

□ 특 집 □

## Pen Windows상에서의 On-Line 필기인식 기술 현황

금성중앙연구소 기초 4실 이현주 · 조문증

● 목	차 ●
I. 서 론	3.5 그외의 요소
II. 한글 펜 윈도우	IV. 한글 펜 윈도우의 응용
2.1 한글 펜 윈도우의 장점	4.1 프로그램 기법
2.2 한글 펜 윈도우의 입력 및 인식	4.2 HEDIT와 BEDIT Class
2.3 한글 펜 윈도우의 구성	V. On-Line 필기체 인식기 구현
III. 한글 펜 윈도우의 구조	5.1 인식 Interface와 펜 윈도우의 연결
3.1 화면 드라이버	5.2 인식 Module과 인식 Interface와 연결
3.2 펜 드라이버	5.3 인식기 시장 현황 및 문제점
3.3 RC Manager	VI. 결 론
3.4 인식기	

### I. 서 론

1980년대에는 컴퓨터 기술이 데스크탑에 집중되어 컴퓨터의 대중화에 크게 기여하여 사람들이 매일 컴퓨터를 이용하여 필요한 정보를 가지게 되었고 의사결정에 도움을 주었다. 1990년대에서 이러한 흐름은 개인화라는 방향으로 컴퓨터 기술의 발전을 이끌어 가고 있으며 컴퓨터가 휴대용으로 변화됨에 따라 언제, 어디서나 컴퓨터를 사용할 수 있게 되었다.

개인용 컴퓨터의 기술발전으로 성능은 좋고, 가격은 낮은 PC가 만들어지게 되었고, 결과적으로 PC의 보급이 확대되었다. PC의 발전을 살펴보면 성능은 점점 좋아지지만 반면에 크기에 있어서는 점점 소형화 되는 것이 특징이다. 따라서 과거처럼 일정한 장소에 놓고 사용하는 컴퓨터가 아니라 쉽게 옮길 수도 있고 경우에 따라 휴대할 정도로 작은 랩탑(LapTop) 컴퓨터가 등장하였고, 최근에는 Notebook 컴퓨터의 수요가

급격히 증가하고 있다[1].

그러나 이러한 소형화를 위한 발전에는 키보드라는 입력장치의 장벽으로 한계를 맞고 있다. 즉 컴퓨터의 텍스트 입력 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. 또한 키보드를 사용하는 경우 항상 컴퓨터를 고정된 상태에서 사용해야 하기 때문에 건설 현장 관리자 같이 이동하는 작업자는 키보드로 컴퓨터를 사용할 수 없다.

이러한 사용자의 욕구를 만족시키기 위해 새로운 입력방법을 생각하게 되었다. 이러한 새로운 입력방법으로 보편적으로 사용하는 것은 그림을 통한 사용자 인터페이스(GUI: Graphical User Interface)이고, 그외 키보드를 이용하지 않고 텍스트를 입력하는 방법으로 온라인 문자

인식이나 음성인식을 사용하고 있다.

그중에서 앞으로 많이 사용하게 될 것으로 여겨지는 것은 온라인 문자인식을 이용한 입력 방법이다. 이 방식은 펜이라는 새로운 입력장치를 사용하여 사용자가 컴퓨터를 사용하게 만든다.

펜은 다른 입력장치와 다른 편리한 입력환경을 제공한다. 사람들은 이미 어렸을 때부터 펜의 사용을 배워왔기 때문에 가장 친숙하고, 자연스러운 입력 수단이다. 또 펜은 키보드나 마우스와는 달리 편편한 정지된 판이 필요없고, 앉아서도 사용할 수 있고, 서서도 사용할 수 있다. 특히 펜만의 유일한 장점은 글씨의 입력뿐만 아니라, 그림의 입력이 자연스럽다. 예를 들어 약도라든지, 싸인 등의 입력이 가능하다.

현재 사용하고 있는 GUI, 소형화 기술, 온라인 문자인식 기술, 데이터 압축기술, 반도체 기술을 접목하여 펜 컴퓨터를 만든다면 컴퓨터 분야의 개인화와 휴대화에 큰 변화가 생길 수 있을 것이다. 또 펜의 장점을 살린 새로운 소프트웨어 개발에 영향을 주어 사용자에게 새로운 컴퓨팅 환경을 만들어 주게 되고, 새로운 컴퓨터 사용자를 만들어 낼 수 있게 된다. 예를 들어 병원에서 펜 컴퓨터를 이용하면, 항상 이동하는 간호사와 의사 사이에 컴퓨터 통신을 이용한 데이터 교환이 가능하고, 특히 병상 기록에는 글씨 뿐만 아니라 그림이 들어가기 때문에 기존의 컴퓨터로는 병상 기록의 전산화가 불가능한 것을 가능하게 한다.

이러한 이유로 몇 회사가 펜 컴퓨터를 개발하였고, 온라인 인식을 위한 운영체제를 개발하기 시작하였고 몇가지 운영체제가 발표되었다.

펜 컴퓨터는 LCD와 입력장치인 태블릿을 결합한 디스플레이 장치를 가지고 있다. 태블릿에서 펜의 위치를 인식한 다음 운영체제가 펜의 움직인 자취를 화면에 보이고, 동시에 이 자취를 가지고 필기된 글씨를 인식하여 문자로 표시하는 역할을 한다. 현재 발표되어 있는 하드웨어로는 처음 Pen Computer를 개발한 GRiD사의 GRiD Pad SL이 있고, NCR의 3125, 3130, NEC의 UltraLite SL /20이 있다. 국내에서 생산되고 있는 펜컴퓨터는 삼성전자의 PenMaster 386L/20이

있고, 삼보 컴퓨터도 펜 컴퓨터를 발표하였다.

운영체제는 GO사의 PenPoint, Microsoft사의 Pen Windows(정확한 명칭은 Windows for Pen Computing)이 있다. 그외 IBM, Apple, GeoWork사가 각기 자신의 운영체제를 발표하였다.

그중에서 현재 가장 각광을 받고 있고, 많이 사용하고 있는 운영체제는 Microsoft사의 Pen Windows이다. 현재 영문 버전과 일본어 버전이 이미 발표되어 있고, 한국어 버전이 93년 8월말 경에 발표될 예정이다[2].

따라서 본 논문의 Pen Windows 환경에 대해 살펴보고, 특히 한글 펜 윈도우에서 한글 및 영문 온라인 문자 인식 기술을 응용하는 방법을 살펴보기로 하겠다.

## II. 한글 펜 윈도우

### 2.1 한글 펜 윈도우의 장점

한글 펜 윈도우는 한글 윈도우 버전 3.1 이상에서 펜을 사용하여 필기체를 인식할 수 있게 하는 윈도우이다. 한글 펜 윈도우는 다른 펜용 운영체제에 비해 많은 장점을 가지고 있다[3].

우선 윈도우 자체가 가지고 있는 장점은

1) 모든 외부장치는 윈도우와 연결되고 프로그램과는 관계가 없기 때문에 프로그램 개발이 간단하고 범용성이 있다.

2) GUI 기능으로 사용하기 간단하고 멀티 태스킹과 효과적인 메모리 관리로 도스와는 완전히 다른 환경을 지원한다.

3) 윈도우는 많은 부분을 개방하고 개발틀이 많기 때문에 새로운 프로그램 개발이 쉽다. 한글 펜 윈도우는 윈도우의 장점을 가지면서 또한 펜만의 장점을 가지고 있다.

1) 한글 펜 윈도우는 윈도우에 펜 기능을 추가하여 만들었기 때문에 펜 윈도우용 프로그램은 물론 기존의 윈도우 프로그램을 전혀 바꾸지 않고 사용할 수 있다.

2) RC Manager를 제외한 모든 부분(인식기, 사전, 디바이스 드라이버)의 교체가 가능하여 확장성이 있다.

3) 한글 펜 윈도우는 키보드가 할 수 있는 모든 문자 입력을 펜으로 할 수 있게 만들었다. IME (Input Method Editor) 상태에 따라 한글 및 영문의 필기 입력이 가능하고 리턴키와 스페이스 같은 키도 편집기호로 입력이 가능하다.

## 2.2 한글 펜 윈도우의 입력 및 인식

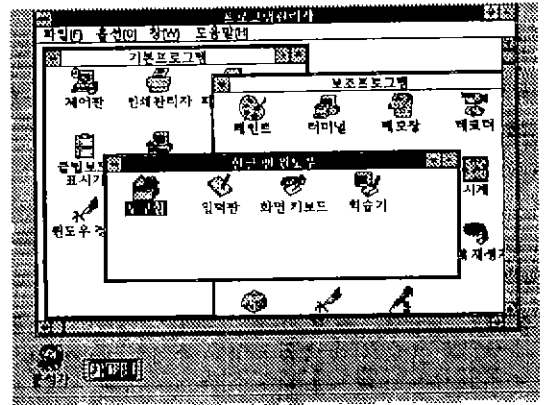
펜 윈도우는 펜의 모든 입력을 잉크로 처리한다. 따라서 일단 입력된 잉크를 글자로 인식할 수도 있고 편집기호로도 인식할 수 있다. 또 약도와 같이 잉크 자체가 의미 있을 경우 그대로 데이터화하여 저장 및 전송이 가능하다.

인식 부분에 있어 펜 윈도우와 인식기는 분리되어 있기 때문에 어떠한 인식기라도 시스템에서 사용할 수 있다. 따라서 펜 윈도우를 바꾸지 않고 인식기만을 바꾸어 여러나라의 언어를 인식할 수 있다. 또 인식기를 위한 규격이 공개되어 있기 때문에 누구라도 자신의 알고리즘으로 인식기를 만들 수 있다. 따라서 필요에 따라 펜 윈도우를 이용하여 문자 인식 뿐만 아니라 서명 인식 시스템도 구현할 수 있다.

이번에 발표될 한글 펜 윈도우 1.1에는 두개의 인식기가 있다. 하나는 마이크로소프트가 만든 영문 전용 인식기 이고, 하나는 금성중앙연구소에서 만든 한글 전용 인식기이다.

두 인식기의 공통적인 특징을 살펴보면 background processing을 통하여 빠른 인식속도를 보인다. 즉 쓰고 있는 동안 Preprocessing과 segmentation 같은 작업을 하게 되어 실제 인식속도보다 빠르게 사용자가 인식 결과를 얻을 수 있다. 예를들어 쓰는 속도가 충분히 느리다면 real time으로 인식결과를 얻을 수 있다. 또 두 개의 인식기로 한글, 영문 대문자, 소문자, 숫자는 물론 일부 기호도 인식이 가능하다.

특이한 점은 한글 펜 윈도우는 박스입력에서 한글 및 영문을 IME 상태를 바꾸지 않고 혼용 입력할 수 있다. 박스입력에서 한글과 영문을 섞어서 쓰면 자동으로 글자의 종류를 분리하여 인식 한다. 그러나 글자의 종류가 분명치 않은 글자는 '도구함'이 정하는 우선 순위에 따라 글자를 인식하게 된다. 예를 들어 박스 입력에서



(그림 1) 한글 펜 윈도우의 초기 화면

'2'를 입력하면 한글 우선 순위로 되어 있으면 '그'가 입력되고 영문인 경우 'Z'가 입력되고 숫자 우선인 경우 '2'가 입력된다.

학습문제에 있어서 영문 인식기는 학습을 허용한다. 처음 설치되는 데이터 베이스에는 일반적인 글씨는 학습되어 있고, 필요에 따라 다른 글자를 추가 할 수 있으며, 이미 학습되어 있는 글자를 삭제할 수 있다. 그러나 학습할 수 있는 글자의 형식이 글자간의 흘림은 허용하고 있지는 않다. 그러나 th와 같이 습관적으로 흘림으로 쓰는 글자는 등록하여 글자열로 mapping이 가능하다. 한글에 있어서는 현재 학습기능을 넣지 않았고 향후 추가 할 예정이다.

## 2.3 한글 펜 윈도우의 구성

한글 펜 윈도우를 설치하면 (그림 1)과 같은 아이콘이 만들어진다.

### 2.3.1 도구함

한글 펜 윈도우는 펜에 관련된 작업을 도구함을 통해 전체적으로 관리한다. 앞에서 설명한 한/영 혼용입력을 위한 우선순위 설정이라든지 펜에 관련된 응용 프로그램을 즉시 수행할 수 있는 단추가 있다. 특히 펜 기능이 없는 기존의 응용 프로그램을 수행할때 도구함을 수행하게 되면 기존의 응용 프로그램을 펜으로 사용할 수 있다. 예를들어 한글 워드 1.2에서 도구함을 수행하면 한글 워드 1.2를 펜용 응용 프로그램처럼

사용할 수 있다.

### 2.3.2 화면 키보드

한글 펜 윈도우는 키보드가 없어도 컴퓨터를 사용할 수 있게 하기 위해 화면 키보드 기능이 있다. 이것을 수행하면 화면에 키보드가 나타나 화면의 키보드를 치면 실제 키보드를 친 것과 같은 효과가 있다. 따라서 이것을 이용하면 Ctrl-Esc 같이 인식으로 불가능한 키 입력도 키보드 없이 사용이 가능하다.

### 2.3.3 학습기

한글 펜 윈도우는 각 개인의 독특한 필체를 인식하기 위해 학습 기능을 넣어 인식률을 향상하였다. 영문의 경우 자신의 필체를 학습시켜 두면 다음에 같은 모양으로 글씨를 쓰면 인식이 가능하다.

### 2.3.4 입력판

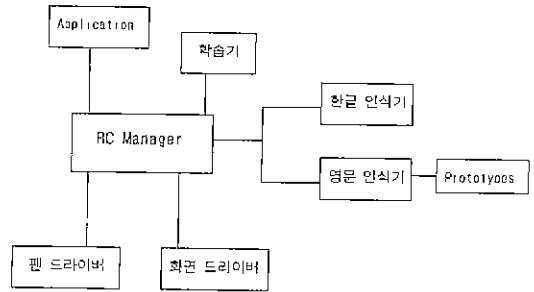
펜 기능이 없는 응용 프로그램에 펜으로 입력하기 위해 한글 펜 윈도우는 입력판이라는 프로그램을 제공한다. 이것을 이용하면 어떠한 응용 프로그램도 펜으로 입력이 가능하다. 즉 입력판에 필기 입력을 하고 OK 단추를 누르면 현재 입력을 받고 있는 응용 프로그램으로 필기된 결과가 입력된다.

### 2.3.5 제어판에 펜 관련 기능 추가

펜에 관한 여러가지 설정을 위해 제어판에 펜에 관한 기능이 추가되었다. 펜의 색이라든지 두께등을 조정하는 기능, 필기입력에 관련된 설정을 정하는 기능을 제어판에서 조절하게 만들었다.

## III. 한글 펜 윈도우의 구조

한글 펜 윈도우는 (그림 2)와 같은 구조로 되어 있다[4,5]. 일반 윈도우의 구조와 달리 RC Manager와 인식기가 추가되었고, 펜 드라이버와 화면 드라이버는 기존의 것을 수정하였다. 수정된 부분은 입력면에서 펜 드라이버에서 펜 데이터를 시스템에 공급한다. 출력면에서 화면 드라



(그림 2) 한글 펜 윈도우의 구조

이버가 펜으로 입력된 자취 즉 잉크를 화면에 보이게 된다. 그외의 환경은 기존의 윈도우와는 아무 관련이 없으므로 다른 하드웨어는 그대로 사용할 수 있다.

펜 윈도우에서는 RC Manager가 핵심을 이루고 있다. 이것이 펜에 관련된 모든 작업을 관리하는 것으로 펜으로 부터 데이터를 받는 것 부터 인식기를 call하는 것과 인식된 결과를 프로그램에 보내는 일을 한다. 인식기는 모여진 펜 데이터를 이용하여 문자 코드를 발생하는 모듈이다.

특이한 점은 위의 모든 부분에서 RC Manager를 제외한 부분을 바꿀 수 있다. 따라서 각 시스템에 맞는 다른 펜 드라이버, 화면 드라이버를 사용할 수 있고, 인식기도 다른것으로 교환할 수 있다. 각 부분에 대해 좀더 자세한 기능은 아래와 같다.

### 3.1 화면 드라이버

화면 드라이버는 인식하는 동안 잉크(ink) 작용을 하게 된다. 잉크는 화면에 펜의 움직임을 의미하는 것으로 필기자가 현재 쓰고 있는 글씨를 볼 수 있게 한다. 또한 화면의 회전을 가능하게 하여 가로 방향 및 세로방향으로 화면을 보이게 할 수 있다. 화면 회전 기능은 펜 컴퓨터에서만 가능하다.

### 3.2 펜 드라이버

펜 드라이버는 태블릿(tablet)으로부터 입력을 관리한다. 즉 펜으로 쓰여지는 입력획을 저장하였다가 RC Manager가 데이터를 요구하면 전달

한다. 또 다른 기능은 펜 윈도우에서는 펜이 마우스 역할을 하기 때문에 마우스 Emulation을 한다. 펜 드라이버는 적어도 초당 120번의 입력을 받을 수 있고, 적어도 200 DPI 이상의 정밀도를 가질 수 있고 다른 태블릿과 드라이버를 사용하게 되면 더 많은 입력점과 높은 DPI를 가질 수 있다. 이 정도라면 온라인 인식에 충분한 데이터가 될 수 있다.

기존의 윈도우와는 다르게 펜 드라이버는 데이터를 마우스의 움직임에 대한 메시지로 받는 것이 아니라 드라이버에서 분리되어 직접 받게 된다. 그리고 펜 데이터는 64K의 크기의 Flat 메모리를 사용하여 저장한다. 이 정도면 약 80초 정도의 입력 데이터를 buffering할 수 있다.

하나의 입력글자의 데이터는 PENDATA라는 Handle로 저장되고 이 Handle내에 PENDATA-HEADER라는 structure가 있어 여기에 입력횟수, 전체 데이터 크기 등의 입력 데이터에 대한 정보가 들어있다. 또 PENDATAHEADER structure에 PENINFO라는 structure가 있어 펜에 관한 정보도 저장된다. PENDATA는 함수를 통해 Offset을 조정할 수 있으며, 크기 변경, 압축 등의 작업이 가능하다.

### 3.3 RC Manager

RC Manager(Recognition Context)는 펜 윈도우가 펜을 이용한 필기 입력을 지원하기 위한 함수와 코드들이다. 이 부분은 penwin.dll 파일에 있는데 이 부분에서 전체 펜 윈도우를 관리하게 되고 필기 입력이 이루어지게 한다. 즉 펜의 입력을 제어하여 펜 데이터를 모으게 되고 이 데이터를 인식기에 전달하여 코드를 만들고 만들어진 코드를 윈도우에 전달하거나 프로그램에 직접 전달한다. 필요에 따라서는 인식된 결과를 사전(Dictionary)에 보내 인식결과를 확인 또는 수정하는 작업을 할 수 있다.

### 3.4 인식기

DLL(Dynamic-linked Library)로 이루어진 이 모듈은 잉크를 글자로 변환할 수 있는 기능이

있다. 인식기가 되기 위해서는 몇가지 조건이 있어야 하는 데 이와 관련된 내용은 뒷부분에서 자세히 다루도록 하겠다. 한글 펜 윈도우에 있는 인식기는 ANSI Character 인식을 위한 MARS.DLL과 한글 인식을 위한 HANREC.DLL, 도형 인식을 위한 SHAPEREC.DLL이 있다. 그외에 필요에 따라 다른 인식기를 추가하거나 다른 것으로 대처가 가능하다.

### 3.5 그외의 요소

펜 윈도우는 필요에 따라 다른 요소들을 추가할 수 있다. 먼저 영문 인식의 경우, 인식이 학습에 의해 이루어지기 때문에 데이터베이스를 가지고 있는데, 이것을 Prototypes이라 한다. 영문 인식기는 입력이 들어오면 Prototypes과 비교해서 글자를 인식한다.

또한 학습을 위한 Interface를 쉽게 하기 위해 학습기라는 프로그램이 있어 사용자가 쉽게 특별한 Style의 필체를 인식기가 학습하도록 한다.

## IV. 한글 펜 윈도우의 응용

위에 보인것과 같은 구조로 한글 펜 윈도우가 이루어져 있다. 이러한 환경하에서 응용 프로그램을 만들어 시스템을 사용하는 방법에 대해 보이기로 하겠다.

펜용 응용 프로그램은 펜의 기능 활용에 따라 3가지로 구분된다.

1) Pen-Capable Application-펜으로 입력된 글씨는 키보드 입력으로 바꾸어 그대로 사용하는 응용 프로그램으로 기존에 사용하던 윈도우 프로그램을 펜 윈도우에서 사용하면 Pen-Capable Application이 된다.

2) Pen-Enhanced Application-기존의 프로그램을 펜의 기능을 생각하여 향상시킨 프로그램으로 박스입력이라든지, 잉크 기능을 넣은 프로그램을 말한다.

3) Pen-Centric Application-완전히 펜 중심으로 이루어진 프로그램으로 펜의 움직임을 정보로

사용하고 글씨 인식 뿐만 아니라 펜의 특수한 기능도 사용하게 만든 프로그램으로 펜용 CAD 프로그램이 대표적인 예이다.

일반적인 윈도우 프로그램에서 인식을 사용하는 방법을 다음과 같은 순서로 이루어진다[6].

1) 사용할 인식을 메모리에 올린다.

2) 펜이 눌러지면 인식에 필요한 옵션을 RC 스트럭처에 넣어 인식을 부른다. 응용 프로그램이 Recognize 함수를 부르게 되면 주도권을 빼앗기게 되고 이때부터 모든 관리는 RC Manager가 하게 되고 인식기와 사전을 이용하여 모든 일을 하게 된다. RC Manager가 인식을 부르면 인식기는 인식옵션에서 지정한 조건에 맞는 펜 입력(Pen Data)를 모으게 된다. 모아진 Pen Data를 주어진 알고리즘을 이용하여 인식하고 그 결과를 RCRESULT라는 스트럭처에 넣어 RC Manager에 보내면 RC Manger가 사전 확인 같은 필요한 작업을 한다.

3) 이렇게 RC Manager관리하에 인식을 통한 결과는 다시 응용 프로그램에 WM-RCRESULT라는 메시지와 함께 보내진다. 이때 인식된 결과 상태와 인식된 결과를 RCRESULT 스트럭처에 담아 온다. 따라서 이 안의 내용을 보고 인식된 글자의 정보를 알아볼 수 있다. 이때 인식기에서 보내는 것은 문자 코드가 아니라 symbol의 형태로 보내게 된다. symbol은 인식기에 따라 인식된 결과를 low-level로 보이게 된다. 현재 시스템에 있는 영문 인식기의 경우 "B"를 쓰면 "B" or "13"의 symbol로 보내 프로그램에서 필요에 따라 B 또는 13을 사용할 수 있다. 한글 인식기의 경우 symbol은 자소인식결과를 보존하기 위해 상용 조합형 코드로 사용하고, SymbolToCharacter 함수를 거치면서 완성형 코드로 변환된다.

4) 2)와 3)의 과정을 반복한 후에 모든 작업이 끝난 후 메모리에서 인식을 내린다.

## 4.1 프로그램 기법

### 4.1.1 인식기 Load

프로그램이 처음 시작하는 때에 인식기를 올

리고 Handle을 가져온다. 시스템이 사용하는 기본 인식기는 윈도우가 시작할 때 설치 되므로 load할 필요가 없다.

```
hrecCur=InstallRecognizer(lpRecogName);
```

### 4.1.2 인식기 call

프로그램의 메시지 처리 루틴에서 다음과 같은 코드가 첨가 된다. 펜이 눌러지는 순간 윈도우가 응용프로그램에 WM-LBUTTONDOWN 메시지를 보내게 되므로 이 순간에 인식을 call할 것인지 아닌지를 결정하고 필요에 따라 선택된 인식을 call한다.

```
case WM-LBUTTONDOWN:
    if (IsPenEvent(message, GetMessageExtraInfo()))
    {
        InitRC(hwndInput, &rc);
        /*시스템에서 정해놓은 인식 Option을 가져온다.*/
        rc.hrec = hrecCur; /*인식을 선택한다.*/
        rc.IPcm|=PCM_TIMEOUT; /*인식되는 순간 결정*/
    }
    break;
```

위의 Option 중에서 인식 순간은 다음과 같은 것으로 변경할 수 있다.

PCM-PENUP : 한 stroke로만 인식하는 경우  
PCM-RANGE : 펜이 태블릿에서 떠날때 인식

PCM-RECTBOUND : 일정한 영역 밖으로 펜이 나갈 때 인식

PCM-RECTEXCLUDE : 일정한 영역으로 펜이 들어가면 인식

PCM-TIMEOUT : 일정한 시간이 지나면 인식

### 4.1.3 인식 결과 처리

인식기가 인식을 끝내게 되면 RC Manager는 WM-RCRESULT 메시지에 인식결과를 넣어 프

로그래밍에 보낸다. 이때 프로그램은 인식기에서 보낸 결과를 해석하여 인식결과를 처리한다.

아래의 프로그램에서

```

case WM_RCRESULT:
    LPRCRESULT    lprcresult=(LPRCRE-
SULT)lParam;
    SymbolToCharacter(lprcresult→lpsyv,
cchMax, szResult, NULL);
    break;
    
```

### 4.2 HEDIT와 BEDIT Class

윈도우에는 텍스트 편집 기능이 있는 프로그램들을 쉽게 만들기 위해 EDIT Class라는 것이 있다. 펜 윈도우에서는 텍스트 편집을 위해 HEDIT(handwriting EDIT control)와 BEDIT(boxed handwriting EDIT)의 두개의 class가 있다. 이 클래스는 인식기와 밀접한 관계를 맺으면서 인식을 위한 명령을 내리거나 인식된 결과를 받아 해석하는 역할을 한다. 또한 키보드의 입력도 받으며 인식된 결과와 키보드 입력으로 얻어진 결과를 메모리속에서 관리하는 작업을 한다. 따라서 이 기능으로 응용프로그램 개발이 쉽게 이루어질 수 있다.

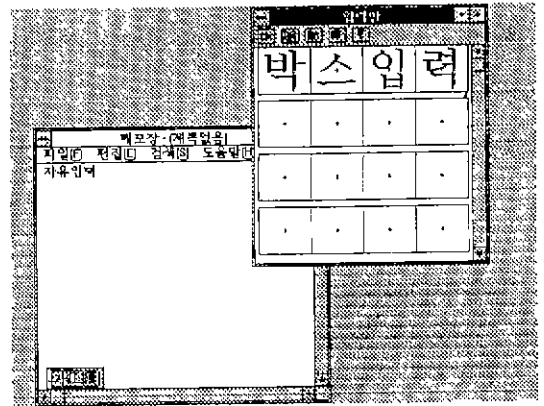
#### 4.2.1 HEDIT Class

HEDIT Class는 윈도우에 있는 EDIT Class의 확장판으로 인식을 위한 Handwriting 기능이 있어 EDIT 클래스가 가지고 있는 기능과 Handwriting Interface 기능이 같이 있게 된다.

펜 윈도우를 수행하게 되면 기존의 프로그램이 가지고 있는 모든 EDIT 클래스는 HEDIT 클래스로 교체가 된다. 따라서 HEDIT는 보통의 EDIT Class와 같은 기능이 있으므로 키보드와 펜 입력을 같이 사용할 수 있게 된다. 한글 펜 윈도우에서는 입력모드에 따라 HEDIT에서 한글 및 영문 gesture 처리가 가능하다.

#### 4.2.2 BEDIT Class

BEDIT 클래스는 박스입력을 하고 편집이 가능한 EDIT 클래스의 일종이다. 화면상으로는



(그림 3) HEDIT와 BEDIT의 예

박스가 있는 사각형안에 한 글자씩 입력하도록 되어 있다. 이것도 마찬가지로 박스 상태에서 한글입력이 가능하고 편집이 가능하며 인식률도 높고 속도도 빠르다. 한글 펜 윈도우에서는 BEDIT 내에서 한글 및 영문 혼용 입력이 가능하다. 즉 한글과 영문을 같이 쓰는 경우 쓸 때마다 모드변환을 하는 것이 아니라 각 박스에 영문과 한글을 쓰면 시스템에서 한글과 영문을 분리 인식하여 결과를 출력한다.

## V. On-Line 필기체 인식기 구현

지금까지 다양한 온라인 문자 인식 알고리즘이 연구개발되었는데 이러한 연구를 상품화된 운영 체제에 인식기로 구현하는 것은 개발된 기술의 실용화라는 측면에서 대단히 중요한 일이다. 즉 개발자는 사용자에게 새로운 컴퓨터 환경을 제공하게 되고, 사용자는 인식기를 사용하여 보고 평가하여 개발자에게 새로운 동기를 부여할 수 있기 때문이다.

펜 윈도우는 이러한 면에서 대단히 개방적이다. 관련된 자료 및 예제 프로그램을 제공하여 누구라도 온라인 인식알고리즘으로 만든 DLL을 가지고 인식을 테스트할 수 있는 환경을 제공한다.

### 5.1 인식 Interface와 펜 윈도우의 연결

인식기를 만들려면 DLL 형태의 파일을 만들

어야 한다. 또한 RC Manager와 Interface를 하기 위해 몇가지 함수를 가져야 한다. 아래에 보이는 함수를 만들어 주어야 하는데 이들 함수는 프로그램이 직접 부를 수 없고 RC Manager를 통해 간접적으로 불러진다[7].

```
int FAR PASCAL LibMain()
WORD FAR PASCAL WEP()
    : DLL이 되기 위한 함수들
BOOL WINAPI InitRecognizer()
    : 인식기를 처음 Install 할때 수행되는 함수
VOID WINAPI CloseRecognizer()
    : 인식기를 없앨때 수행되는 함수
UINT WINAPI ConfigRecognizer()
    : 인식기의 Option을 바꿀때 수행되는 함수
BOOL WINAPI TrainInkInternal()
    : Pen Data를 가지고 학습을 할때 수행되는 함수
BOOL WINAPI TrainContextInternal()
    : symbol을 가지고 학습을 할때 사용되는 함수
REC WINAPI RecognizeInternal()
    : Pen Data를 받으면서 인식을 할 때 사용되는 함수
REC WINAPI RecognizeDataInternal()
    : 이미 주어진 Pen Data를 가지고 인식을 할때 사용하는 함수
```

위의 함수중에서 인식에 사용하는 두 함수의 구현 방법을 살펴보기로 하겠다.

```
REC WINAPI RecognizeInternal(LPRC lprc,
LPFUNCRESULTS lpFuncResults)
{
    while ((rec=GetPenHwData(rgpnt, lpvOem,
cpntMax,lprc->wTimeOut, &si))
        ==REC_OK)
    {
        if(si.wPdk & PDK_DOWN)
        {
            /*좌표점들을 buffering 한다.*/
```

```
        }
        if(한획이 모두 끝났는지?)
        {
            /*한획에 대한 전처리 과정*/
            /*획수 증가*/
        }
    }
    if(rec==REC_TERMTIMEOUT)
    {
        DoRecognition(lprc, hpendata, &rcresult);
        (*lpFuncResults)((LPRCRESULT) &rcresult, rec);
    }
    return rec;
}
REC WINAPI RecognizeDataInternal(LPRC lprc,
HPENDATA hpendata, LPFUNCRESULTS lpFuncResults)
{
    if (! GetPenDataInfo(hpendata, &pendataheader, &peninfo, NULL))
        return REC_NOINPUT;
    for(i=0;i<(int)pendataheader.cStrokes;i++)
    {
        GetPenDataStroke(lppendata, i, &lppoint,
NULL, &si);
        if(si.wPdk&PDK_DOWN)
        {
            /*한획에 대한 전처리 과정*/
            /*획수 증가*/
        }
    }
    rec=DoRecognition(lprc, hpendata, &rcresult);
    (*lpFuncResults)((LPRCRESULT) &rcresult,
rec=REC_OK);
    return rec;
}
```

## 5.2 인식 Module과 인식 Interface와 연결

실제 인식 알고리즘과 인식 INTERFACE의



연결은 다음과 같은 방법으로 한다. 우선 인식하는 문자의 종류를 지정해야 한다. 한글 펜 윈도우의 경우 한글, 영문을 인식 하지만 한자도 할 수 있다. 즉 인식한 결과를 다음과 같이 지정하면 RC Manager가 인식된 결과를 자동으로 처리한다.

RCRT-NORECOG : 인식된 결과가 없음  
 RCRT-DEFAULT : 인식된 결과가 영문, 또는 기호, 숫자일 때

RCRT-GESTURE : 인식된 결과가 Gesture 일 때

RCRT-HANGEUL : 인식된 결과가 한글일 때

펜 윈도우에서는 Symbol을 만들 때 symbol 값과 확신도(Confidence level)을 넣어야 한다. 그래야 RC Manager가 이 부분을 해석해서 결과를 처리한다. 예를들어 인식기에서 rcresult.syg.lpsyv[0].syv = '가' 이고, rcresult.syg.lpsyv[0].cl=65로 넣으면 인식된 글자가 '가' 이고 확률이 65%라는 것을 의미한다. 이 결과를 RC Manager에게 넘기면 현재 정해진 인식 level이 65보다 낮으면 '가' 라는 것을 인정해서 rcresult.lpsyv[0] = '가' 라고 최종 인식을 하고 높으면 인식하지 못한것으로 결과를 보낸다.

```

REC DoRecognition(LPRC lprc, HPENDATA hpendata, LPRCRESULT lprcresult)
{
    if (lprc->lRcOptions & RCO_BOXED)
        /*박스 입력의 경우*/
        HangeulBoxRecognition();
    else /*Free 입력의 경우*/
    {
        do {lpsyv = syeGlobal + CharacterNum;
            CharacterNum++;
            HangeulRecognition();
            /*Free 입력 한글 인식*/
            lpsyv->syv = 한글 Symbol;
            lpsyv->cl = 확신도;
        }
    } while (모든 획을 인식할 때까지);
}
    
```

```

lprcresult->wResultsType = RCRT_HANGEUL;
/*인식된 글자의 종류를 한글로 지정*/
lprcresult->syg.cSye = CharacterNum;
/*인식된 글자의 갯수*/
lprcresult->syg.lpsyv = syeGlobal;
/*인식된 글자들의 symbol과 확률을 저장한다.*/
return REC_OK;
    
```

### 5.3 인식기 시장 현황 및 문제점

현재 많은 회사들이 온라인 인식기를 상품으로 내고 있다. 각 인식기는 각기 다른 특성을 가지고 있다. 그런데 이러한 각 특성이 서로 연관이 있는 것이 특징이다. 예를 들어 학습을 허용하는 인식기의 경우, 초기 인식률이 낮지만 학습으로 인식률이 향상되고, 학습을 허용하지 않는 인식기는 상당히 높은 인식률을 가지고 있다. 또한 글자간 흘림을 허용하는 인식기의 경우, 단일 글씨의 인식률이 낮은 것이 보통이다. <표 1>에 현재 나와 있는 인식기에 대한 비교표를 보았다[8].

무엇보다 온라인 문자 인식에 있어서 가장 큰 문제점은 오인식과 미인식이다. 즉 쓰여진 글자와 다르게 글씨를 인식하는 것과 쓰여진 글씨를 인식하지 못하는 점이다. 이러한 문제점이 발생하는 가장 큰 원인은 사람에 따라 글씨를 쓰는 방법이 너무도 다양하기 때문이다. 따라서 이러한 문제점을 해결하기 위해 많은 인식기는 학습 기능을 넣는다.

이 경우 발생할 수 있는 문제점은 첫째 학습 정도에 따라 인식률은 계속 향상되지만 아무리 학습을 많이 시킨다고 인식률이 100%가 되지 않는다는 점이다. 둘째, 잘못 필기된 글씨를 학습시켜 결과적으로 전체의 인식률을 저하 시키는 경우가 종종 발생한다, 셋째, 한글에서 생기는 문제점으로는 사용자가 자신의 필체에 맞게 학습을 하기 위해서는 한글과 같이 조합 구조를 갖는 글자에서는 조합의 가지 수가 너무 많기 때문에 사용자에게 장시간의 학습을 요구하게 된다.

이러한 문제점을 해결하기 위해 단어별 사전을

〈표 1〉 펜 윈도우용 인식기 비교.

상품명	Handwriter	Longhand	WfPC Recognizer	Nestror Writer	Calligrapher
회사명	CIC	Lexicus	Microsoft	Nestor	Paragraph
필기인식					
인쇄체	O	X	O	O	O
홀림체	X	O	?	?	O
인식문자					
영어	O	O	O	O	O
일본어	O	X	X	X	X
유럽문자	O	X	O	O	X
인식특징					
사전사용	O	O	O	O	O
확순 무관	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%
판매방법	OEM	Bundled	OEM	OEM	OEM

사용하는 방법이 있다. 사람의 경우에도 자별 인식률이 그렇게 높지 않지만 단어 인식률이 높은 것은 사람이 하나의 글자만을 가지고 인식하는 것이 아니라 문맥을 보고 인식하기 때문이다. 따라서 단어별 처리를 통해 글자의 오인식을 막는 작업도 필요하다. 물론 현재 여러곳에서 새로운 연구가 진행되고 있고, 새로운 알고리즘의 적용을 시도하고 있지만 실용화 하기에는 아직 많은 문제가 남아 있다.

또 인식 속도에 관한 문제도 아직 남아있다. 일반적으로 복잡한 처리를 할 수록 인식률이 증가한다고 할 때 인식률과 인식속도가 반비례하게 된다. 그런데 온라인 문자인식을 펜컴퓨터에 응용한다면 인식속도도 큰 문제가 될 수 있다. 현재 대부분의 펜컴퓨터가 386 SL을 CPU로 사용한다는 점에서 너무 복잡한 인식 알고리즘을 사용하여 펜컴퓨터에 탑재하는 경우, 글씨를 쓰고

나서 너무 많은 시간을 기다리게 되는 경우가 발생할 수 있기 때문이다. 따라서 인식률과 인식속도 사이에 적절한 타협점을 찾는 것이 중요하다.

상용화 측면에서 고성능 문자 인식기의 부재가 펜 컴퓨팅 분야의 발전 속도를 느리게 하는 한 원인이지만, 현재 문자 인식기의 인식수준은 정 자체인 경우 매우 우수한 성능을, 홀림체에 대해서는 고무적인 성능을 가진 것으로 평가되고 있으며, 계속적으로 그 성능이 향상될 전망이다. 따라서 이제부터는 펜 컴퓨팅 분야의 발전을 위해서는 고성능 문자인식 기술 뿐만 아니라, 무선통신 기술과 같은 관련 요소 기술개발도 병행해야 시너지효과를 기대할 수 있다.

또한 고객의 needs에 부합하는 다양한 응용 프로그램 개발도 지속적으로 되어야 할 것으로 생각된다.

## VI. 결 론

지금까지 펜용 운영체제로 각광받고 있는 펜 윈도우에 대해 살펴보고, 펜 윈도우 하에서 프로그램 개발 방안을 제시 하였다.

특히 개발된 인식 알고리즘을 이용하여 펜 윈도우 하에서 테스트 할 수 있도록 펜 윈도우용 인식기를 만드는 방법을 기술하였다. 이 방법을 이용하면 기존 인식 알고리즘의 Interface를 변환하여 펜 윈도우 환경하에서 테스트는 물론 응용 프로그램 개발도 가능할 수 있을 것이다.

더우기 마이크로소프트사의 영문 펜 윈도우와 자사의 한글 인식 알고리즘을 이용한 표준 한글 펜 윈도우 Spec.과 설계방법을 기술하였으므로 앞으로 제 3의 인식기 개발과 펜 윈도우용 응용 프로그램을 개발하고자 하는 개발자들의 많은 참고가 되길 바란다.

그리고 향후 전자잉크 정보의 공유를 위하여 Slate사 주도로 운영체제 컨소시엄과 응용 시스템을 만드는 회사들이 모여서 개발한 잉크 저장 방식과 전자 잉크 데이터의 교환을 위한 표준 형식인 JOT에 입각한 인식기 및 응용 프로그램 개발도 필요하다.

## 참 고 문 헌

1. Christopher Barr, "Pen PCs", PC Magazine, Nov. 10, 1992, pp. 175-185.
2. Microsoft, "Microsoft Windows for Pen Computing: Programmer's Reference", Microsoft Press, 1992.

3. Jon Udell, "Windows meets the pen", BYTE, Jun. 1992, pp. 159-168.
4. Eric Berman, "Recognizers and Windows for Pens", Microsoft, Mar. 6, 1992.
5. Ray Duncan, "An Introduction To Pen-Based Computing", PC Magazine, Jan. 14, 1992, pp. 415-418.
6. Ray Duncan, "Recognizing Handwritten Input in Pen Windows", PC Magazine, Mar. 17, 1992, pp. 431-435.
7. Ray Duncan, "Adding a Pen to Windows: A Look at The Pen Windows API", PCMagazine, Jan. 28, 1992, pp. 335-338.
8. Mark Gibbs, "Handwriting Recognition a Comprehensive Comparison", PEN, Mar./Apr. 1993, pp. 31-35.

### 이 현 주



1983 경북대학교 전자공학과 (학사)  
 1985 연세대학교 전자공학과 (석사)  
 1990 연세대학교 전자공학과 (박사)  
 1990 ~ 현재 금성사 중앙연구소 선임연구원  
 관심 분야: 영상처리, 문자인식, Computer Vision

### 조 문 증



1989 한양대학교 전기공학과 (학사)  
 1991 포항공과대학 전자전기공학과(석사)  
 1991 ~ 현재 금성사 중앙연구소 주임연구원  
 관심 분야: 신경회로망 및 그 응용, 온라인 문자인식, Computer Vision