論文2002-39SP-1-11

유닛 재구성 방법을 이용한 PDA용 온라인 필기체 한자 인식

(On-line Handwriting Chinese Character Recognition for PDA Using a Unit Reconstruction Method)

陳 元 *, 金 基 斗 *

(Won Chin and Ki-Doo Kim)

요 약

본 논문에서는 PDA용 온라인 필기체 한자 인식기를 구현하였다. PDA는 PC보다 느린 CPU와 적은 메모리를 사용하기 때문에, 본 논문에서는 적은 연산량과 적은 메모리를 사용하면서 높은 인식률을 갖는 인식기를 개발하는데 초점을 맞추었다. 따라서, 빠른 인식을 위하여 적은 연산 과정을 갖는 인덱스 매칭 방법을 사용하였고, 필기 한자의 획순 변동과 획수 변형을 수용함과 동시에, 문자 모델의 저장을 위한 메모리를 최소화하기 위하여 유닛 재구성 방법을 제안하였다. 사전에 정의된 유닛을 사용하여 1800개의 표준 문자 모델을 설정하였다. 입력된 데이터는 전처리 및 특징 추출 과정을 거친 후 표준 문자 모델과의 획수 및 형태적 특징을 기준으로 선정된 후보 문자들과의 유사도를 측정한다. 실험 대상 문자는 중·고등학교 표준 기초 한자 1800자를 대상으로 하였으며, 획수와 획순에 구애받지 않고 정서체로 필기한 5인의 문자 셑을 사용하였다. 실험은 문자 당 평균 인식 속도와 인식률을 측정하였으며, 이 결과 문자 셑에 대한 평균 인식률 94.3%를 얻었다. 문자 당 평균 인식 속도는 MIPS R4000 CPU를 사용한 PDA에서 0.16 초의 결과를 내었다.

Abstract

In this paper, we propose the realization of on-line handwritten Chinese character recognition for mobile personal digital assistants (PDA). We focus on the development of an algorithm having a high recognition performance under the restriction that PDA requires small memory storage and less computational complexity in comparison with PC. Therefore, we use index matching method having computational advantage for fast recognition and we suggest a unit reconstruction method to minimize the memory size to store the character models and to accomodate the various changes in stroke order and stroke number of each person in handwriting Chinese characters. We set up standard model consisting of 1800 characters using a set of pre-defined units. Input data are measured by similarity among candidate characters selected on the basis of stroke numbers and region features after preprocessing and feature extracting. We consider 1800 Chinese characters adopted in the middle and high school in Korea. We take character sets of five person, written in printed style, irrespective of stroke ordering and stroke numbers. As experimental results, we obtained an average recognition time of 0.16 second per character and the successful recognition rate of 94.3% with MIPS R4000 CPU in PDA.

* 正會員, 國民大學校 電子工學部
(Department of Electronics Engineering, Kookmin University)

接受日字:2001年3月9日, 수정완료일:2001年12月21日

Ⅰ. 서 론

오늘날 휴대용 컴퓨터나 PDA(Personal Digital Assistant)의 보급률은 빠르게 성장하고 있다. 이들 기

기의 특징은 크기가 소형화되어 휴대가 용이하다는 점이다. 이에 따라 입력 장치의 변화가 요구되었고, 현재, 펜 입력 방식의 필기체 문자 인식(On-line Handwritten Character Recognition) 기술이 음성인식 기술과 더불어 중요하게 대두되고 있으며 많은 연구가 이루어지고 있다. 필기체 입력 문자는 글자의 모양, 획수, 획순(필순)이 개인마다 습관에 의한 변형 수가 많으며, 특히 한자의 경우 표현되는 자형(字形)이 많으며, 그 중에서도 유사한 형태의 문자가 많기 때문에 정밀한 문자 인식을 위해서는 많은 연산량과 저장 공간이 요구된다.[1] 이는 PC에 비하여 상대적으로 느린 CPU를 사용하는 PDA의 경우에는 인식률과 더불어 중요한 요소인 인식 속도의 제한과 밀접한 관계를 갖는다. 또한 PDA에 On-line 필기체 인식 기술을 사용할 경우 메모리의 제약으로 인하여 데이터베이스의 크기가 제한되며, 이는 한자와 같이 구별 대상이 많은 경우의 문자인식에서는 중요한 제약 조건이 된다. 따라서, PDA용 필기 인식을 위해서는 메모리 사용의 최소화가 요구된다.

본 논문에서는 PDA용 온라인 필기 한자 인식기를 구현하기 위한 알고리즘을 제안한다. 이는 기존 인식 방식인 신경망, 퍼지, HMM 등의 인식 알고리즘이 갖는 연산량보다 적은 연산으로 좋은 인식률을 가질 수 있는 방법이다. 인식 대상 문자는 중·고등용 기초한자 1800자를 선정하였다. 적은 연산량을 갖는 인식을 위하여 획 인식 방법을 기본 획과 특수 획에 따라 다르게 적용하였으며, 특수 획 분리 방법으로 선형 분리 방법을 제안하였다. 또한 한자 필기에서 개인적 성향에 의하여 나타나는 획순 변형과 획수 변형을 수렴하고, 문자 모델에 대한 저장성의 효율을 증대시키기 위하여 사용된 유닛 재구성 방법을 제안한다. 테스트에 사용된 데이터베이스는 정서체로 필기한 5 셀의 1800 한자 필기 문자 데이터를 사용하였다. 한자 1800자는 중,고등학교 교육용 기초 한자를 대상으로 하였다.[2]

본 논문의 구성은 다음과 같다. I장의 서론에 이어 II장에서는 인식 대상 한자에 대한 분석을 다루며, III장과 IV장에서는 본 논문에 사용된 입력 문자에 대한 전처리 과정과 특징 추출 과정에 대하여 설명한다. V장에서는 유닛 재구성 방법을 설명하고, 유닛을 사용하여 문자 모델을 구성하는 방법에 대하여 설명한다. VI장에서는 후보 선택에 의한 분류 및 스코어 계산 방식에 의한 인식 방법을 다루며, VII장은 실험 및 결과에 대한 고찰을 논한다.

II. 인식 대상 한자의 특징 분석

본 논문은 국내 교육용 기초 한자 1800자를 인식 대상으로 선정하였다. 이번 장에서는 표준 문자 셀에 대한 특징을 통계 조사한다. 정서체 필기 방식으로 입력받은 필기 데이터에 대하여 획의 종류, 획수, 획순, 필기 경사도 등을 조사하여 이를 바탕으로 특징 파라미터 추출 시 필요한 각종 경계 값을 선정한다.

대상 문자의 획수 분포는 인식 방식을 설계하는데 필요하다. 본 논문에 사용된 대상 문자들에 대한 표준 획수별 분포도는 그림 1과 같다. 문자별 가장 많은 획수는 26획이며, 최고 빈도 획수는 11획이다. 또한 8-12획 사이의 문자가 전체 문자의 약 45%를 나타낸다.

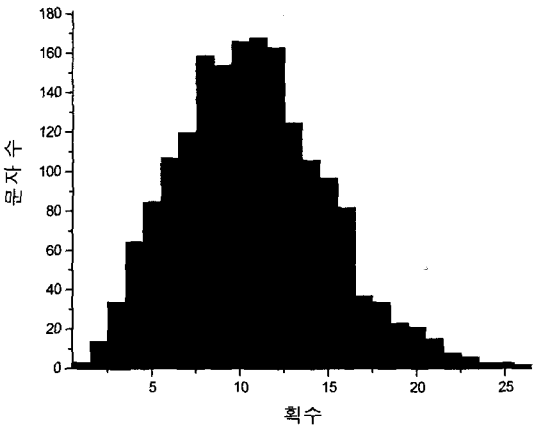


그림 1. 교육용 기초 한자 1800 자의 획 수 분포도
Fig. 1. Distribution graph of Chinese character stroke number.

대상 문자들은 개인의 필기 습관에 따라, 약 1-3획의 획수 변화가 가능하다. 또한 획 순서(이하 획순)도 개인적 성향에 따라 차이를 나타낸다. 평균적으로 약 20%가 틀린 획순으로 쓰여지며, 획순변동의 비율은 획수가 많을수록 늘어난다. 또한 같은 자종이라도 여러개의 획순으로 쓰여질 수 있다. 그러나 대부분의 문자는 일반적으로 과도한 획순변화를 보이지 않으므로, 과도한 획순변동에는 대응할 필요가 없다.[3] 또한 입력 방식의 제약이 없다면, 필기 문자의 크기도 다를 수밖에 없다. 따라서, 일반적인 획수 및 획순의 변화를 수렴하면서, 크기의 제약을 받지 않는 인식기 설계가 필요하다. 실제로 위의 변화를 수렴하기 위하여 여러 가지 방

법이 시도되고 있으며, 많은 좋은 알고리즘들이 개발되어 왔다.[4] 그러나, 인식률과 메모리 사이즈 및 인식 시간과는 역의 관계가 있기 때문에, 적은 메모리와 연산량을 사용하면서 인식률을 높이고자 하는 것이 본 논문의 목적이다. 본 논문에서는 이를 위하여, 유닛 재구성 방법과 인덱스 비교를 통한 스코어 할당 방식을 사용한다.

본 논문에서는, 문자의 특징을 나타내기 위하여 획 인덱스를 할당하는 방식을 사용한다. 획 인덱스를 할당하는 방식에는 여러 가지 방법이 소개되고 있다.[2,4] 본 논문에서는 획을 크게 기본 획과 특수 획으로 구분하고자 한다. 기본 획은 직선 형태의 획으로 획의 꺾임 부분이 없는 획으로, 5가지 종류로 정의한다. 단, 획의 끝점 부근의 삐침이 포함된 형태는 꺾임 부분이 있지만 획 분류 단계에서 기본 획으로 정의된다. 특수 획은 한 부분 이상의 꺾임 부분을 포함하는 획으로, 7가지 종류로 분류하였다. 일반적으로 획을 구분할 때 7종 분류를 많이 사용하지만,[1] 유사 문자들에 대한 변별력을 높이기 위하여 본 논문에서는, 전체 12종의 획 형태로 분류하였다. 획 종류는 그림 2에서 볼 수 있다. 본 논문에서 사용된 필기 문자 데이터베이스로부터 추출한 기본 획과 특수 획의 빈도는 그림 3과 같다.

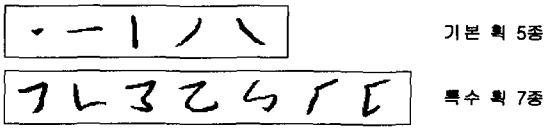


그림 2. 획종 정의
Fig. 2. Stroke classification.

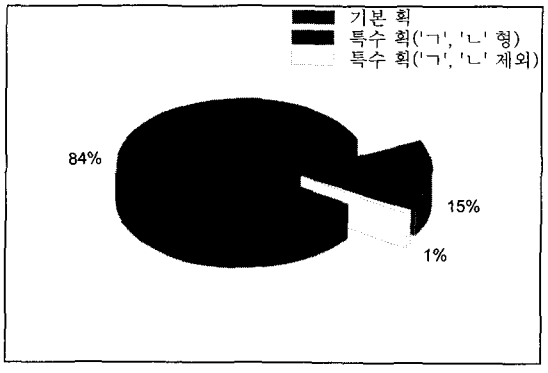


그림 3. 획 형태 분류
Fig. 3. Classification of stroke form.

Ⅲ. 한자 유닛 재구성 방법

1. 유닛 재구성 방법(Unit Reconstruction Method)

유닛(Unit)은 문자 내에서 독립적으로 존재하는 구성 요소라 정의한다. 유닛 재구성 방법은 획순 및 획수 변동을 수렴하는 표준 문자 모델을 정의하기 위한 방법으로 사용된다. 유닛을 정의하기 위한 가설은 다음과 같다.

가설 1. 유닛과 유닛간의 획순 변동은 없다.
가설 2. 모든 한자는 유닛의 결합으로 이루어진다.

대부분의 획순 변화는 부분적으로 일어나며, 이는 유닛 내부적으로 일어난다. 따라서, 가설 1은 문자의 정확한 인식을 위해서 유닛간의 획순 변동은 없어야 함을 나타낸다. 가설 2는 모든 표준 문자는 유닛의 결합으로만 이루어지며, 따라서 문자를 정의하기 위하여 사전 정의된 유닛의 인덱스만을 사용하면 됨을 의미한다. 이는 중복되는 문자 모델 내의 정보를 단일화시킴으로서 결과적으로 메모리 사용량을 감소시킨다. 또한 인식 대상 문자의 확장을 위하여 편리한 환경을 제공한다.

2. 유닛 구성

1개의 유닛은 문자가 갖을 수 있는 다양한 형태의 획수와 획순을 수렴하기 위하여 다중의 특징 파라미터로 구성된다. 필기자의 개인적 습관에 따라 유닛 필기시에 다양한 패턴의 획순이 발생할 수 있으나, 일반적인 경우 획순의 변화는 일정한 규칙이 있으며, 따라서 일반적으로 발생하는 획순은 전체 가능한 획순보다 훨씬 적다. 모든 가능한 변화를 표현하기 위해서 메모리를 사용하는 것은 전체 인식 속도와, 메모리의 효율적인 사용 측면에서 불합리하다. 따라서, 본 논문에서는 일반적으로 빈도 높은 획순 및 획수의 특징 파라미터를 유닛 각각에 삽입함으로서 전체 문자가 가질 수 있는 획순 및 획수 변화에 대처할 수 있도록 하였다. 그림 4는 획순 변동과 획순 변동에 대처하기 위해 문자를 구성하는 유닛의 내부형태를 나타낸다.

유닛은 획순과 획수의 변화 종류만큼의 획 특징 파라미터 열과, 가상 획(Ligature) 특징 파라미터 열로 구성된다.[5]

3. 유닛 설정

위의 정의된 가설을 만족시키기 위하여 모든 문자를 구성하는 요소를 유닛으로 구성한다. 이에 필요한 유닛은 4가지 기준에 의하여 분류하며, 이는 본 논문에 사용된 1800자 문자에 대한 형태적 분석에 의하여 정의되었다. 유닛을 나누는 기준은 문자의 형태를 보는 관점에 따라 달라질 수 있다. 그림 5는 유닛 설정 기준 예를 보여준다.

독립적 영역을 갖는 형태에 해당하는 유닛 중 부수 형태는 가장 먼저 유닛으로 설정될 수 있다. 그러나 부수 모음 형태에 해당하는 유닛은 가장 마지막에 설정되며 경험적으로 설정된다. 유닛과 유닛의 결합으로 이루어지는 유닛이기 때문에 사용 빈도에 따라 결정될 수 있기 때문이다. 이는 유닛의 개수를 증가시키는 역할을 하지만, 문자의 위치적 특징을 나타내는데 적합하다. 예외 형태에 해당하는 유닛 중 부분 형태는 단일 획, 혹은 적은 획 수로 이루어지는 경우이며, 의미 없는 연결 고리이다. 독립 형태는 상형문자에 대부분 나타나는 형태로서 문자를 유닛으로 나누는 것이 부자연스러운 경우이다. 이 경우는 각각의 획이 문자 내에서 유닛의 역할을 하며, 위치 관계를 나타내는 단위도 획이 된다. 표 1은 앞서 설명한 유닛 설정 내용을 요약한 것이다.

표 1. 유닛 설정
Table 1. Setting up of unit.

형태 특징	유닛구분	형태적 특징
독립적 영역을 갖는 형태	부수	한자에서 자주 사용되는 부수 형태
	부수모음	유닛과 유닛의 결합 형태
예외 형태	부분	연결고리, 단일 획, 부수 아님
	독립	상형문자, 문자 전체가 단일 유닛

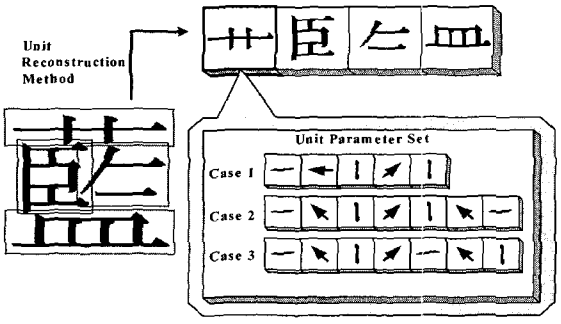


그림 4. 문자 모델 구성도
Fig. 4. Word model configuration.

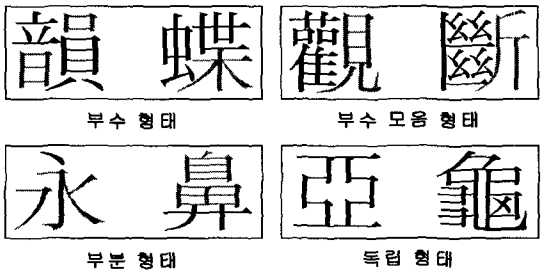


그림 5. 유닛 설정 기준 예
Fig. 5. Example of unit setting up.

Ⅳ. 특징 추출

Ⅳ장에서는 입력된 2차원 데이터 열로부터 문자의 특징을 추출하는 과정에 대하여 설명한다. 입력된 데이터는 정규화시키기 위하여 전처리 과정을 수행하며, 획, 가상획, 영역 특징 파라미터를 추출하여 인식과정에서 문자를 대표하는 파라미터로 사용된다.

1. 전처리(Preprocessing)

전처리 과정에서는 입력 디바이스로부터 들어온 원시 데이터를 정규화 한다. 본 논문에서는 스무딩(Smoothing), 훅 제거(Dehooking), 거리 여과(Distance Filtering) 및, 크기 및 위치 정규화의 과정을 거친다.[6]

1) 스무딩(Smoothing)

입력 디바이스에 한자를 필기할 때 발생하는 손의 떨림, 디바이스 표면의 문제, 그리고, 입력 방식의 이산적 특징 등으로 인하여 필기 입력 데이터의 불연속성을 제거하는 필터링이다. 본 논문에서는 계산의 효율성을 위하여 식(1)과 같은 스무딩 함수를 사용한다.

$$p_t = \{x_t, y_t\}$$
$$p'_t = \alpha p'_{t-1} + (1-\alpha) p_t \quad 0 < \alpha < 1 \tag{1}$$

여기서, p_t는 x_t와 y_t로 이루어진 획을 구성하는 한 점을 나타내며, α는 스무딩 상수로 $0 < \alpha < 1$의 값을 갖는다. p'_t는 스무딩 과정으로 변화된 점을 의미한다.

2) 훅 제거(Dehooking)

훅 제거 과정은 필기 데이터 입력 시 생기는, 필자가 원하는 펜의 궤적과 무관하게 발생한 데이터를 삭제하는 과정이다. 훅은 획의 시작점 부근과 끝점 부근에서

발생하며 이는 부정확한 펜 눌림이나 잘못된 필기 습관, 한자의 경우 의도적인 삐침의 추가 등에 의하여 발생한다. 훅 제거 과정은 이들을 제거함으로서 획 인덱스 할당 과정에서의 오류 확률을 줄일 수 있다. 단, 한자 필기 시 나타나는 끝점 부근의 의도적 삐침은 그 길이의 정도가 개인적으로 많은 편차를 나타내므로, 본 논문에서는 시작점 부근에 한하여 훅 제거를 실시한다. 끝점 부근에 나타나는 훅은 특수 획 인덱스 할당 방법인 선형 분리 방법에서 끝점 부근의 훅의 방향에 대하여 구별된 처리를 함으로서 이를 해결하였다. 시작점 부근의 훅을 제거하는 방법은 다음과 같다. 그림 6은 훅 제거 방법에 대한 순서도를 나타낸다.

단계 1. 시작점 부근의 훅 영역 설정

단계 2. 훅 영역 내에 있는 점들의 사이각 계산

단계 3. 가장 작은 사이각을 갖는 점 조사

단계 4. 단계 3.에서 조사된 사이각과 임계각을 비교 후 획 시작점 변경

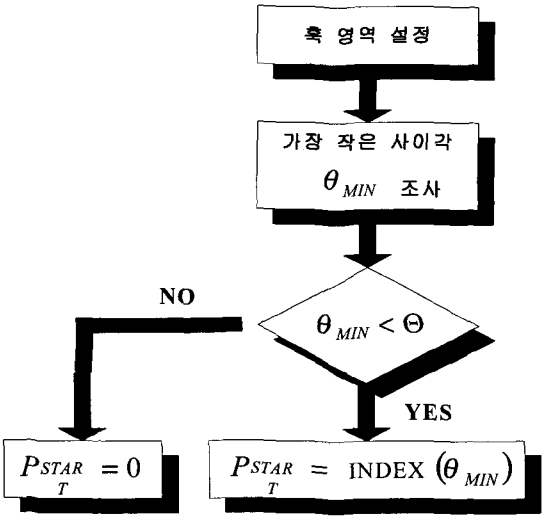


그림 6. 훅 제거 과정 순서도
Fig. 6. Flow chart of hook elimination procedure.

3) 거리 여과

입력 디바이스로부터 들어오는 문자 데이터는 일정한 샘플링 시간 간격에 의하여 생성되기 때문에 필기 속도에 의해 입력된 점과 점 사이의 거리의 변화가 생긴다. 따라서, 특징 추출과정에서 필요한 균일한 거리의 점 분포를 만들기 위하여 거리 여과 과정이 요구된다. 거리 여과 과정을 거쳐서 점 사이 거리가 일정한 문자 데이터는 획 인덱스 할당을 위한 선형 분리 방법에서 필요하다. 거리 여과 과정은 점이 몰려 있는 부분, 즉 점과 점 사이의 거리가 짧은 부분의 점을 제거하는 과정과, 점과 점 사이가 먼 부분에 점을 추가시켜 주는 과정의 두 과정을 거친다.

4) 크기 정규화

크기 정규화 과정에서는 입력 데이터의 문자 영역 사이즈를 64×64 픽셀의 크기로 조정한다. 입력 데이터가 갖는 영역이 정사각형의 형태가 아니므로 문자의 폭과 높이 중 큰 값을 기준으로 정규화 비율을 설정한다. 정규화 후에 글씨의 형태가 변화되지 않도록, X축과 Y축에 같은 비율로 정규화 과정이 수행되며, 문자 영역의 중심점이 정규화 후에도 중심점이 되도록 조정한다. 그림 7은 전처리 과정을 거친 필기 데이터베이스의 예를 보여준다.

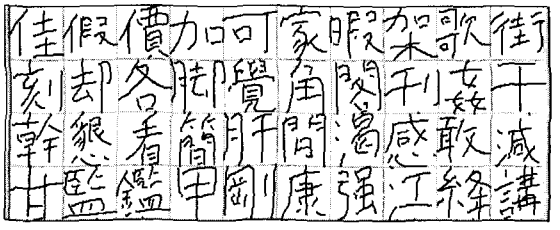


그림 7. 전처리 과정을 거친 필기 데이터베이스의 예
Fig. 7. Handwritten database after pre-processing.

2. 특징 추출

본 논문에서는 문자를 구성하는 기본 단위를 획(Stroke)이라고 정의한다. 획은 입력 디바이스에서 펜을 떼지 않고 한번에 쓴 궤적을 의미한다. 특징 추출 과정은 대상 문자의 인식을 위하여 문자가 갖는 특징을 파라미터화 하는 과정이다. 여기서 추출된 특징 파라미터들은 유사 형태의 다른 문자들과 구분될 수 있는 특징을 가져야 한다. 본 논문에서는 연속 입력으로 생성되는 획순에 의거한 획별 특징 파라미터와 획과 획 사이에 펜의 움직임에 대한 가상 획 특징 파라미터와, 전체 입력 데이터에 대한 문자 내의 영역을 나타내는 특징 파라미터를 추출한다. 연산량을 줄이기 위하여 획 인덱스 부여 과정에서 기본 획과 특수 획으로 나누어 인덱스를 부여하였고, 특수 획 인덱스를 부여하기 위하여 선형 분리 방법(Line Segment Method)을 사용하였다.

1) 획 특징 추출

본 논문에서는 획이 갖는 특징을 4개의 특징 파라미터를 사용하여 나타내며, 이는 각각 획 인덱스, 곡률,

길이, 방위이다. 대상 문자에 대하여 4개의 파라미터가 획별로 각각 구해지고, 이 파라미터들을 사용하여 대상 문자와 표준 문자와의 기본적인 획간 유사도를 측정하게 된다. 정서체 한자의 경우 대부분이 단순한 직선의 형태를 가지며 이를 기본 획이라 명명하였고, 비교적 간단한 방법을 통하여 인덱스가 부여된다. 그러나 그렇지 않은 획, 즉 특수 획의 경우, 어떤 형태의 획인지를 나타내는 인덱스를 부여하기 위하여 기본 획과 다른 방법을 도입한다.

① 획 길이

전처리 된 획의 모든 점을 연결하는 선분의 길이를 나타낸다. 이를 파라미터로 사용하기 위하여 3 종류의 길이 인덱스가 부여된다. 길이 인덱스의 부여 방식은 문자의 복잡도에 따라 조건이 변경된다. 문자가 복잡해지면 64×64 정규화 작업으로 인하여 획의 길이가 전체적으로 짧아지는 경향이 있으므로, 이로 인하여 길이 인덱스의 부여 시에 나타나는 왜곡을 최소화하기 위함이다. 문자의 복잡도는 문자의 획의 개수와, 문자의 정규화 비율과의 조합으로 구하여진다.

$$D_i = \sum_{j=0}^{N_i-2} \text{distance}(p_i(j),\ p_i(j+1)) \qquad (2)$$

여기서, distance 함수는 두 점의 거리를 구하는 함수이며, D_i는 문자 내의 i번째 획의 길이를 나타낸다. $p_i(j)$는 i번째 획의 j번째 점을 의미하며, N_i는 i번째 획의 점의 개수를 나타낸다.

$$\alpha = \frac{l}{L}T \quad , \quad M \le 10 \qquad (3.a)$$
$$\alpha = \frac{l}{L}(0.1(M-10)+1)T \quad , \quad M > 10$$

$$\beta = 2\alpha \qquad (3.b)$$

여기서 $\frac{l}{L}$은 문자의 가로와 세로의 비율 중 작은 값 l과 표준 비율 L의 비율로 문자비율을 나타내며, M은 입력 데이터가 갖는 획수, T는 획 길이 임계값을 나타낸다. α와 β는 획 길이 인덱스를 지정하기 위한 임계값이다. 표 2는 식(3)에서 구한 α와 β값을 사용한 길이 인덱스 지정 조건을 나타낸다.

표 2. 길이 인덱스 할당 조건
Table 2. Condition for length index allocation.

	점	짧은 획	긴 획
영역	$D_i < \alpha$	$\alpha \le D_i < \beta$	$D_i \le \beta$
비고	α : 점 획 판단 임계값 β : 짧은 획 판단 임계값		

② 획 곡률

본 논문에서 정의된 획의 곡률은 획의 전체 길이와, 획의 시작점과 끝점을 연결하는 직선의 길이의 비를 나타낸다. 곡률 파라미터는 기본 획과 특수 획을 구분하는 기준 파라미터로도 사용되며, 획간 유사도 측정에서 획 인덱스의 불일치를 보상하는 보상 파라미터로 사용된다.

$$D_i' = \text{distance}(p_i(0),\ p_i(N_i-1)) \qquad (4.a)$$

$$C_i = \frac{D_i}{D_i'} \qquad (4.b)$$

식(4.a)에서 D_i'은 i번째 획의 시작점과 끝점을 연결하는 직선의 길이를 의미한다. 또한 식(4.b)에서 C_i는 획의 곡률을 나타낸다.

③ 획 인덱스 할당

기본 획과 특수 획을 나누는 기준 파라미터는 곡률이다. 획 곡률이 임계값 미만이기 때문에 기본 획으로 구분된 일반적인 직선 형태의 획은 방향 코드를 사용하여 사전 정의된 기본 획의 종류로 분류되며, 파라미터화 하기 위하여 인덱스가 할당된다. 그리고, 획 곡률이 임계값 이상인, 일반적인 직선 형태의 획이 아닌 다른 형태의 획, 즉 특수 획에 대한 인덱스를 할당하기 위하여, 본 논문에서는 선형 분리 방법을 사용한다. 표 3은 기본 획과 특수 획의 구분 조건을 나타낸다.

표 3. 기본 획과 특수 획의 구분
Table 3. Classification of basic stroke and special stroke.

	기본 획	특수 획
영역	$C_i \le 1.20$	$C_i > 1.20$

(가) 기본 획의 인덱스 할당

기본 획은 획 곡률이 임계값 미만이므로 직선으로 간주된다. 따라서, 시작점과 끝점만이 직선을 나타내는 정보를 나타내는 점으로 사용된다. 기본 획 인덱스 부여 방법은 그림 8과 같이 각 영역을 분류한 테이블에 의하여 인덱스가 부여된다. 여기서 θ는 필기 문자의 수평 획 평균 경사도를 의미한다. 이와 같은 영역 할당은 기본적으로 필기 문자의 수평 획에 대한 평균 경사

도가 포함된다. 이는 일반적으로 문자 필기 시에 나타나는 약 10도의 수평 획의 경사도를 흡수하는 역할을 한다. 실제로 기본 획의 각도 영역에 포함되지 않는 기본 획은 대부분 '점'의 형태를 취하는 짧은 길이의 획이며, 이에 해당하지 않는 획은 0.001% 미만으로 대부분이 정상적으로 필기된 획이 아니다. 따라서, 고려 대상에서 제외된다. 점 형태의 획은 각도 파라미터가 아닌 길이 파라미터에 의해서 인덱스가 부여되므로 각도 파라미터의 영역이 설정되어 있지 않을지라도 문제가 되지 않는다.

일차로 점 형태의 인덱스가 부여되고 난 나머지 획들에 대하여 기본 획 인덱스가 부여된다. 따라서, 기본 획 인덱스가 부여되는 획들은 길이 인덱스가 중간 획, 긴 획에 해당되는 획들이고 표 4에 나타난 바와 같다.

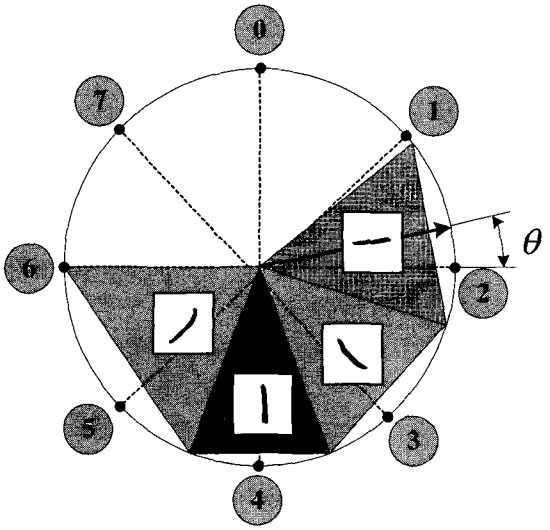


그림 8. 수평 획의 경사도를 포함한 기본 획의 각도 영역

Fig. 8. Angle area of basic stroke including horizontal stroke gradient.

표 4. 기본 획 인덱스 테이블

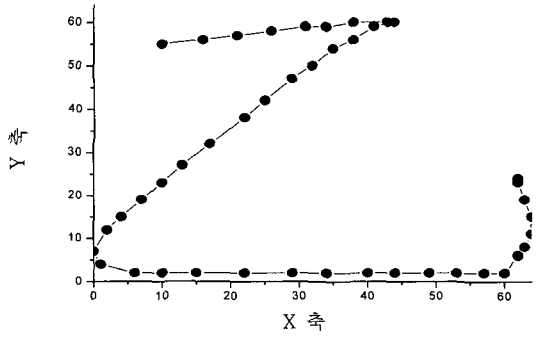
Table 4. Basic stroke index.

영역 코드	A영역	B영역	C영역	D영역	그 외
길이	짧은 획, 긴 획				길이와 무관
인덱스	01	02	03	04	99
형태	—	丨	丿	乀	알 수 없음

(나) 선형 분리 방법

일반적인 획 분류 방법에는 신경망(neural network)을 이용하는 방법, fuzzy 함수를 사용하는 방법, 탄력 정합(elastic matching), 동적 프로그래밍 정합(dynamic programming matching) 등이 있다. 그러나 이들 방법들은 많은 연산량으로 인하여, PC에 비해 상대적으로 느린 CPU를 갖는 PDA에 적합하지 않다. 특수 획의 획 인덱스 할당을 위하여 본 논문에서는 선형 분리 방법을 소개한다. 획의 곡률이 임계값 이상이기 때문에 특수 획으로 구분된 획들은 1개 이상의 굴곡 점을 가지고 있다. 이에 필요한 굴곡 점을 구하기 위하여 2차원 벡터인 점을 X축, Y축 각각으로 분리하여, 편미분 값을 구한 뒤 이 값들 중 기준선을 통과하는 점을 굴곡 점의 후보로 선정한 후, 이 점들 중 의미를 갖는 굴곡 점을 최종으로 선택한다. 의미를 갖는 분절이 되기 위해서는 전체 획이 갖는 길이에 대하여 일정 길이 이상이 되어야 한다. 이러한 굴곡 점을 찾은 뒤 분리된 분절의 곡률이 임계값 미만인 경우에는 이 분절을 직선으로 간주하여 전체 획을 구성하는 분절로서 인덱스를 부여한다. 분절의 인덱스는 8방향 코드를 사용한다. 이렇게 구해진 획을 구성하는 분절들에 대한 인덱스 시퀀스가 구해지면, 사전 정의된 트리 구조의 인덱스 테이블에 의하여 특수 획 인덱스를 할당한다. 그림 9는 선형 분리 방법에 의하여 후보 굴곡 점을 찾는 방법을 나타낸다. 다음은 선형 분리 방법에 대한 순서를 나타낸다.

단계 1. X축, Y축 각각의 편미분 값 계산
단계 2. 영 교차점을 기준으로 후보 굴곡 점 생성
단계 3. 후보 굴곡 점 중 임계 길이 미만의 분절 제거
단계 4. 분절에 대한 방향코드 할당
단계 5. 트리 구조 테이블 매칭



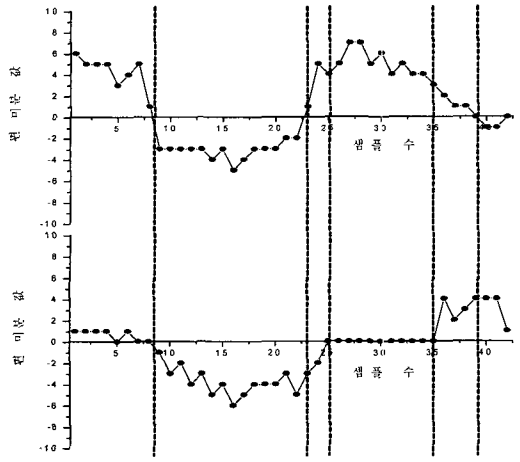


그림 9. 선형 분리 방법 중 굴곡 점 후보 선택 방법
Fig. 9. Selection method of curved point candidates out of line segmentation methods.

④ 획 방위

획 방위 파라미터는 특수 획간의 유사도 측정에서 좀 더 정확한 측정을 위한 보조 파라미터로 사용된다. 방위 파라미터는 기본 획의 경우 인덱스 부여 과정에서 계산되며, 특수 획의 경우는 특수 획 인덱스 부여 과정 후에 계산된다. 획 방위는 획의 시작점과 끝점을 잇는 선분의 각도로서 구하여지며, 범위는 0에서 359 사이의 값을 갖는다. 이는 획 인덱스 할당에서 생기는 오류를 획 간 유사도 측정에서 보상하기 위함이다.

2) 가상 획 특징 추출

가상 획은 Ligature[8]라 불리기도 하며, 이전 획의 끝점과 현재 획의 시작점을 연결하는 선분을 나타낸다. 이는 실제로 필기 된 것이 아닌, 다음 획을 쓰기 위하여 펜이 이동한 궤적을 나타낸다. 가상 획은 이전 획과 현재 획의 위치 관계를 부분적으로 나타낼 수 있는 특징이 있다. 또한, 가상 획은 획순에 종속적이다. 따라서, 개인적인 개성에 의해 획순이 변화하면, 함께 변화하는 파라미터이다. 본 논문에서는 문자 모델의 특징 파라미터로서 가상 획의 특징 파라미터를 사용하고, 획 특징 파라미터와 함께 대상 문자와 표준 문자와의 유사도를 나타내는 스코어 계산에 필요한 파라미터로 사용된다. 본 논문에서는 문자 모델을 유닛 모델로 분해하여 표현하기 때문에, 이에 해당되는 가상 획 파라미터도 유닛 모델 내에 포함되는 가상 획 파라미터와, 유닛과 유닛을 연결하는데 사용되는 가상 획 파라미터로 구분되며, 유사도 측정에서는 같은 파라미터로 사용된다.

가상 획 특징 파라미터는 길이 파라미터와 방위 파라미터로 구성된다. 길이 파라미터는 이전 획의 끝점과 현재 획의 시작점을 연결하는 선분의 길이이며, 방위 파라미터는 이 선분의 방위로서 8 방향 코드의 인덱스가 할당된다. 그림 10은 가상 획 특징 파라미터를 나타낸다. 여기서 l은 가상 획의 길이, θ는 가상 획의 각도를 나타낸다.

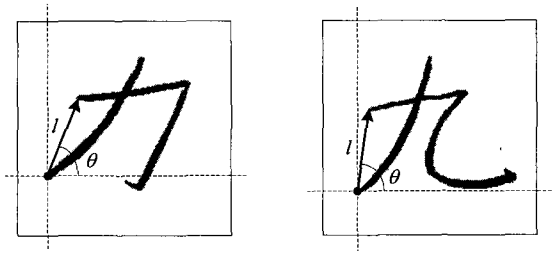


그림 10. 가상 획의 특징 파라미터
Fig. 10. Ligature stroke feature parameters.

3) 획 영역 특징 추출

문자를 유닛으로 재구성 할 때, 문자 내의 영역을 나타내는 획순과 무관한 위치 관계 파라미터가 필요하다. 이는 On-line 필기 데이터와 무관한 Off-line 데이터의 특징이다. 따라서, 본 논문에서는 획순에 무관하게 문자 내의 획들의 위치 관계를 나타내기 위하여 획 영역 특징 파라미터를 추출한다. 이 특징을 나타내기 위하여 후보 문자와의 유사도를 결정할 때, 많은 연산량을 요구하는 특징 파라미터를 사용하는 것은 한자와 같이 비교 대상 문자가 많은 경우에는 적합하지 않다. 따라서, 본 논문에서는 영역 특징을 나타내는 파라미터의 정확성을 중요시하는 것 보다, 연산량을 줄이며 문자의 유사도에 효율적으로 기여할 수 있는 방식을 개발하여 사용하였다. 이 방법은 64x64 의 정규화 영역인 정사각형의 각 꼭지점과 가장 가까운 점을 소유한 획들을 지정하는 방식이다. 대상 문자와 표준 문자 모델과의 유사도를 측정하기 위하여 대상 문자가 유닛으로 재구성 될 때, 재구성된 유닛에 해당하는 획들이 가지고 있는 소유 꼭지점의 위치를 사용하여 유닛과 유닛간의 유사도에 일정 스코어를 부여하는 방식이다. 이 방식은 대상 문자가 표준 문자 모델의 유닛 구성 방식으로 재구성될 때, 재구성된 유닛이 문자 내에서 어떤 영역을 나타내는지를 판단하는 기준 파라미터가 된다. 이 방법은 대상 문자와 표준 문자와의 형태적 유사도를 표현하기

위하여 적은 계산량이 소요되므로, 표준 문자의 확장이 비교적 자유롭다.

획 영역 특징 파라미터는 정규화 영역의 4개의 꼭지점으로부터 가장 가까운 점을 갖는 획을 찾는 것이다. 이 방법에 의하여 선택된 4개의 획은 각각 LT (Left Top), LB(Left Bottom), RT(Right Top), RB(Right Bottom)의 인덱스를 부여받게 된다.

V. 문자 인식

본 논문에서는 표준 문자 셀을 인식 대상 문자의 수와 동일한 1800개의 표준 모델로 설정하고, 문자 모델을 구성하였다. 각각의 문자 모델들은 일반적으로 가능한 획수 및 획순 변형을 수렴한다. 문자 모델은 해당 문자가 갖는 모든 특징을 포함하며, 최종 출력을 위하여 고유 인덱스를 저장하고 있다. 1개의 문자 모델이 갖을 수 있는 획수는 문자 모델을 구성하는 유닛의 획수에 따라 변화될 수 있다.

1. 문자 모델 구성

문자 모델은 유닛들의 결합으로 이루어진다. 유닛과 유닛 사이에는 가상 획 특징 파라미터가 삽입되며, 각 유닛들은 문자의 영역 할당 코드를 가지고 있다. 이는 문자의 형태에 대한 파라미터를 문자 모델이 소유함으로서, 문자의 형태적 특징을 나타낼 수 있기 때문이다. 문자 모델이 가질 수 있는 획수는 문자를 구성하고 있는 유닛들이 가질 수 있는 획수에 의해 결정된다. 그림 11의 문자는 17획부터 19획 사이의 획수를 가질 수 있다. 따라서, 대상 문자의 획수가 17획에서 19획 사이의 값이면, 아래 문자 모델과 유사도를 측정할 수 있다. 그림 11에서 T는 True, F는 False로 이는 문자 영역 사각형의 각 꼭지점을 점유하는 유닛에게 주어지는 인덱스를 나타낸다.

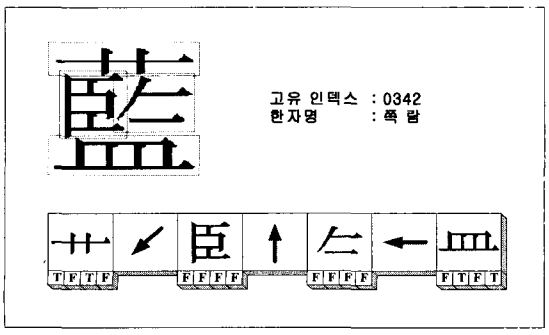


그림 11. 문자 모델 구성
Fig. 11. Configuration of word model.

2. 문자 분류

입력 문자가 갖는 획수와 일치하는 획수를 가질 수 있는 표준 문자 모델이 1차 후보로 선정된다. 따라서, 후보 중에 대상 문자에 대한 인식 문자 모델은 반드시 포함되어야 한다. 표준 문자 모델은 유닛의 결합으로 이루어지므로, 유닛이 가질 수 있는 획수가 정확하게 지정되어 있다면, 표준 문자 모델은 모든 획수 모델을 포함한다. 또한, 입력된 문자의 획별 영역 파라미터를 조사하여 유닛으로 구성된 1차 후보 문자 모델 중에서 형태적으로 유사한 문자를 2차 후보로 선정한다.

3. 스코어 계산

대상 문자의 유닛과 후보 문자의 유닛 사이의 유사도는 인덱스 매칭 방법에 의해 스코어를 구함으로써 이루어진다.[7,8] 또한, 특징 추출 과정에서 구해진 획의 보조 파라미터를 사용하여 스코어는 조정된다. 스코어 계산은 유닛이 가지고 있는 획순 모델의 종류만큼 구하여지며, 이 중에서 가장 높은 스코어를 산출한 값이 최종 스코어로 결정된다. 계산을 위한 파라미터 정의 및 스코어 계산 방식은 다음과 같다. 표 5와 6은 각각 SI(Score_Index)와 SL(Score_Length)을 구하기 위한 조건을 나타낸다. 표 7은 획 방향 코드 비교를 위한 테이블이다.

표 5. 획 인덱스 비교 테이블
Table 5. Stroke index comparison table.

획 인덱스 비교 : SI	
S_Index == R_Index	S_Index != R_Index
10	0

SI - Score_Index : 인덱스 비교 결과 값
SL-Score_Length : 길이 비교 결과 값
SC-Score_Curve : 곡률 비교 결과 값
SD-Score_Direction : 방향 코드 비교 결과 값
ST-Score_Total : 합산된 스코어

$$SC = |\text{Data Stroke Curvature} - \text{Standard Stroke Curvature}| \quad (5)$$

표 6. 획 길이 비교 테이블
Table 6. Stroke length comparison table.

획 길이 비교(가로축이 표준 획) : SL			
	0(점)	**1(짧은 획)**	**2(긴 획)**
0	10	5	0
1	5	10	2
2	0	2	10

표 7. 획 방향 코드 비교 테이블
Table 7. Stroke direction code comparison table.

획 방향 코드 비교(가로축이 표준 획) : SD								
	0	1	2	3	4	5	6	7
0	10	5	0	0	0	0	0	5
1	5	10	5	0	0	0	0	0
2	0	5	10	5	0	0	0	0
3	0	0	5	10	5	0	0	0
4	0	0	0	5	10	5	0	0
5	0	0	0	0	5	10	5	0
6	0	0	0	0	0	5	10	5
7	5	0	0	0	0	0	5	10

최종 획별 스코어 계산은 식 (6)을 따르며, 여기서 μ는 획 보조 스코어 가중치를 나타낸다.

$$ST = SI + \mu\{SL + (SD \cdot SC)\} \quad (6)$$

가상 획 특징 파라미터를 사용한 스코어 계산은 획 특징 파라미터를 사용한 스코어 계산 방법과 유사하며, 인덱스로서 가상 획 방향 코드를 사용하며, 곡률 특징 파라미터는 사용되지 않는다. 식(7)은 가상 획 특징 파라미터를 사용한 스코어 계산식이며, 여기서 ν는 가상 획 보조 스코어 가중치를 나타낸다.

$$ST = SD + \nu SL \quad (7)$$

영역 특징 파라미터는 위의 계산 결과에 따라 최종 후보가 선택되면, 후보 문자를 대상으로 비교하여 유사 형태의 후보 문자에 스코어를 추가시킨다. 이는 문자 획 필기의 순서가 비슷한 후보 중 전체적인 형태적 특징이 비슷한 후보의 스코어를 높이는 역할을 한다.

VI. 실험 및 결과

실험을 위하여 교육용 기초 한자 1800자에 대한 필기 문자 데이터베이스 5 세트를 수집하였다. 필기 방법은 정서체이며, 표준 획수와 표준 획순은 지정되지 않은 상태에서 필기하도록 요구된 문자 데이터베이스를 사용하였다.[9]

각각의 데이터베이스 세트는 동일한 PDA로 입력받았으며, 데이터 수집방식은 필기속도 및 필순을 저장할 수 있도록 일정 시간 간격으로 샘플링된 point들의 위치 값에 대한 시퀀스를 저장하였다. 필순을 저장하기 위하여 모든 획들은 PDA의 창의 펜을 누르는 순간부터 펜이 떨어지는 순간까지를 하나의 획으로 저장하여 모든 획을 시간 순서대로 저장하였다.

실험 방법은 최종 결과 문자와 필기 문자와의 인식률을 측정하였다. 인식 대상 필기 문자는 필기 오류로 인한 문자를 제외한 문자 셀에 한하여 측정하였다. 또한, PDA의 성능에 만족하기 위하여 문자당 평균 인식 속도를 측정하였다. 표 8은 실험 환경을 나타낸다.

표 8. 실험 환경
Table 8. Test environment.

	1차 실험(PC 환경)	2차 실험(PDA 환경)
CPU	AMD K6-2 400MHz	MIPS R4000(Max 150MHz)
OS	Window NT 2000	Window CE
Language	MFC	WinCE용 MFC

인식 속도의 측정은 데이터베이스에 등록된 단어의 point 시퀀스의 입력 완료 시간을 시작시간으로 측정하고, 인식 완료 시간을 마침 시간으로 측정하였다. 시간 측정 방식은 개발 환경(WinCE용 MFC)에서 제공하는 0.001초 단위의 SYSTEMTIME 값으로 측정하였다. 표 9는 실험 환경에 따른 평균 인식률과 인식 속도를 나타낸다.

표 9. 인식 결과
Table 9. Recognition results.

5인 DB 셀	평균 인식률	평균 인식 속도	
		1차 실험	2차 실험
JJH	92.32	0.009	0.164
KJY	93.59	0.009	0.164
WON	96.65	0.009	0.162
JYS	94.70	0.009	0.168
SKB	94.19	0.009	0.163
평균	94.29	0.0088 sec	0.164 sec

실험 결과에서 볼 수 있듯이 본 논문에서 구현한 한자 인식기는 유닛 재구성 방법을 사용하였기 때문에, 인식 대상 단어의 증가에 따른 연산량의 증가는 발생하지 않는다. 따라서, 좋은 인식률과 더불어 평균 인식 속도를 보장한다. 또한, 유닛 모델의 사용으로 표준 문자 모델 셀을 구성하는데 적은 메모리를 사용한다. 빠른 인식 속도와 적은 메모리의 사용은 기존 유닛의 조합으로 구성되는 문자 셀의 확장을 허용하며, 이는 전체 인식 속도 및 메모리 사용에 크게 기여하지 않는다.

본 논문에서 처리되지 않은 유사 문자에 대한 처리가 이루어지고, 체계적인 유닛 설정이 이루어진다면 전체 문자 셀의 편차를 수용할 수 있으며, 평균 인식률은 상당히 개선될 수 있다.

참 고 문 헌

[1] Zheng Luo and Chwan-Hwa Wu, "A Unit Decomposition Technique Using Fuzzy Logic for Real-Time Handwritten Chinese Character Recognition," *IEEE Transactions on Industrial Electronics*, Vol. 44, No. 6, December 1997.

[2] 김상균, 정종화, 김진욱, 김항준, "ART-1 신경망을 이용한 온라인 한자 인식," 전자공학회논문집, 제 33권, B편, 제 2호, 1996

[3] Hajime Nambu, Takenori Kawamata, Fuyuki Maruyama, and Fumio Yoda, "On-line Handwriting Chinese Character Recognition; Comparison and Improvement to Japanese Kanji Recognition," *Proceedings of the 14th International Conference on Pattern Recognition*, Vol. 2, pp. 1145-1149, 1998.

[4] Ju-Wei Chen and Suh-Yin Lee, "On-Line Handwriting Recognition of Chinese Characters via a Rule-Based Approach," *Proceedings of the 13th International Conference on Pattern Recognition*, Vol. 3, 1996.

[5] Bong-Kee Sin and Jin H. Kim, "Ligature Modeling for Online Cursive Script Recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 6, June 1997.

[6] 이성환 외, 문자 인식, 홍릉 과학 출판사, 1993

[7] Derek C. W. Pao, M. C. Sun, and Murphy C. H. Lam, "An Approximate String Matching Algorithm for On-Line Chinese Character Recognition," *Image & Vision Computing*, Vol. 15, No. 9, 1997.

[8] Chang-Keng Lin, Kuo-Sen Chou, Bor-Shenn Jeng, Chun-Hsi Shih, Tzu-Kai Su, and Tzu-I J. Fan, "a Knowledge Model Based On-line Recognition System," Proceedings of the 1992 International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, 1992.

[9] Wing-nin Leung and Kam-shun Cheng, "A Stroke-Order Free Chinese Handwriting Input System Based on Relative Stroke Positions and Back-Propagation Networks," *Official Program of the 1996 ACM Symposium on Applied Computing*, 1996.

저 자 소 개

陳 元(正會員)

1973년 10월 4일생. 1999년 2월 국민대학교 전자공학과(공학사). 2001년 2월 국민대학교 전자공학과(공학석사). 2000년 3월~현재 Nissi Media Korea에서 음성인식 응용프로그램 개발담당. 주 관심분야는 휴먼 인터페이스, 디지털 신호처리 등임

金 基 斗(正會員) 第 38券 TC編 第3號 參照.

現在 國民大學校 電子工學部 敎授