

# 획 정보를 이용한 한글문자와 펜 제스처 인식 시스템의 설계 및 구현

오 준 택<sup>†</sup> · 김 옥 현<sup>††</sup>

## 요 약

본 논문은 멀티미디어 단말기, PDA 등에 적용할 목적으로 획 정보를 이용한 한글문자와 펜 제스처 인식 시스템을 설계 및 구현한다. 한글문자 인식은 다양한 필체 유형을 수용하기 위해 한글의 특성정보와 획 정보 등을 기반으로 구축한 한글데이터베이스를 이용하며 빠른 자소분리를 수행하기 위해서 획간의 위치정보를 이용한 순차적 자소분리와 자소를 이루는 획 수의 변경에 의한 백트래킹 자소분리를 이용한다. 펜 제스처 인식은 정의한 15가지 유형의 펜 제스처에 대해서 민감한 획 정보가 아닌 획 내의 교차수, 방향변화, 방향벡터, 방향코드, 개수, 위치관계, 획에 대한 시작점과 끝점간의 거리 비율정보 등을 분류특징으로 이용함으로써 강력한 인식과 빠른 처리속도를 가진다. 제안한 방법에 의해 구현한 인식 시스템은 실시간으로 수행하며 실험결과, 높은 인식률과 빠른 처리속도를 보였다.

## Design and Implementation for Korean Character and Pen-gesture Recognition System using Stroke Information

Juntaek Oh<sup>†</sup> · Wookhyun Kim<sup>††</sup>

## ABSTRACT

The purpose of this paper is a design and implementation for korean character and pen-gesture recognition system in multimedia terminal, PDA and etc, which demand both a fast process and a high recognition rate. To recognize writing-types which are written by various users, the korean character recognition system uses a database which is based on the characteristic information of korean and the stroke information which composes a phoneme, etc. In addition, it has a fast speed by the phoneme segmentation which uses the successive process or the backtracking process. The pen-gesture recognition system is performed by a matching process between the classification features extracted from an input pen-gesture and the classification features of 15 pen-gestures types defined in the gesture model. The classification feature is using the insensitive stroke information, i.e., the positional relation between two strokes, the crossing number, the direction transition, the direction vector, the number of direction code, and the distance ratio between starting and ending point in each stroke. In the experiment, we acquired a high recognition rate and a fast speed.

키워드 : 한글문자 인식(Korean Character Recognition), 펜 제스처 인식(Pen-gesture Recognition), 자소분리(Phoneme Segmentation)

### 1. 서 론

최근 다양한 멀티미디어 환경서비스를 위해서 입출력 방식을 보다 다양화하여 인간의 총체적 잠재성을 최적화하고 만족할 수 있도록 입출력 방식을 상호 통합하는 방향으로 바뀌어 가고 있다. 인간은 상황에 따라 다양한 정보들을 폭넓게 파악함으로써 세밀한 인식이 가능하며 이에 멀티모달은 가장 가까운 시도라고 할 수 있다. 멀티모달은 음성, 제스처, 표정, 필기, 펜 제스처 등 다양한 형태의 정보를 가장 자연스러운 기계로의 정보 입출력을 실현하고자 하는 시도이며 멀티미디어 시대에서 인간과 기계간의 정보 교환을 원활하게 하는 새로운 인터페이스 기술이다. 멀티미디어 단말기술은 이러한 멀티모드를 이용하여 사용자의 의도를 충분히 전달하고 기계측

에서는 이들 정보를 통합하여 이해하게 함으로써 전달하고자 하는 정보를 완전히 획득할 수 있는 휴먼 인터페이스 핵심기술의 개발을 목적으로 한다[1, 2]. 특히 멀티미디어 단말기술 중에 문자와 펜 제스처 인식은 PDA, 멀티미디어 단말기 등과 같이 사용자의 다양한 요구사항을 빠르게 처리하는 지능형 단말기의 구현을 위해 필수적인 연구과제이다. 그러나 기존의 한글문자 및 펜 제스처 인식 시스템은 단일 시스템으로써 높은 인식률만을 위해 복잡한 알고리즘과 처리과정을 요구하므로 응용분야로의 적용은 미흡하거나 어려운 실정이다.

### 2. 관련 연구

본 장에서는 흐름에 따른 한글 인식방법에 대해 살펴보고 한글 및 펜 제스처 인식에 대한 기존연구와 일반적인 처리과정에 대해 알아본다.

<sup>†</sup> 준 회 원 : 영남대학교 대학원 컴퓨터공학과

<sup>††</sup> 정 회 원 : 영남대학교 전자정보공학부 교수

논문접수 : 2002년 7월 15일, 심사완료 : 2002년 10월 22일

### 2.1 홀림에 따른 한글 인식 방법

홀려 쓴 필기 또는 무제약 필기의 인식은 동일한 필기자에 의한 자소나 문자라도 필기 중에 변형이 가미될 수 있어 많은 어려움이 따른다. 이러한 문제점을 해결하기 위해 자소나 문자별로 분리나 비분리를 전제로 한 여러 가지 방법을 사용한다.

분리는 외부분리와 내부분리로 분류할 수 있다[16, 17]. 외부분리는 시공간적인 정보를 이용하거나 사용자의 신호에 의해 인식 이전에 분리를 시도하는 방법이다. 반면에 내부분리는 가능한 자소나 문자의 경계를 인식 단계에서 유지하고 분리와 인식 결과를 동시에 얻는 방법으로 무제약 필기에서 자소 및 문자간 홀림이 많이 가미될수록 복잡한 알고리즘과 많은 처리시간을 요구한다[4, 16, 17].

문자를 인식하는 기본 단위는 획, 획 세그먼트, 자소, 문자 단위 인식으로 분류할 수 있다[16, 17]. 획 단위 인식은 획의 방향변화, 회전 정도, 시·종점의 방향 등의 특징을 기반으로 획을 인식한 후 자소와 문자를 인식하는 방법으로 빠른 처리속도와 높은 인식률을 얻을 수 있지만 기본 획의 규정이 모호하고 자소 간의 인식이 어렵다는 단점이 있다. 획 세그먼트 단위 인식은 문자를 구성하는 직선 획들로부터 획 세그먼트를 분리하기 쉬운 경우 사용한다. 그러나 홀림체 한글에서 일정한 획 세그먼트를 추출하는 것은 어려운 일이다. 자소 단위 인식은 바로 자소를 인식하므로 문자 인식에 보다 쉽게 접근할 수 있으나 홀려 쓴 경우에 자소분리를 위한 알고리즘의 구현이 어렵다는 단점이 있다. 문자 단위 인식은 입력문자를 데이터베이스에 저장된 문자들과 일일이 비교하는 방법으로 가장 간단하지만 입력문자의 변형이 크면 인식속도가 느리고 한정된 수의 문자만 인식 가능하다. 그 외에도 문자구분방식에 따라 제한된 격자 내에 필기하는 격자구분필기방법, 글자 사이에 공백을 두어 필기하는 공백구분필기, 한 글자 필기 후 어느 정도의 시간을 두고 필기하는 시간구분필기 등 많은 외부 분리기법들이 있다. 인식 알고리즘에 의존한다는 측면에서 외부분리는 내부분리에 비해 더 많은 대화성과 빠른 처리속도를 가지며 또한 인식부에서 처리할 자료를 간단하게 하는 장점이 있어 현재 외부분리를 이용하는 인식 시스템들은 좋은 결과를 보이고 있다[18].

### 2.2 기존 연구

한글 인식의 경우, 심동규[3]는 DP(dynamic programming) 정합에 근거한 자소 분할과 문자 인식 방법을 제안하였다. 그러나 모든 자소들의 정합을 기반으로 인식을 수행하므로 잘못된 자소 정합의 경우 오인식을 유발하며 분류 및 인식 시간의 문제 등을 가진다. 권오성[4]이 제안한 선분정합방법은 외부와 내부 획 분할에 의해 이중적인 형태로 진행하며 인식은 선분정합의 결과를 탐색하여 처리한다. 그러나 자소 내 홀림이 발생하는 경우 정합과정에서 오인식을 유발한다. 신경망을 이용하는 방법[5, 6]은 표준 패턴에 대한 학습에 의해서 변형된 패턴에 대해서도 원형에 가장 가까운 결과를 얻을 수 있어 문자 인식뿐만 아니라 다른 패턴 인식분야에서

도 최선책으로 손꼽히고 있다. 그러나 끊어진 획이나 잡음이 추가되는 경우 오인식을 유발하며 자소 간에 연결 획이 존재할 때 자소를 분리하기에 어려움을 가진다. 또한 느린 수렴 속도, 부적절한 임계치 설정, 중복 인식 등으로 인하여 많은 처리시간을 요구하는 단점이 있다. 신봉기[7, 8]은 은닉 마르코프 모델(HMM)을 이용한 통계적 방법을 제안하였다. 그러나 확률적 모델링이 가지는 자체적인 문제점과 처리 속도 저하 등과 같은 문제점을 지닌다. 김태균[9]은 복잡한 획 해석과 위치관계를 기반으로 자모 인식테이블과 문자조합규칙을 이용함으로써 연속필기문자에 대해서도 인식이 가능하다. 그러나 획 수에 의존적이며 복잡한 획 해석과 오인식이 유발되는 경우에 복잡한 재처리과정을 필요로 한다. 이러한 기존 한글 인식방법들은 복잡한 획 해석에 의한 인식과정과 오인식 및 미인식이 유발할 경우에 불필요한 재처리과정을 요구하므로 응용 시스템으로의 적용은 어려운 실정이다.

펜 제스처 인식은 하나 이상의 획들로 구성된 패턴을 대상으로 빠르게 인식과정을 수행해야 하며 텍스트 명령보다 사용자들에게 기억하기 쉽고 편리해야 한다. 최근 PDA(personal digital assistant)나 멀티미디어 단말기 등에서 효율적으로 이용하지만 사용자들은 좀 더 많은 펜 제스처와 기능을 요구하기에 현재 많은 연구가 진행 중이다[11]. 기존의 펜 제스처 인식 시스템은 텍스트, 테이블, 리스트, 그림 등 다양한 대상들에 대해 그래픽 편집기로서의 사용자 인터페이스를 강조하고 있다[12, 13]. J. S. Lipscomb[13]는 그래픽 상태로 출력된 텍스트를 대상으로 펜 제스처 인식 및 펜 제스처에 따른 편집 시스템을 설계하였으며 M. J. Fonseca[14]는 퍼지이론을 이용하여 하나 이상의 획들로 구성된 기하학적 형태의 펜 제스처 인식을 수행하였다. 그러나 이러한 기존 인식 방법들은 펜 제스처 유형이나 기능이 제한적이며 단지 펜 제스처에 따른 그래픽 영역에서의 편집기능을 수행하므로 실질적인 텍스트 편집기능을 요구하는 응용 시스템으로의 적용에는 어려움이 있다.

### 2.3 일반적인 인식 시스템의 처리과정

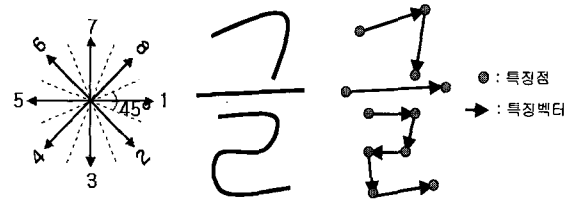
일반적인 인식 시스템은 크게 전처리과정, 특징추출과정, 인식과정으로 구성된다[15, 16]. 전처리과정은 정확도, 양자화, 손동작의 오류, 펜 접촉의 부정확한 측정 등에 의해 발생하는 잡음을 제거하기 위한 목적으로 크기 정규화, 난폭점 제거, 고립점 제거, 평활화, 재표본화, 여과, 장식선 및 고리 제거 등[3, 15, 16]을 이용한다. 특징추출과정은 개선된 입력 좌표 데이터열로부터 인식을 위한 정보를 추출하는 과정으로써 특징점을 기반으로 진행방향정보인 특징벡터와 자소 간이나 자소 내의 위치정보인 가상벡터 등을 추출한다[15, 16]. 본 논문에서는 획의 시작점, 끝점, 굴곡점을 특징점으로 정의하며 굴곡점은 인접한 점간의 각도와 전 참조점(특징점)간의 각도 변화가 임계치( $40^\circ$ )보다 큰 점으로 정의하였다. (그림 1)은 문자 '간'의 각 자소에 대해서 추출한 특징점을 보여준다. 가상벡터는 입력문자의 구조적인 특성정보로써 Tail-Head Vector, Head-Head Vector, Last Tail-First Head Vector로 구성된다[15, 16]. Tail-Head Vector은

이전 획의 마지막 점과 현재 획의 시작점간의 방향정보이며 Head-Head Vector는 이전 획의 시작점과 현재 획의 시작점간의 방향정보이다. 그리고 Last Tail-First Head Vector는 전체 획에서 마지막 획의 끝점과 시작 획의 시작점간의 방향정보를 나타낸다. (그림 2)(a)는 특징벡터와 가상벡터를 추출하기 위한 체인코드를 보여주며 (그림 2)(b)와 (그림 2)(c)는 입력문자 '글'과 입력문자에 대해서 추출한 특징벡터를 보여준다. (그림 3)는 입력문자 '글'에 대해서 추출한 가상벡터들을 보여준다. 인식 과정은 특징정보를 이용하여 자소 및 문자를 인식하는 과정으로 현재 많이 이용하는 인식 방법으로는 전체적인 특성에 의한 결정론적 방법, 구조적 특성에 의한 구문론적 방법, 신경망을 이용한 방법들이 있다[15-18]. 결정론적 방법으로는 형틀 정합(template matching), 변환(transformation), 수열 확장(series expansion) 등이 있으며 패턴을 구분하기 위한 결정함수로서 역전파 신경망을 많이 이용한다. 그러나 패턴의 종류가 많은 경우 특성벡터의 수가 커지고 결정함수의 복잡도가 증가하여 비효율적이며 계산상으로 불가능할 수 있다. 이러한 문제점을 해결하기 위해서 복잡한 패턴일 경우 계층적으로 간단한 부분패턴과 더 이상 분류할 수 없는 기본패턴으로 분류하고 이들 기본 패턴에 대하여 결정론적 방법으로 해결하는 구문론적 방법 및 획 정합 방법 등을 많이 이용한다[15, 19]. 또한 신경망을 이용하는 방법[5, 6]은 표준 패턴을 학습하고 변형된 패턴에 대해 원형에 가장 가까운 출력을 내므로 문자 인식 분야에서 상당한 성과를 올리고 있다.

본 논문은 멀티미디어 단말기나 PDA 등과 같이 빠른 처리와 높은 인식률을 요구하는 응용 시스템으로의 적용을 목적으로 획 정보를 이용한 한글문자와 펜 제스처 인식 시스템을 설계 및 구현한다. 기존의 한글문자 인식 방법들은 오인식의 경우 불필요한 재처리과정을 요구하지만 본 논문에서는 획간의 위치정보에 의한 순차적 자소분리와 획 수의 변경에 의한 백트래킹 자소분리를 이용함으로써 불필요한 재처리과정을 제거하였다. 또한 다양한 자소 정보를 기반으로 구축한 한글데이터베이스를 이용함으로써 다양한 필체 유형에 강건하다. 펜 제스처 인식은 15가지의 펜 제스처들에 대해 민감한 정보가 아닌 획 내의 교차수, 방향벡터, 방향코드의 수, 획 내의 방향변화, 위치관계, 시작점과 끝점간의 거리 비율정보를 분류특징으로 정의한 후 이용함으로써 강건한 분류 및 인식이 가능하다. 인식 시스템은 텍스트의 입력이나 입력된 텍스트를 대상으로 펜 제스처나 텍스트 모듈에서의 직접적인 편집기능에 의해 실시간으로 수행한다.



(그림 1) '간'의 자소별 특징점 추출

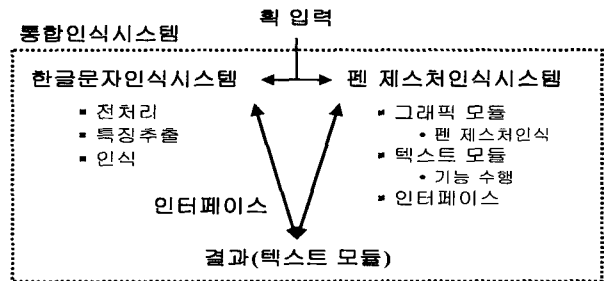


(a) 8방향 체인코드 (b) 입력문자 '글' (c) 특징벡터 추출  
(그림 2) 8방향 체인코드와 '글'에 대한 특징벡터 추출



(a) Tail-Head Vector (b) Head-Head Vector (c) Last Tail-First Head Vector

(그림 3) '글'에 대한 가상벡터 추출



(그림 4) 통합 인식시스템 구성도

### 3. 한글문자와 펜 제스처 인식 시스템

제안하는 인식 시스템은 (그림 4)와 같이 한글문자 인식 모듈과 펜 제스처 인식모듈로 구성되며 인터페이스에 의해 각 모듈간의 데이터를 갱신 및 공유한다. 본 장에서는 각 모듈별 인식 시스템에 대해 살펴본다.

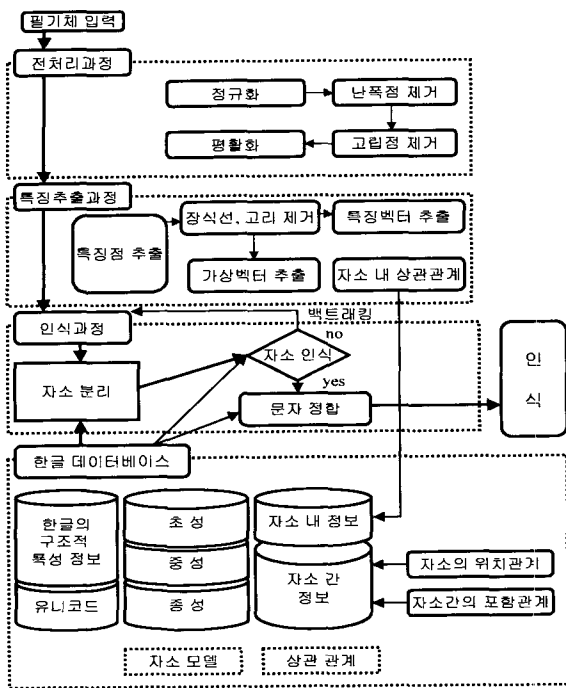
#### 3.1 한글문자 인식 시스템

한글문자 인식 시스템의 전체 처리과정은 (그림 5)와 같이 크게 전처리과정, 특징추출과정, 인식과정으로 구성되며 한글데이터베이스를 이용한다.

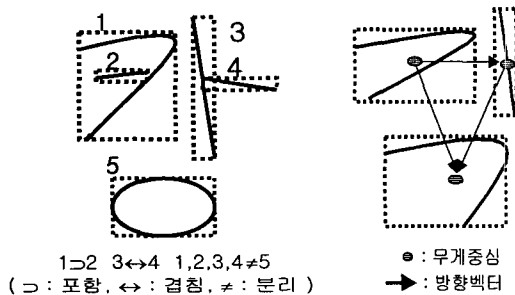
##### 3.1.1 전처리 및 특징추출과정

전처리과정은 크기 정규화, 난표점 제거, 고립점 제거, 평활화[15, 16]로 구성되며 정의된 사각형안에서 문자를 입력 받음으로써 문자간 홀림은 배제하였다.

특징추출과정은 개선된 획들의 좌표 데이터 열로부터 자소 인식을 위한 정보로서 특징벡터, 가상벡터[15, 16], 획간의 위치관계[4, 15, 16] 등을 추출한다. 입력 획은 필기체의 특성상 정형화되지 않은 길이와 각도를 가지며 여러 가지 형태의 홀림이 가미될 수 있다. 그러므로 입력 획의 좌표 데이터열



(그림 5) 한글문자 인식 시스템의 전체 처리과정



(a) 포함관계정보 추출 (b) 위치관계정보 추출

(그림 6) 획간의 위치관계 추출

로부터 불필요한 정보의 양을 줄이기 위해 특징점을 인식의 기본단위로 이용한다[15, 16]. 그러나 필기 속도나 각도 변화에 따라 입력 획의 특징부분에 여러 개의 특징점이 존재할 수 있으므로 고리, 장식선 제거 및 인접한 특징점 제거과정에 의해 주어진 거리 임계치 이내에 군집된 불필요한 특징점들을 제거한다. 특징벡터와 가상벡터는 한글데이터베이스와의 정합을 위한 정보로서 특징벡터는 입력 획에 대한 순서 및 방향정보이며 가상벡터는 자소 내 획 간의 위치관계나 자소 간 위치관계에 대한 정보이다[15, 16]. 그러나 자소 인식시에 후보 자소들이 많이 발생하지 않으면 빠른 처리를 위해 가상벡터는 Tail-Head Vector만을 이용한다. 또한 자소분리와 인식을 위해서 획간의 위치정보인 포함관계와 위치관계를 추출한다. 포함관계는 현재 획을 포함하는 최소의 점 사각형과 이웃하는 획들의 최소의 점 사각형간의 인접한 정도를 나타내는 정보로서 포함, 겹침, 분리로 나누어진다. (그림 6)(a)는 입력문자 '강'에 대해서 포함관계를 추출하는 예를 보여준다.

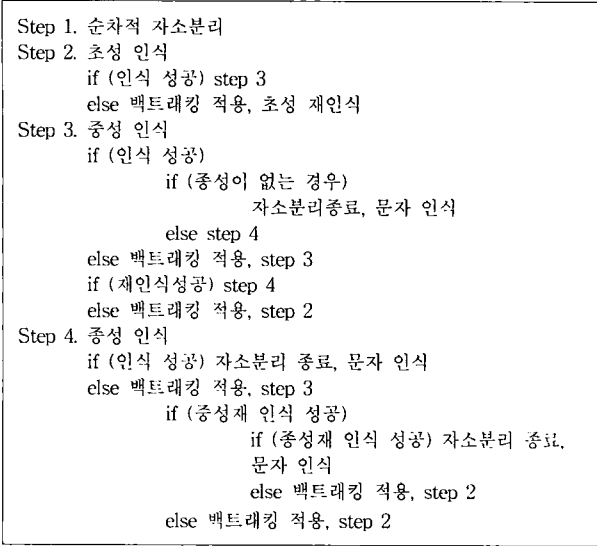
점선사각형은 획을 포함하는 최소의점사각형이며 번호는 점선사각형안에 있는 획 번호이다. 즉, 점선사각형 1은 획 'ㄱ'에 대한 전체 크기를 나타낸다. 점선사각형 2는 점선사각형 1에 포함되며 점선사각형 1, 3, 5는 서로 분리되어있으며 점선사각형 3과 4는 겹쳐있음을 알 수 있다. 위치관계는 현재 획에 대한 무게중심과 모든 입력 획의 무게중심간의 방향정보로서 자소 내, 자소 간 위치관계정보로 이용한다. (그림 6)(b)는 입력문자 '기'의 위치관계를 추출하는 예로서 추출된 위치관계정보는 문자를 구성하는 각 자소들의 무게 중심들간의 방향정보이다. 이러한 획간의 위치관계는 순차적 자소분리와 자소 및 문자 인식을 위한 정보로 이용한다.

### 3.1.2 자소분리 및 인식

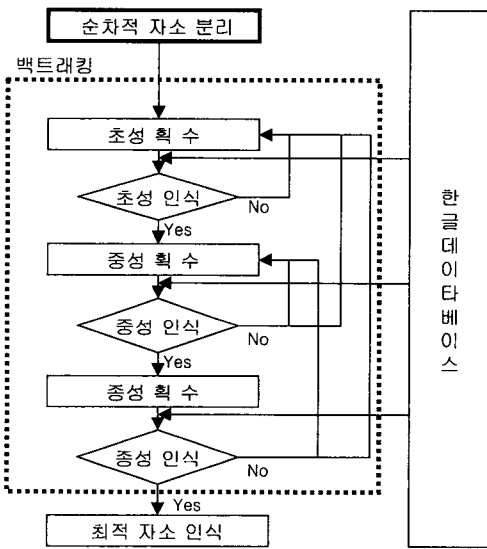
자소분리는 자소별 인식을 위해 반드시 필요한 과정으로, 본 논문에서는 자소분리와 인식을 병행하여 처리한다. 자소 인식은 자소분리에 의해 추출한 각 자소별 획 정보와 한글 데이터베이스에 저장된 자소별 획 정보간의 정합에 의해 자소를 인식하며 인식된 각 자소들의 정보를 유니 코드 2.0의 부코드로 변경하여 문자를 생성한다.

자소분리는 획간의 위치관계를 이용한 순차적 자소분리와 한글 데이터베이스와의 정합에 의해 각 자소들을 구성하는 획 수를 변경함으로써 인식을 수행하는 백트래킹 자소분리를 이용한다. 순차적 자소분리는 획간의 포함관계가 분리일 경우 다른 자소의 획이라고 인식하며 겹침이나 포함일 경우 같은 자소 내의 획으로 인식한다. 그러나 자소 내 획간의 포함관계가 분리일 경우 순차적 자소분리는 정확한 자소분리 능력이 미흡하므로 오인식이나 미인식이 발생할 경우 백트래킹 자소분리를 수행한다. 기존의 백트래킹방법은 모든 과정을 재처리해야하는 단점을 가진다. 그러나 제안된 백트래킹 자소분리는 기존에 추출한 획 정보를 기반으로 입력문자가 가지는 전체 획의 개수를 각 자소별 인식 여부에 따라 최적의 획 개수로 분리하여 인식을 수행한다. (그림 7)은 이러한 백트래킹 자소분리에 대한 알고리즘을 나타내며 (그림 8)은 자소분리와 인식이 병행하여 처리되는 과정을 보여준다. 본 논문은 입력문자나 필체의 형태 정보에 따라 차별적으로 자소분리를 수행함으로써 빠른 처리속도로 가진다.

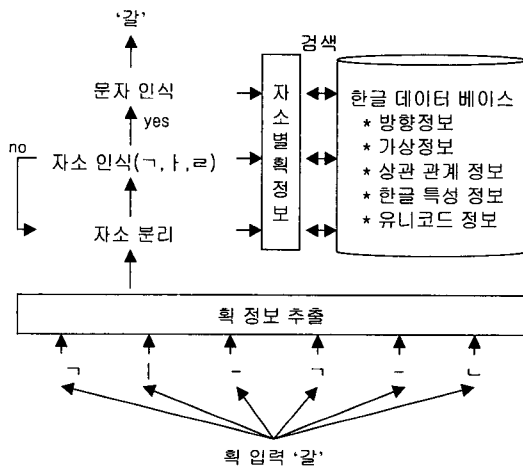
자소 인식은 한글데이터베이스와의 정합과정으로 이루어지며 한글데이터베이스는 한글의 구조적 유형정보와 각 자소에 대한 다양한 사용자들의 획 정보 등을 지닌다. 한글의 구조적 유형정보는 중성 인식에서 사용되는 자소 간 결합 정보, 모음 정보, 획간의 위치관계, 유형에 따른 자소 분류 정보 등을 포함한다. 또한 자소별 인식을 위한 자소 모델은 각 자소에 대해서 일반적으로 사용하는 필체 정보와 사용자 정의에 의한 다양한 필체 정보 등을 지닌다. 하지만 심한 곡선이나 자소 간 홀림이 가미된 자소에 대해서는 정확한 정보를 추출하기 힘들기 때문에 직선, 완만한 곡선 정보, 다양한 방향의 필체 유형만을 이용한다. (그림 9)은 입력문자 '갈'에 대해 계층적으로 인식하는 처리과정을 보여준다.



(그림 7) 백트래킹 자소분리 알고리즘



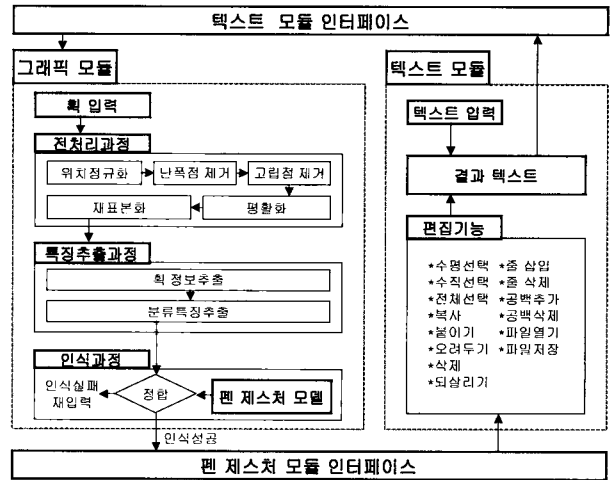
(그림 8) 자소분리와 인식 처리도



(그림 9) 계층적 인식 과정

### 3.2 펜 제스처 인식 시스템

펜 제스처 인식 시스템은 펜 제스처의 인식을 위한 그래픽 모듈, 펜 제스처에 의한 기능부여나 텍스트 모드에서의 기능을 수행하기 위한 텍스트 모듈, 모듈간의 데이터 공유 및 갱신을 위한 인터페이스로 구성되며 인식 시스템의 전체 처리 과정은 (그림 10)과 같다.



(그림 10) 펜 제스처 인식 시스템 구성도

#### 3.2.1 그래픽 모듈

그래픽 모드에서 펜 제스처를 입력 및 인식하는 부분으로, 한글문자 인식 시스템이나 사용자의 직접적인 작업에 의해 입력받은 텍스트를 그래픽 모드로 출력한 후 텍스트를 대상으로 펜 제스처를 인식한다.

전처리과정으로는 크기 정규화, 난폭점 제거, 고립점 제거, 평활화, 재표본화를 이용하며[15, 16] 재표본화는 DDA(dot density algorithm)를 이용하여 일정한 간격거리를 가지는 점들로 정규화한다.

특징추출과정에서는 특징벡터와 가상벡터를 이용하여 펜 제스처를 인식하기 위한 분류특징을 추출한다. 본 논문에서는 (그림 11)와 같이 일반적으로 많이 이용하는 펜 제스처 유형과 기능을 정의한 후, 펜 제스처를 분류하기 위한 특징으로 획 내의 교차수, 방향벡터, 방향코드의 수, 획 내의 방향변화, 위치관계, 시작점과 끝점간의 거리에 대한 비율정보를 이용한다. 획 내의 교차수는 두 개의 특징점을 잇는 대상 직선과 이웃하는 특징점을 잇는 직선간의 교차수에 대한 정보로서 (그림 12)은 오려두기와 삭제 기능을 가진 펜 제스처에 대해서 추출한 교차점을 보여준다. 오려두기 기능을 가진 펜 제스처는 획 내에 교차점이 한 개가 존재하지만 삭제 기능을 가진 펜 제스처는 획 내에 교차점이 존재하지 않는다. 방향벡터는 획의 진행방향정보로서 (그림 13)는 삭제와 오려두기 기능을 가지는 펜 제스처에 대해서 방향벡터를 추출하는 예를 보여준다. 방향코드의 수는 입력 획에 얼마나 많은 굴곡이 가미되었는지를 파악하기 위한 정보로 이용한다. 획 내의 방향변화정보는 진행방향의

변화정보로써 시계 방향, 반시계 방향, 무 방향으로 나누어진다. 시계방향은 획 내의 진행방향이 반시계 방향보다 시계 방향이 많은 경우이고 반시계 방향은 진행방향이 시계 방향보다 반시계 방향이 많은 경우이다. 그리고 무 방향은 시계 방향변화와 반시계 방향변화가 같은 경우이거나 전혀 방향변화가 없는 경우이다. (그림 14)는 되살리기와 삭제 기능을 가진 펜 제스처로써 획의 진행방향이 동등하거나 획 내에 아무런 방향변화가 없으므로 무 방향정보를 가진다. 위치관계는 단일 획이나 획 간의 방향정보이다. 단일 획의 경우 Tail-Head Vector와 Head-Head Vector 정보는 생성되지 않고 Last Tail-First Head Vector만 생성된다. (그림 15)(a)처럼 붙이기 기능을 가진 펜 제스처는 단일 획으로만 생성되므로 Last Tail-First Head Vector만 추출되며 (그림 15)(b)의 삭제 기능을 가진 펜 제스처는 두 개의 단일 획으로 구성되므로 모든 가상벡터들이 추출된다. 하지만 분류 및 인식과정에서 후보 펜 제스처들이 많이 추출되지 않거나 뚜렷한 정보를 가진 펜 제스처라면 빠른 처리를 위해 Tail-Head Vector 정보만을 이용하거나 위치관계정보를 이용하지 않는다. 시작점과 끝점간의 거리비율정보는 추가적인 분류특징으로 펜 제스처인 'c'와 'o'를 분류하기 위한 정보로 이용한다. 입력 획에 대해 최소외접사각형을 구한 후 최소외접사각형 높이의 1/3보다 획의 시작점과 끝점간의 높이가 작으면 이를 펜 제스처 'o'로 인식하고 높이가 크면 펜 제스처 'c'로 인식한다. (그림 16)은 복사 기능과 함께 줄 삭제 기능을 가지고 있는 펜 제스처 'c'와 파일열

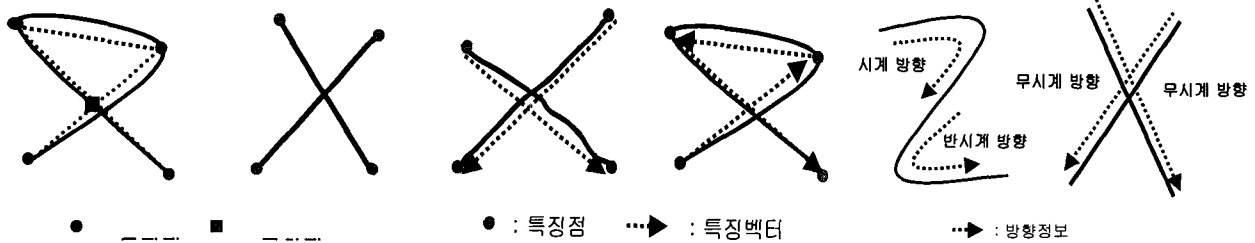
기 기능을 가진 펜 제스처 'o'에 대한 예를 보여준다.

인식과정은 펜 제스처들의 분류특징들을 기반으로 구축한 펜 제스처모델과 입력 펜 제스처에 대해서 추출한 분류특징들간의 정합과정에 의해 수행하며 사용자 특성에 따라 정의된 펜 제스처의 분류특징들을 펜 제스처 모델에 추가함으로써 다양한 형태의 펜 제스처 분류 및 인식이 가능하다.

복사, 오려두기, 삭제 기능은 선택 기능을 가진 펜 제스처에 의해 처리할 텍스트를 선택한 후 수행하고 그 외의 기능을 가진 펜 제스처는 텍스트를 선택하는 과정없이 수행한다. 선택 기능을 가진 펜 제스처는 수직과 수평 펜 제스처를 사용하거나 전체 선택 펜 제스처를 사용함으로써 텍스트의 다양한 범위 선택이 가능하다. 그래픽 모듈에서의 인식과정은 펜 제스처 인식과 더불어 처리할 텍스트의 범위와 펜 제스처에 따른 위치정보 등을 추출하며 인터페이스에 의해 텍스트 모듈로 전달한다. (그림 17)은 입력되는 펜 제스처를 계층적으로 인식하는 과정과 추출하는 정보들을 보여준다.

|      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|
| 수평선택 | 수직선택 | 전체선택 | 복사   | 붙이기  | 오려두기 | 삭제   |
| —    |      | +    | C    | V    | L    | X    |
| 되살리기 | 줄삽입  | 줄삭제  | 파일열기 | 파일저장 | 공백삽입 | 공백삭제 |
| Z    | □    | C    | O    | S    | >    | □ □  |

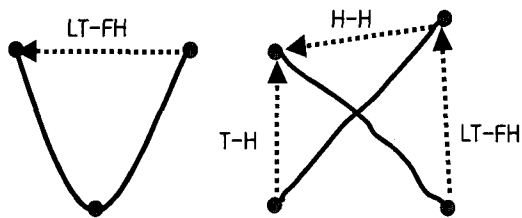
(그림 11) 펜 제스처 유형 및 기능



(그림 12) 교차점 추출

(그림 13) 방향벡터 추출

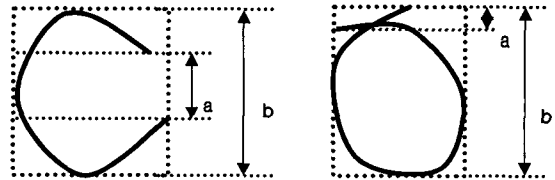
(그림 14) 방향변화 추출



(a) 붙이기

(b) 삭제

(그림 15) 위치관계 추출



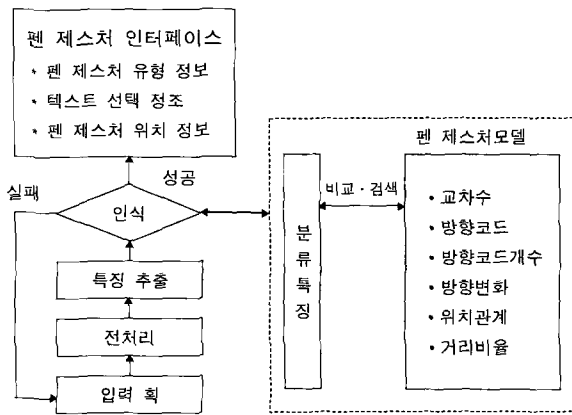
a : 시작점과 끝점간의 높이    b : 입력 획의 높이

if  $\frac{a}{b} \leq \frac{1}{3}$  then pen gesture 'o'

(a) 복사 및 줄 삭제

(b) 파일열기

(그림 16) 거리비율정보 추출



(그림 17) 펜 제스처 인식 처리도

3.2.2 텍스트 모듈

텍스트 모듈에서는 텍스트 입력, 편집기능 수행, 결과 텍스트부분으로 구성된다. 텍스트 입력은 키보드나 문자 인식 시스템을 이용하여 텍스트를 입력하는 부분이며 기능 수행은 펜 제스처나 텍스트 입력 창에서의 인터페이스에 의해 선택된 텍스트들을 편집 및 수정하는 부분이다. 결과 텍스트는 기능 수행에 의해 편집되거나 수정된 결과를 보여주는 부분이다. 각 모듈에서 출력된 텍스트는 인터페이스에 의해 실시간으로 갱신된다.

3.2.3 인터페이스

실시간으로 모듈간의 텍스트를 공유 및 갱신하기 위한 부분으로써 텍스트 모듈 인터페이스와 그래픽 모듈 인터페이스로 구성된다. 텍스트 모듈 인터페이스는 텍스트가 입력되거나 수정 및 편집되었을 때 새로운 결과 텍스트부분의 값을 부여하는 기능을 수행한다. 그래픽 모듈 인터페이스는 펜 제스처의 인식에 따른 유형정보, 펜 제스처에 의해 선택된 텍스트의 범위 정보, 선택된 텍스트에 대해서 수정 및 편집을 수행하기 위한 펜 제스처의 위치정보 등을 텍스트 모듈에 알려줌으로써 펜 제스처에 따른 정보를 전달하기 위한 기능을 수행한다.

4. 실험 및 결과

개발 환경은 Pentium 450Mhz에서 윈도우 환경 하에 Java 언어를 사용하였고 입력 장치로는 해상도 2540LPI인 Accat III 5"× 5"테블릿을 사용하였다. 외부 분리기법을 사용하여 자소 내의 흘림은 어느 정도 허용하지만 문자 간, 자소 간의 흘림은 배제하였으며 입력문자나 펜 제스처에 대한 회전은 가미도 배제하였다. 실험은 테블릿에 훈련을 받은 사용자의 한글문자와 펜 제스처를 대상으로 하였다. 인식을 위해 이용되는 한글데이터베이스는 총 20명의 필기자들에게 대한 자소 유형정보를 추출하여 자소 당 평균 3개의 필체유형정보로 구축하였고 펜 제스처모델은 각 펜 제스처

에 대해 6개의 분류특징만으로 구축하였다.

4.1 한글문자 인식

한글문자 인식의 경우 한글데이터베이스 구축에 참여하지 않은 사용자를 대상으로 한 필기한글 600문자(실험 1)와 한글데이터베이스 구축에 참여한 사용자와 참여하지 않은 사용자를 대상으로 한 필기한글 900문자(실험 2)를 대상으로 자소 및 문자 인식률을 평가하였다. 또한 인식률 성능평가를 위해 사용된 필기한글 중 600문자를 대상으로 처리속도를 평가하였다.

<표 1>은 실험 1, 2에 대해 자소 및 문자 인식률을 보여준다. 실험 1에서는 1500개의 자소들과 600개의 문자들을 대상으로 96.67%의 자소 인식률과 95.3%의 문자 인식률을 얻었고 실험 2에서는 2400개의 자소들과 900개의 문자들을 대상으로 98.7%의 자소 인식률과 98.3%의 문자 인식률을 얻었다. 문자 인식률은 인식된 자소들을 기반으로 구성되므로 자소 인식률보다 낮은 인식률을 보였다. 이것은 자소 간 위치관계정보의 잘못된 추출로 인하여 자소 결합시에 오인식된 문자를 유발하였기 때문이다. 또한 한글데이터베이스 구축에 참여한 사용자들에 대한 인식률이 참여하지 않은 사용자들의 인식률보다 높게 나와 사용자의 필체 특성에 어느 정도 의존적이었다. <표 2>는 자소분리에 따른 처리속도를 보여준다. 순차적 자소분리와 백트래킹 자소분리에 의해 인식이 가능한 문자들에 대해서 각각 3.01msec와 19.8msec의 평균처리속도를 보였으며 입력된 600문자를 대상으로 13msec의 빠른 평균처리속도를 보였다. <표 3>과 <표 4>는 기존의 인식 시스템과 제안한 인식 시스템간의 문자 인식률과 처리속도에 대한 성능 비교를 보여준다. 제안한 인식 시스템은 97.13%의 높은 평균문자 인식률과 0.013sec의 빠른 처리속도를 보여 응용 시스템으로의 적용이 유용함을 확인하였다. 그러나 한글데이터베이스 구축 시에 사용할 자소 당 필체 정보가 적을수록 낮은 인식률을 보여 필체 정보의 개수에 의존적이었음을 알 수 있었다.

<표 1> 자소와 문자 인식률

| 구분   | 총 수   | 정인식  | 오인식  | 미인식 | 인식률 |        |
|------|-------|------|------|-----|-----|--------|
| 실험 1 | 자소 인식 | 1500 | 1450 | 35  | 15  | 96.67% |
|      | 문자 인식 | 600  | 572  | 26  | 2   | 95.33% |
| 실험 2 | 자소 인식 | 2400 | 2370 | 25  | 5   | 98.75% |
|      | 문자 인식 | 900  | 885  | 12  | 3   | 98.33% |

<표 2> 자소분리에 따른 처리 속도

| 구분        | 실험문자 총 개수 | 처리개수 | 평균처리 시간  | 전체평균 처리시간 |
|-----------|-----------|------|----------|-----------|
| 순차적 자소분리  | 600       | 248  | 3.01msec | 13msec    |
| 백트래킹 자소분리 |           | 352  | 19.8msec |           |

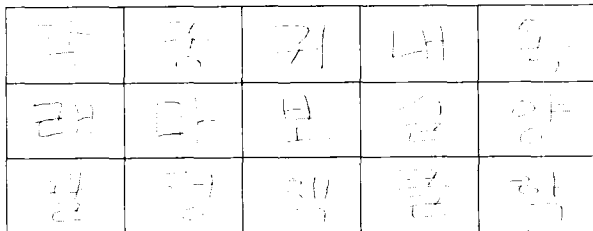
<표 3> 문자 인식률에 의한 성능비교

|                      | 문자 인식률  |
|----------------------|---------|
| DP 정합을 이용한 인식[3]     | 95 %    |
| 신경망을 이용한 방법[5]       | 94.8 %  |
| 획 해석에 의한 방법[9]       | 94 %    |
| 퍼지 결정 트리를 이용한 방법[10] | 91 %    |
| 제안한 시스템              | 97.13 % |

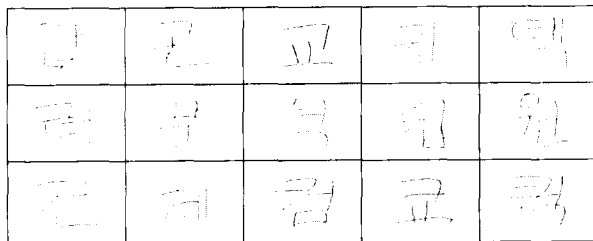
<표 4> 처리속도에 의한 성능비교

|                      | 처리속도(초/문자) |
|----------------------|------------|
| DP 정합을 이용한 인식[3]     | 0.16       |
| 퍼지 결정 트리를 이용한 방법[10] | 0.33       |
| 제안한 시스템              | 0.013      |

(그림 18)은 순차적 자소분리와 백트래킹 자소분리를 사용함으로써 인식이 가능한 입력문자들을 보여주며 (그림 19)은 본 논문에서 배제한 자소 간 홀림이나 회전이 가미되어 미인식된 입력문자들을 보여준다. 오인식 및 미인식을 유발하는 경우는 사용자들의 각기 다른 필체특성으로 인하여 한글데이터베이스에서 검색에 실패한 문자들이었으며 하나의 자소에 대해 평균 3개의 정보로 구축된 한글데이터베이스로는 독특한 필체정보의 수용이 어려웠다. 즉, 사용자들의 필체에 따라 한글데이터베이스에 없는 특징벡터나 위치관계정보를 추출하는 경우에 가장 많은 오인식과 미인식을 유발하였다. 또한 전처리과정에서 미흡하게 설정한 거리 임계치로



(a) 순차적 자소분리에 의해서 인식 가능한 문자



(b) 백트래킹 자소분리에 의해서 인식 가능한 문자

(그림 18) 인식 가능한 입력문자



(그림 19) 미인식된 입력문자

인하여 오인식을 유발하였다. (그림 20)은 오인식 및 미인식을 유발하는 입력문자들이다. (그림 20)(a)는 전처리과정의 고립점 제거에 의해 모음 'ㅏ'를 'ㅑ'로 오인식하였고 (그림 20)(b)는 전처리과정의 크기 정규화로 인하여 초성 'ㅂ'에 대해 잘못된 위치관계정보를 추출하였기 때문에 미인식을 유발하였다. (그림 20)(c)은 초성 'ㄹ'을 구성하는 획에 대해서 잘못된 진행방향에 따른 특징점의 추출에 의해 특징벡터가 한글데이터베이스에서 검색되지 않아 미인식을 유발한 경우이며 (그림 20)(d)와 (그림 20)(e)는 자소를 구성하는 획의 위치관계에 의해 미인식을 유발하였다.

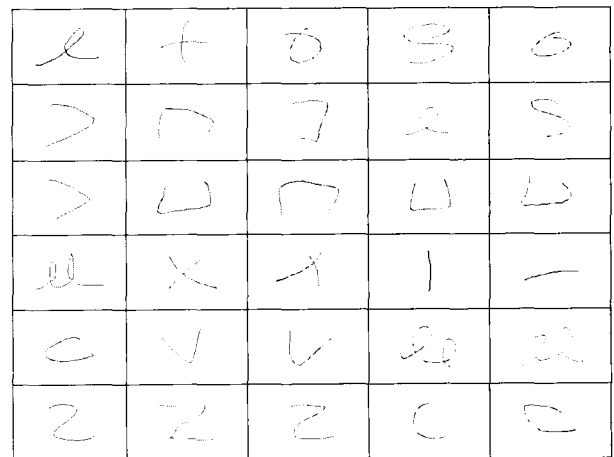


(a) '거' (b) '비' (c) '릿' (d) '변' (e) '정'

(그림 20) 오인식 및 미인식된 입력문자

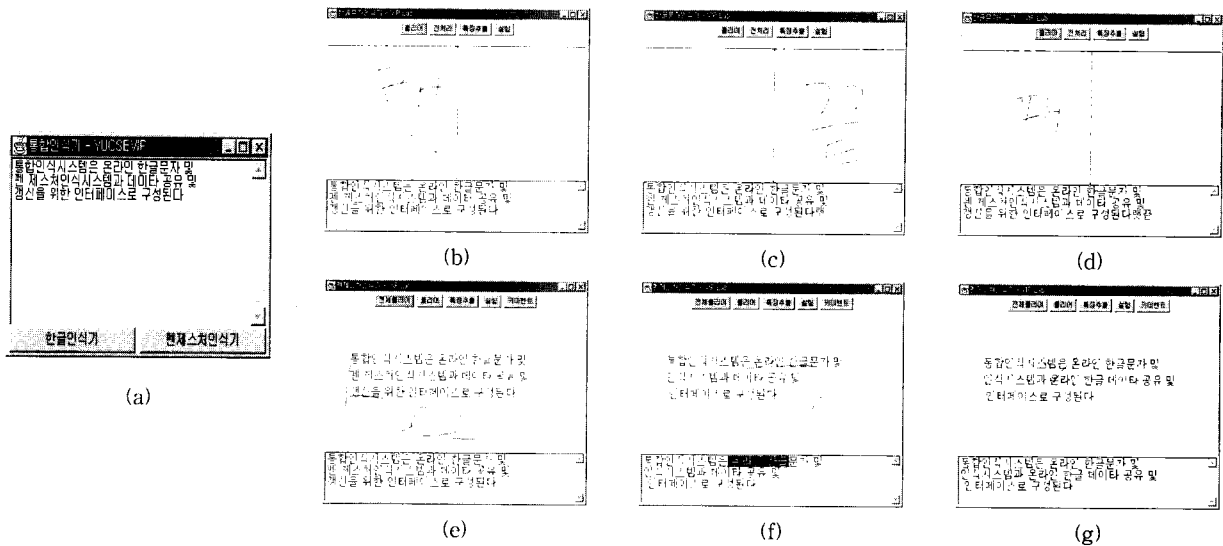
4.2 펜 제스처 인식

한글문자 인식실험에 참여한 사용자들에 의해 쓰여진 500개의 펜 제스처를 대상으로 성능 평가한 결과 <표 5>와 같이 97.8%의 펜 제스처 인식률과 30msec의 평균처리속도를 얻었다. 정의된 15가지의 펜 제스처에 대해 6가지의 분류특징만을 정의하여 펜 제스처모델을 구축함으로써 높은 인식률과 빠른 평균처리속도를 얻었다. (그림 21)은 인식에 성공한 펜 제스처를 보여준다. 미인식을 유발하는 펜 제스처는 변형이 가미되어 정확한 분류특징을 추출할 수 없는 (그림 22)과 같은 펜 제스처들이었다. (그림 22)(a)와 (그림 22)(b)는 굴곡이 너무 가미되어 정확한 특징점을 추출하지 못해 잘못된 방향벡터와 방향코드 수를 추출한 경우이다. (그림 22)(c)는 획의 잘못된 입력으로 인하여 펜 제스처모델에서 검색할 수 없는 방향벡터와 위치관계정보를 추출하였고 (그림 22)(d)는 특징점의 잘못된 추출로 인하여 교차점을 찾지 못한 경우로써 미인식을 유발하였다.

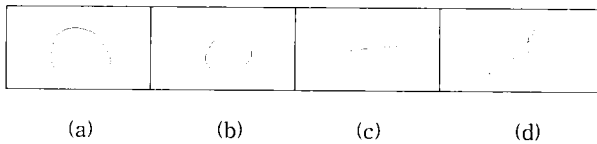


(그림 21) 인식 가능한 펜 제스처





(그림 23) 통합인식 시스템의 실행화면 : (a) 텍스트 모듈, (b)~(d) 한글문자 인식 시스템의 실행화면, (e)~(g) 펜 제스처 인식 시스템의 실행화면



(그림 22) 인식 불가능한 펜 제스처

〈표 5〉 펜 제스처 인식을 및 처리속도

| 실험개수 | 미인식 개수 | 정인식 개수 | 인식률   | 평균 처리시간 |
|------|--------|--------|-------|---------|
| 500  | 11     | 489    | 97.8% | 30msec  |

### 4.3 통합 인식 시스템

실질적인 응용 시스템의 적용을 위해 본 논문에서 제안한 한글문자 인식 시스템과 펜 제스처 인식 시스템을 통합하여 구현하였다. (그림 23)는 구현한 통합 인식 시스템의 실행화면으로 (그림 23)(a)는 입력되거나 갱신된 텍스트를 보여주는 텍스트 모듈이며 (그림 23)(b)~(그림 23)(d)는 텍스트를 입력하기 위한 한글문자 인식 시스템의 실행화면을 보여준다. (그림 23)(e)~(그림 23)(f)는 사용자가 키보드를 이용하여 입력한 텍스트나 한글문자 인식 시스템에서 인식된 텍스트를 대상으로 편집 및 수정하기 위한 펜 제스처 인식 시스템의 실행화면을 보여준다. 실시간으로 각 모듈간의 텍스트를 공유 및 갱신하기 위한 인터페이스는 텍스트 모듈에서 사용자의 직접적인 텍스트 입력, 펜을 이용한 한글문자 인식 시스템에서의 문자 입력, 펜 제스처 인식 시스템에서의 텍스트 편집 및 수정 등에 의해 수행된다.

## 5. 결 론

본 논문에서는 획 정보를 이용한 한글문자와 펜 제스처 인식 시스템을 설계 및 구현한다.

한글문자 인식에서 자소분리는 입력문자의 형태정보에 따라 순차적 자소분리와 백트래킹 자소분리를 수행함으로써 불필요한 재처리과정을 제거하였다. 또한 한글의 특성에 따른 다양한 자소 정보들로 한글데이터베이스를 구축하여 사용자들의 다양한 필체 유형에 따른 문자도 인식이 가능하였다. 펜 제스처 인식은 획 내의 교차수, 방향변화, 방향벡터, 방향코드의 개수, 위치관계, 시작점과 끝점간의 거리비율정보를 분류특징으로 정의하여 펜 제스처모델을 구축함으로써 정의된 펜 제스처들에 대해서 빠른 처리속도와 높은 인식률을 보였다. 통합 인식 시스템에서의 텍스트 편집 및 수정 기능은 펜 제스처나 사용자의 직접적인 텍스트 편집기능에 의해 수행되며 실시간으로 모듈간의 데이터를 공유 및 갱신한다. 제안한 인식방법에 의해 설계 및 구현된 인식 시스템은 키보드나 펜 사용자인터페이스를 요구하는 개인용 컴퓨터, 멀티미디어 단말기, PDA(personal digital assistant)등 다양한 응용 시스템으로의 적용이 가능하다.

향후과제로써 한글문자 인식은 제한된 필체 유형정보를 가지는 한글데이터베이스의 확장과 본 논문에서 배제한 자소나 문자간의 흘림이나 회전이 가미된 문자에 대한 연구가 필요하며 펜 제스처 인식은 좀 더 많은 펜 제스처를 인식하기 위한 추가적인 분류특징에 관한 연구와 불규칙한 문자나 도형에서의 적용이 기대된다.

## 참 고 문 헌

[1] M. T. Vo, C. Wood, "Building an Application Framework for Speech and Pen Input Integration in Multimodal Learning Interfaces," In Proceedings of IEEE, pp.3545-3548, 1996.  
 [2] A. C. Long, S. Narayanaswamy, A. Burstein, R. Han, K.

Lutz, B. Richards, S. Sheng, R. W. Brodersen and J. Rabaey, "A Prototype User Interface for a Mobile Multimedia Terminal," Conference Companion on Human Factors in Computing Systems, pp.81-82, May, 1995.

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

[4] 권오성, 권영빈, "선분정합에 의한 흘림체 온라인 한글 인식", 인지과학논문지, 제3권 제2호, pp.271-289, 1992.

[5] 최정훈, 권희용, 김춘석, 황희용, "신경망 모델을 이용한 한글 필기체 온라인 인식", 정보과학회논문지, 제17권 제5호, pp. 540-549, 1990.

[6] 장석진, 강선미, 김혁구, 노우식, 김덕진, "자소 인식 신경망을 이용한 한글문자 인식에 관한 연구", 전자공학회논문지, 제31-B권 제1호, pp.81-87, 1994.

[7] 신봉기, 김진형, "은닉 마르코프 모델을 이용한 한글 필기 생성", 정보과학회논문지, 제24권 제1호, pp.32-42, 1997.

[8] 신봉기, 김진형, "자소 탐색에 기반한 온라인 한글 인식", 정보과학회논문지, 제23-B권 제11호, pp.1135-1144, 1996.

[9] 김태균, 이은주, "한글에 적합한 획 해석에 의한 연속 필기 한글의 온라인 인식에 관한 연구", 정보과학회논문지, 제15권 제3호, pp.171-180, 1988.

[10] 전병환, 김성훈, 김재희, "퍼지 결정 트리를 이용한 온라인 필기 문자의 계층적 인식", 전자공학회논문지, 제31-B권 제3호, pp.132-140, 1994.

[11] A. Waibel, M. T. Vo, P. Duchnowski, S. Manke, "Multi-modal Interfaces," Artificial Intelligence Review 10, pp.299-319, 1996.

[12] S. N. Krishnan, Shinji Moriya, "One Stroke Operations : A New Pen based User Interface that can Integrate or Separate Operand Specification, Menu Opening and Selection, and Action Execution, in One or More Strokes," Transactions of Information Processing Society of Japan, Vol.37, No.12, pp.2419-2437, 1996.

[13] J. S. Lipscomb, "A trainable gesture recognizer," Pattern Recognition 24, Vol.9, pp.895-907, September, 1991.

[14] M. J. Fonseca, J. A. Jorge, "Using Fuzzy Logic to Recognize

Geometric Shapes Interactively," In Proceedings of IEEE, pp.291-296, 2000.

[15] 윤영선, "벡터 표현에 의한 온라인 필기 한글 인식", 한국과학기술원 전산공학과 석사학위논문, 1992.

[16] 성운재, "계층적 곡선표현기법을 이용한 온라인 필기 한글 인식", 한국과학기술원 전산공학과 석사학위논문, 1991.

[17] C. C. Tappert, C. Y. Shuen, T. Wakahara, "The State of the Art in OnLine Handwriting Recognition," IEEE Transactions on PAMI, Vol.12, No.8, pp.787-808, August, 1990.

[18] K. S. Fu, "Step towards unification of syntactic and statistical pattern recognition," IEEE Transactions on PAMI, Vol.5, No.2, March, 1983.

[19] 김태균, "구조해석에 의한 한글 인식", 한국정보과학회 인공지능연구회 영상처리 및 이해에 관한 워크샵 발표논문집, pp.110-116, 1989.



**오 준 택**

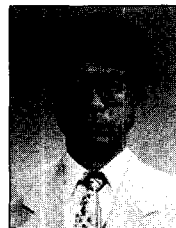
e-mail : ojt@cse.yu.ac.kr

1999년 영남대학교 컴퓨터공학과 졸업 (학사)

2001년 영남대학교 대학원 컴퓨터공학과 졸업(공학석사)

2001년~현재 영남대학교 대학원 컴퓨터공학과 박사과정

관심분야 : 문자 인식, 영상처리, 시각정보처리



**김 욱 현**

e-mail : whkim@yu.ac.kr

1981년 경북대학교 전자공학과 졸업(학사)

1983년 경북대학교 대학원 컴퓨터공학과 졸업(공학석사)

1993년 일본 쓰쿠바대학 공학연구과(공학박사)

1983년~1993년 한국전자통신연구원 선임연구원

1994년~현재 영남대학교 전자정보공학부 부교수

관심분야 : 시각정보처리, 패턴 인식, 영상처리