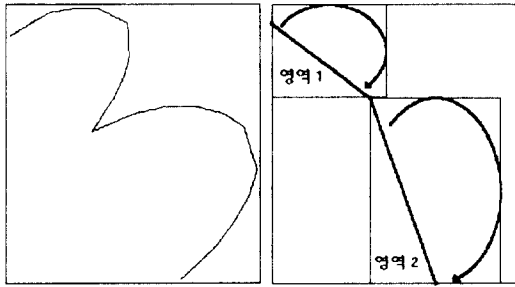
그림 1. 학습형 다중언어 인식기 구조

## 3. 입력 처리부

입력처리부는 그림에서 보인 것과 같이 거리 필터링, 영역 분리와 특징 추출의 2단계로 나누어져 있다.

### 가) 거리 필터링

일반적으로 입력된 데이터를 가지고 특징 추출을 하기 이전의 단계까지의 과정을 전처리라 하는데 인식 방법에 따라 보통 거리 필터링, 각 필터링, 혹은 제거 같은 과정을 거친다. 거리 필터링은 입력된 x, y 점들 중에 중복된 점들과 일정한 거리안에 있고 제거해도 전체적인 모양에 큰 영향을 주지 않는 점들을 제거해주는 과정이고, 각 필터링은 전체적인 모양의 왜곡을 최소화시키면서 획을 이루는데 필요한 최소한의 점들만 남기기 위해 연속된 두 직선 사이의 각이 일정 각 이하이면 제거해 주는 과정이다. 훅(hook) 제거부는 획을 쓰기 위해 펜을 태블릿에 대는 순간과 떼는 순간에 발생하는 삐침, 즉 훅을 제거한다. 본 연구에서는 전처리 과정으로 단지 거리 필터링만 사용한다. 그 이유는 사용하는 특징 추출과정이 각 필터링의 역할을 하기 때문에 전처리에서 각 필터링을 따로 할 필요가 없었다. 또 훅 처리는 입력된 획에서 훅을 판단하는 기준이 애매하기 때문에 인식과정에서 처리하는 것이 좋을 것으로 여겨져 사용하느 특징과 인식 과정에서 훅을 흡수하도록 하였다. Cell이 입력처리부에 들어오면 거리 필터링에서 중복된 점들과 일정한 거리안에 존재하는 점들은 제거를 한다. 거리 필터링을 통해 필기 속도에 상관없이 입력점들을 일정한 시간 간격으로 샘플링하는 효과를 얻을 수 있다. 따라서 필기속도에서 오는 변화를 흡수할 수 있게 되고, 또한 입력점의 갯수를 줄여 뒤에서 행해지는 계산량을 줄일 수 있다.



(a) 거리 필터링 후 분자 (b) 영역 분리 후

그림 2. 거리 필터링 및 영역 분리 과정

나) 영역 분리와 특징 추출

거리 필터링을 거친 입력 데이터들에서 특징 추출 과정을 거치게 되는데, 이 과정은 입력 획을 이루고 있는 부분 중에 특징으로 작용할 수 있는 영역을 분리하는 방법으로 이루어진다. 영역을 분리하는 방법은 획의 구성과 각 점들이 움직이면서 만드는 누적각에 따라서 분리하게 된다. 본 연구에서는 여러 획으로 이루어진 Cell에서 각 획은 다른 영역으로 분리하고, 각 획 내의 누적각이 360도 이상이 되거나 획의 방향이 바뀌면 새로운 영역으로 분리한다. 즉 다음 3가지 경우에 새로운 영역으로 분리된다.

- ① 획이 1바퀴를 도는 순간
- ② 시계 방향으로 돌다가 시계반대 방향으로 도는 경우
- ③ 반대로 시계반대 방향으로 돌다가 시계 방향으로 도는 경우

그림2에서는 Cell이 거리 필터링을 거친 후의 그림과(그림 a), 영역 분리과정 이후의 그림을(그림 b) 보인 것이다.

이때 분리된 각 영역에서 대해 특징값을 정하게 된다. 특징값의 기준은 Cell의 크기로, 각 특징값은 Cell의 크기를 15라고 가정하고, 이 값에 비례하도록 값을 정의하여 메모리 요구량 감소와, 또 이 정규화를 통해 특징의 작은 변화를 흡수 효과를 도모할 수 있었다.

본 연구에서는 다음과 같은 특징값을 정의하여 사용하였다.

1) 영역의 크기에 관한 정보 : RT, RB

Cell내에서 분리된 영역의 크기와 위치를 특징화하는 값으로, 영역의 top-left의 x, y값과 bottom-right의 x, y 값으로 이루어진다.

나) 영역에서 획의 시작점과 끝점에 관한

정보 : RS, RE

영역내에서 획의 움직이는 방향을 특징화하는 값으로 영역내의 획의 시작점과 끝점을 Cell에서의 상대적인 위치로 표시한 값이다.

3) 영역내에서 획의 방향에 관한 정보 : RC

영역내의 획의 쓰여진 방향을 특징화한다. 앞에서 영역을 획의 구분과 누적각으로 나누기로 하였다. 이 중에서 누적각으로 나누는 경우, 누적각이 변화하는 방향으로 코딩하고, 그외의 경우에는 영역의 시작점과 끝점이 만드는 직선의 각도로 방향에 관한 특징값을 만든다. 이 경우, 각도를 45도 간격으로 방향을 구분하는 8 방향 코드를 사용한다. 누적각에 의해 코딩은 반시계 방향의 회전을 9번으로 할당하고, 시계 방향의 회전을 10번으로 할당한다.

결과적으로 하나의 영역이 특징하는 특징값은 RT, RB, RS, RE, RC로 정의되어진다. 만일 Cell이 여러 획으로 이루어지고 각 획이 여러 영역으로 이루어져 있으면, 입력된 획의 순서에 따라 각 특징값을 지정하게 된다. 즉 제1획이 3개의 영역으로 나누어지고 각 특징값이 F11, F12, F13으로 이루어져 있고, 제2획이 2개의 영역으로 나누어지고 특징값이 F21, F22로 이루어져 있다면, 전체 2획으로 이루어진 입력의 특징값은 아래 5개의 특징값들로 표현된다.

F = {F11, F12, F13, F21, F22}

4. 인식부

1) 의미단위 인식

이렇게 만들어진 특징값들을 가지고 의미단위를 인식하게 된다. 의미단위 인식은 이미 저장해 놓은 기준 의미단위의 특징값과 Cell의 특징값의 비유사도를 계산해서 비유사도가 가장 낮은 기준 의미단위를 인식된 의미단위로 출력하게 된다. 따라서 기준이 되는 의미단위들이 특징값이 데이터베이스 형태로 저장되어 있다고 가정한다. 데이터베이스는 영역의 개수에 따라 각기 다른 데이터베이스를 가지게 된다. 예를 들어 직선 값이 가장 간단한 모양으로 이루어진 의미단위의 경우에는 영역이 1개이므로 DB1에 저장되어 있고, 복잡한 획이나 여러 개의 획으로 이루어져 5개의 영역을 가진 의미단위의 경우, DB5에 특징값이 저장되어 있다. 인식은 Cell을 영역 분할하고, 각 영역의 특징값을 만든 상태에서 하는데, 비교 알고리즘은 Cell의 총 영역 개수와 같은 개수의 영역으로 이루어

진 데이터베이스 사이에서 이루어진다. 이때 선택된 데이터베이스에는 같은 갯수의 영역을 가지고 있기 때문에 간단한 계산만으로도 Cell을 인식할 수 있다. 특징값 중에서 RC에 관한 비교를 먼저하게 된다. 즉 Cell의 방향 특징값과 기준 의미단위의 방향 특징값을 비교하여 각 영역의 방향 차이의 합이 일정 이상이 되면, 다른 의미단위로 인식하고, 비슷한 것으로 확인이 되면 나머지 다른 특징값을 비교하여 유사도를 계산한다. n개의 영역으로 이루어진 Cell의 i번째 영역의 특징값을  $F_{ci}$ 라 하고, 데이터베이스의 기준 의미단위의 i번째 영역의 특징값을  $F_{ri}$ 라 할때, i번째 영역의 비유사도는 다음과 같은 식으로 표현된다.

$$F_{ci} = \{cRT, cRB, cRS, cRE, cRC\}$$

$$F_{ri} = \{rRT, rRB, rRS, rRE, rRC\}$$

$$E_i = \|rRT - cRT\| + \|rRB - cRB\| + \|rRS - cRS\| + \|rRE - cRE\|$$

또 전체 Cell의 비유사도는 다음과 같은 식이 된다.

$$E = \sum_{i=1}^n E_i$$

결과적으로 전체 비유사도가 계산되는데, 비유사도가 일정한 값 이하가 되면 Cell과 데이터베이스에 있는 기준자소와 유사한 것으로 인식하게 인식 후보로 선정하게 된다. 이러한 방법으로 데이터베이스의 모든 기준 패턴과 비교한 후에 인식 후보를 비유사도에 따라 sorting하게 된다. 이 중에서 가장 작은 비유사도를 갖는 것을 인식결과로 출력하고, 이 이외의 것을 인식후보로 출력한다.

### 2) 의미단위 조합

의미단위 인식 후에는 결과가 의미단위 조합부로 이동한다. 의미단위 조합부에서는 인식된 의미단위를 조합하여 글자를 만들게 된다. 만약 의미단위 인식부의 결과가 정확하다면 의미단위 조합부는 단지 인식된 의미단위를 모으는 역할만 하면 된다. 그러나 일반적으로 의미단위 인식에서 오인식이 일어나게 되므로 의미단위 조합부에서 인식된 의미단위의 검증과 보정 과정이 있게 된다. 의미단위가 하나의 글자로 이루어진 문자의 경우, 이러한 과정은 사전을 통한 검증이 이 부분에 들어갈 수 있다. 이러한 검증 과정이 필요없는 경우, 인식된 의미단위를 그대로 인식된 글

자로 보내게 된다. 이러한 문자의 의미단위 검증은 다른 연구분야이므로 본 연구에서는 제외하였다. 본 연구에서는 의미단위의 조합을 한글에 중점을 두고 구현하였다. 한글과 같이 자소로 이루어진 문자에 대한 의미단위 조합은 대단히 중요하다. 예를 들어 처음 인식된 자소가 중성이면 초성을 잘못 인식한 것으로 생각하고 후보로 인식된 의미단위 중에서 초성으로 보정하는 작업을 할 수 있고, 자음만 세개가 계속해서 인식이 된다면 중간의 자음도 후보 의미단위에서 중성으로 오인식된 것이라고 생각하고 보정을 할 수 있다. 이렇게 의미단위가 글자와 같이 많은 한글의 경우에 대해 좀더 자세히 설명하겠다.

앞에서 설명한 것처럼 한글은 자소를 조합하는 방식으로 글자를 만들게 된다. 따라서 다른 언어와 달리 의미단위 조합이 중요한 과정으로 작용한다. 한글의 경우, 획의 모양은 완전히 같지만 자소의 위치에 따라, 다른 자소로 인식된다. 이러한 문제점을 해결하기 위해 각 의미단위의 위치와 현재 인식된 글자를 완성해 가면서 글자를 만들어 가는 과정이 필요하게 된다. 또한 이러한 과정을 통하면서 가장 확률이 높은 인식결과와 후보가 될 수 있는 인식결과를 만들게 된다. 그림3는 한글의 경우, 이미 인식된 의미단위를 가지고 글자를 만들어가는 글자조합과정을 보인다.

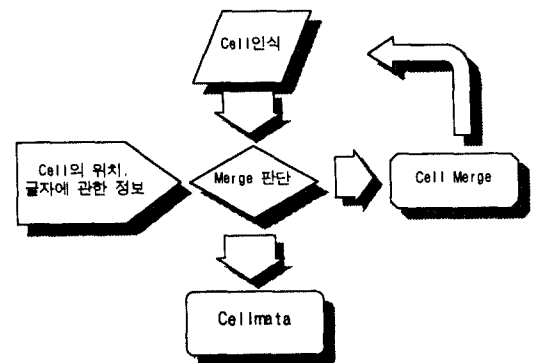


그림 3. 한글 경우 글자조합 과정도

그림3에서 보이는 것과 같이 하나의 Cell에 대한 인식결과를 받게 되면 이 Cell의 인식결과를 위치에 관한 정보, 현재 글자를 이루어가는 과정을 종합하여 받아들일 것인지, 거부할 것인지 판단한다. 만일 받아들일 것으로 결정되면 다음의 Cell과 합하여 보고, 다시

인식을 한다. 이러한 결과를 정해진 횟수만큼 해보고, 최종적으로 하나의 Cell을 결정하고 그 결과를 Cellmata로 보낸다. 여기서 인식된 Cell을 결합하여 글자를 만들게 된다.

### IV. 구 현

#### 1. 구현 환경

지금까지의 대부분의 온라인 문자인식의 연구는 실험실에서 주로 이루어졌다. 그러나 온라인 문자인식과 같이 실제 사용자를 위한 기술은 연구결과 못지않게 사용자를 위한 사용환경 구현이 중요한 문제가 된다. 온라인 문자인식이 상용에 적용되기 시작한 91년대부터 이러한 상품들이 선보이기 시작하였다. 먼저 키보드가 필요없는 펜 컴퓨터와 이러한 펜 컴퓨터를 사용할 수 있게 만드는 펜 전용 운영체계가 발표되었다. 개발된 운영체제로는 Go사의 PenPoint, 마이크로소프트사의 Pen Windows가 있는데 이 중에서 타윈도우 프로그램과의 호환성과 프로그램 개발에 있어서 Pnc Windows가 우수한 것으로 알려져 있다[5, 6]. 운영체계에 관한 국내의 상황은 이미 1993년 10월에 한글 펜윈도우 1.1 버전이 금성중앙연구소에 의해 개발되어, 한국 마이크로소프트가 발표하였다[7]. 이러한 상황에서 본 연구는 한글 펜윈도우의 기준에 맞추어 개발되었다. 따라서 펜윈도우가 정해놓은 모든 규격을 그대로 사용하게 되므로 윈도우가 설치되어 있는 시스템에 한글 펜윈도우를 설치하였다면 만들어진 시스템을 모든 윈도우 응용 프로그램에서 그대로 사용할 수 있다.

#### 2. 입력 방법

한글 펜윈도우가 사용하는 코드가 한글 완성형을 사용하고, 본 연구에서 개발한 인식기 역시 완성형의 코드에 있는 문자를 인식할 수 있으므로 모든 문자의 처리가 가능하다. 그러나 인식을 향상과 사용자의 편의를 위해 우선순위와 인식문자선택의 옵션을 처리하게 하였다. 우선순위는 여러 종류의 글자가 인식되었을 때, 가장 높은 우선순위를 주고 출력문자의 종류를 결정하게 된다. 인식문자 선택은 자주 사용하지 않는 문자는 disable하여 평소에는 사용하지 못하게 하고, 필요한 때에 enable하여 그 종류의 문자를 인식하게 된다. 이렇게 해서 평소에 자주 쓰는 문자의 인식을 높일 수 있다. 보통 국내에서 사용하는 문자 입력 습관을 확인한 결과, 한글의 경우 평소에 한글, 숫

자, 영문을 주로 사용하고, 특별한 경우 한자와 일본어를 사용하므로 인식문자 선택에 한자와 일본어 버튼을 넣었고, 우선순위에는 모든 문자에 대해 우선순위를 정할 수 있게 하였다. 그림에 인식문자선택과 우선순위를 제어하는 툴인 도구함의 구성을 그림4에 보였다.

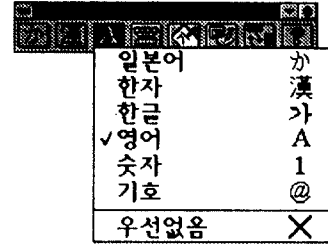


그림 4. 도구함 구성

#### 3. 학습방법

학습은 자신만의 독특한 필체의 인식이 가능하도록 인식기에 자신의 필체를 등록하는 것이다. 한글 펜윈도우는 사용자가 자신의 필체를 학습시키기 위해 그림5, 그림7과 같은 학습기를 제공한다. 학습기 화면에는 펜이 움직인 자취, 즉 잉크를 다시 보여주고, 그 잉크를 어떠한 글자로 인식한 것인지 지정할 수 있는 사용자 인터페이스를 제공한다. 인식과 데이터베이스

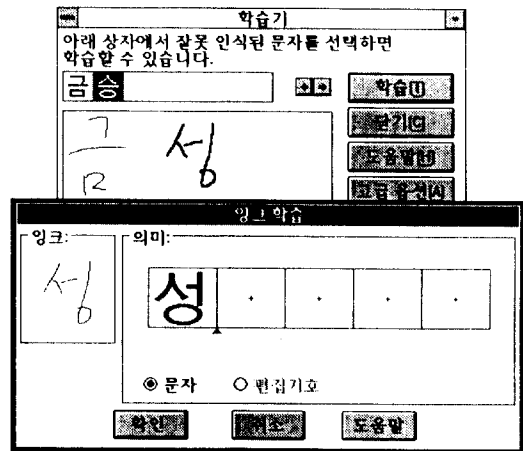


그림 5. 학습 인터페이스 구조



스가 의미단위를 기준으로 이루어지므로, 학습기에서 잉크를 문자로 지정하게 되면 인식기에서는 지정된 문자를 의미단위로 바꾸어 데이터베이스에 저장하게 된다. 따라서 영어, 한자, 일본어, 숫자와 같이 의미단위와 글자가 같은 글자는 그대로 쓰여진 잉크의 특징값과 의미단위를 데이터베이스에 저장하고, 이를 위한 학습과정도 그림7과 같이 문자단위로 학습하는 것으로 종료된다. 그러나 한글과 같이 의미단위와 글자가 같지 않은 경우, 인식기는 그림6과 같이 의미단위, 즉 자소를 지정하는 대화상자를 열게된다. 그림에서 보이는 것과 같이 화면에 인식된 결과를 자소단위로 분리를 해서 보이고, 그 중에서 잘못 인식된 자소를 선택해서 원하는 자소를 지정한 후에 학습을 시키게 된다.

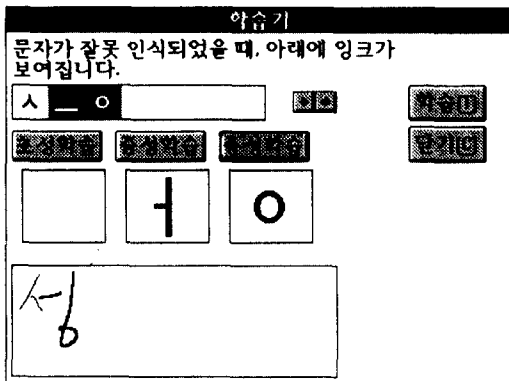


그림 6. 자소 학습 인터페이스 구조

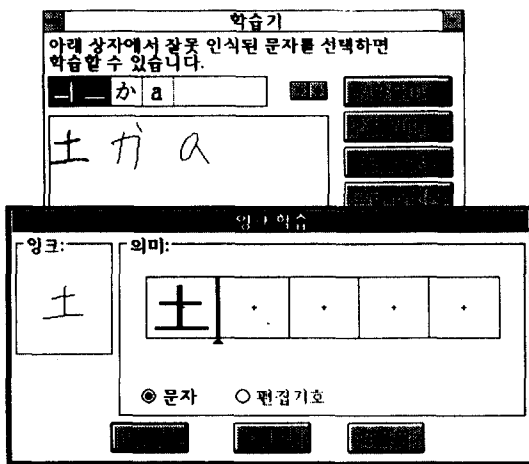


그림 7. 의미단위와 글자가 동일한 경우 학습 인터페이스

#### 4. 오학습 문제 및 해결 방법

학습 기능이 있는 인식기에서의 가장 큰 문제점은 사용자가 잘못 등록한 패턴에 의해 전체 인식률이 떨어지는 경우가 생기는 것이다. 예를 들어, 'ㄷ' 모양으로 쓴것을 'ㄱ'이라 학습하게 되면 'ㄷ'이라고 쓴것이 'ㄱ'으로 인식하는 경우가 발생한다. 이러한 문제점을 해결하기 위해 인식기 내부에 데이터베이스를 관리할 수 있는 대화상자를 만들었다. 이미 데이터베이스에 있는 의미단위를 보이고 이 중에서 잘못 입력된 패턴을 삭제할 수 있다. 그림8은 이 대화상자를 통해 이미 'ㅅ'으로 등록되어 있는 패턴들을 보여준다.

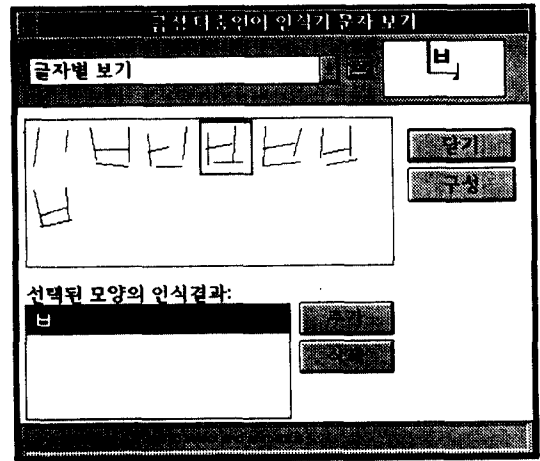


그림 8. 학습 데이터베이스 관리 틀

### V. 결론 및 연구방향

본 논문에서는 휴대용 정보단말기를 포함하는 펜 입력을 필요로 하는 시스템의 문자입력 수단인 온라인 문자 인식 시스템 개발시 요구되는 design issues와 philosophy에 대해 기술하였으며, 이를 바탕으로 학습기능, 국제화, 소형화 전략을 갖고 국제적 경쟁력을 가질 수 있는 온라인 문자인식기의 설계기술에 대하여 기술하였다.

본 연구에서 제안한 온라인 문자 인식 방법은 의미단위로 인식을 하기 때문에 기존의 인식기와 달리 다중언어를 처리할 수 있는 커다란 장점이 있다. 또한 인식방법이 사용자가 학습을 통해 이루어지기 때문에 개인의 필체를 충분히 인식할 수 있는 장점이 있

다. 인식 방법에 있어서도 하나의 인식결과를 출력하는 것이 아니라, 인식결과 및 후보결과를 출력할 수 있으므로 비록 오인식 되었더라도 다음 후보 결과로 글자를 구성해 봄으로써 오인식을 정정할 수 있는 기능을 가질 수 있게 만든다. 특히 사용하는 특징이 단순한 획과 복잡한 획에 대해 충분한 정확도를 가지고 인식을 할 수 있기 때문에 한글의 복잡한 흘림 글자 뿐만 아니라, 한자, 일본어, 영문의 필기체도 인식할 수 있어 다중언어 기능을 가지게 된다.

구현에 있어서도 본 연구는 상품화를 염두하여, 운영체계를 펜 전용운영체계를 사용하였을 뿐만 아니라, 요즘 가장 많이 사용되는 윈도우 환경을 사용하여 사용자가 모든 윈도우 응용프로그램에 본 인식기를 사용할 수 있게 구현하였다. 이러한 관점에서 연구의 진행도 인식속도와 다른 운영체계의 이식성을 충분히 고려하여 설계하였다. 따라서, CPU의 성능이 빠르지 않은 개인용 컴퓨터에서도 본 인식기를 충분히 사용할 수 있고, Object-Orient 개념을 도입하여 다른 운영체제로의 이식도 쉽게 이루어질 수 있게 만들어져 있다.

앞으로 연구방향은 인식을 향상에 있다. 이러한 연구는 의미단위 인식과 의미단위 조합의 부분에 초점이 맞추어진다. 좀더 정확한 의미단위 인식방법을 찾

는 방법과 의미단위를 조합하여 글자를 인식하는데 연구방향을 삼고 있다. 특히 의미단위 조합은 각 언어의 특수성을 고려할 수 있는 영역이므로 각 언어에 맞게 오인식과 미인식을 줄이는 방향으로 연구될 것이다.

### 참 고 문 헌

1. "PDAs Arrive: But Aren't Quite Here Yet", Byte, 1993. 10.
2. Christopher Barr, "Pen PCs", PC Magagine, Nov. 10, 1992.
3. "Handwriting Recognition Research: Panel Note", PenExpo Conference Proceedings, Jan. 1993.
4. Mark Gibbs, "Handwriting Recognition a Comprehensive Comparison", PEN, Mar./Apr. 1993.
5. Microsoft, "Microsoft Windows for Pen Computing: Programmer's Reference", Microsoft Press, 1992.
6. Ray Duncan, "Recognizing Handwritten Input in Pen Windows", PC Magagine, Mar. 17, 1992.
7. 이현주, 조문중, "Pen Windows 상에서의 On-Line 필기인식기술 현황", 한국정보과학회지 제11권 제 5호, 1993년 10월.



이 현 주

- 1983년 경북대학교 전자공학과(학사)
  - 1985년 연세대학교 전자공학과(석사)
  - 1990년 연세대학교 전자공학과(박사)
  - 1990~현재: 금성중앙연구소 책임연구원
- ※ 관심분야: 멀티미디어, 패턴인식, 영상코딩 등임