
Pocket PC기반의 효율적인 한글 정합 시스템 구현

박종민* · 조범준*

Implementation of an efficient Pocket PC- based Hangul Matching System

Jong-Min Park* · Beom-Joon Cho*

요 약

전자 잉크 데이터는 펜 기반 컴퓨터나 PDA(Personal Digital Assistants)등에서 자연스럽게 편리한 데이터 입력을 제공하기 위해 펜으로 입력한 데이터를 온라인 문자 인식기를 이용하여 아스키 문자로 변환하지 않고 스크립트 형태로 저장하는 데이터를 말한다. 전자 잉크 데이터를 사용하기 위해 가장 중요한 것 중 하나는 전자 잉크 데이터의 검색 문제이다. 본 연구에서는 전자 잉크 데이터를 획 특징 벡터 형태로 저장하고, 이를 이용해서 잉크 데이터를 검색하는 정합 알고리즘을 제안하고 구현하였다. 제안된 정합 알고리즘은 입력된 데이터를 곡률을 이용하여 기본획으로 분리하고 기본획의 종류를 결정한 다음 획 특징 벡터를 생성한다. 그리고 동적 프로그래밍 기법에 의해 획 특징 벡터의 거리값을 계산한다.

ABSTRACT

Electronic Ink is a stored data in the form of the handwritten text or the script without converting it into ASCII by handwritten recognition on the pen-based computers and Personal Digital Assistants(Pocket PC) for supporting natural and convenient data input. One of the most important issues is to search the electronic ink in order to use it. We proposed and implemented a script matching algorithm for the electronic ink. Proposed matching algorithm separated the input stroke into a set of primitive stroke using the curvature of the stroke curve. After determining the type of separated strokes, it produced a stroke feature vector. And then it calculated the distance between the stroke feature vector of input strokes and one of strokes in the database using the dynamic programming technique.

Key words

Pocket PC, Electronic Ink Data, Hangul, Matching

1. 서 론

휴대용 컴퓨터에서 전자 잉크 데이터(electronic ink data)의 처리와 인식을 위한 연구는 1990년대 초반부터 이루어져 다양한 연구 결과가 나왔다 [1,2]. 전자 잉크 데이터의 한글 인식을 위한 몇 가지 대략적인 잉크 정합 알고리즘 (approximate ink

matching algorithms)들이 제안되었으며[3,4] 대용량의 데이터 베이스에서 전자 잉크 데이터를 효율적으로 검색하기 위한 방법으로 은닉 마르코프 네트워크와 R-tree를 이용한 방법들이 제안되었다 [5,6]. 그런데 제안한 방법들은 영어 스크립트를 주된 입력 형태로 개발되었다. 펜으로 영어와 한글 스크립트를 작성할 때 영어는 연속된 원호의 합으

로 분해될 수 있는 글자인 반면 한글의 경우 획이 꺾이는 위치와 획의 방향이 영어에 비해 뚜렷하다는 차이점이 있다. 영어 스크립트 정합을 위해 제시된 알고리즘을 구현하여[4] 한글 스크립트에 적용한 결과 정합률이 매우 저조하게 나타났다. 따라서 영어 스크립트에 적용했던 정합 알고리즘을 한글 스크립트에 그대로 적용하는 것은 무의미하며 한글 스크립트 위주의 전자 잉크 데이터를 사용하기 위해서는 한글의 기하학적인 특성을 고려한 한글 정합 알고리즘의[7] 적용이 필요하다.

본 연구에서는 한글위주의 스크립트를 전자 잉크 데이터형태로 Pocket PC에서 사용하기 위한 한글 정합 알고리즘을 적용하여 구현하였다. 적용된 한글 정합 알고리즘은 전자 잉크 데이터를 스크립트 형태로 변환한 후 이를 동적 프로그래밍 기법을 사용하여 시스템 구현에 적용한다. Pocket PC의 하드웨어적인 제약을 고려하여 효율적인 속도를 보장하면서 인식률을 높이기 위해 기본획을 인식한 후, 획정보와 획간의 위치관계를 이용한 데이터의 값으로 변환하여 이를 CF메모리상에 있는 통계적 수치 데이터로 저장된 한글 데이터 값과 비교하여 한글을 인식할 수 있는 시스템 구현에 목적이 있다.

CF 메모리는 인식이 좋고 충격에 강한 장점과 가격이 비싼 단점이 있다.

II. 관련연구

2.1 전자 잉크 데이터 개요

전자 잉크 데이터란 펜으로 입력한 문자, 심볼, 그림 등 펜 스트로크(pen-strokes) 데이터를 의미한다. 사용자가 펜으로 입력한 데이터를 인식할 필요 없이 잉크 데이터 자체로 데이터 값으로 변환하여 이를 CF 메모리상의 한글 데이터 값과 비교하여 한글을 인식하는 시스템을 만들자는 것이다. 이 방법의 주된 장점은 사용자에게 입력 문자의 종류에 전혀 제한을 주지 않는다는 것이다. 문자 인식 시스템의 경우 숫자나 알파벳, 한글, 몇 개의 특수 문자만을 사용하도록 하고 있다. 하지만 잉크 데이터 자체를 처리하는 시스템에서는 이러한 제한을 들 필요가 없다. 물론 한글 전자 잉크 데이터 인식을 위해 패턴 정합과 연관된 문제들을 해결해야 되지만 기존의 문자 인식에 비해 처리 방법상에 속도의 향상 및 문자에 다양한 접목을 할 수 있다는 이점이 있다.

2.2 전자 잉크 데이터의 구성

전자 잉크 데이터는 펜으로 입력한 획들(strokes)의 집합이다. 획은 pen-down에서 pen-up 사이에 입력된 점들(points)의 순서 있는 집합으로 정의된다. 잉크 데이터 Ink는 사용자가 펜을 이용하여 입력한 획들 S_i 의 집합으로 다음과 같이 표현할 수 있다.

$$Ink = \langle S_j \rangle, 1 \leq j \leq n$$

$$S_i = \langle (x_{ij}, y_{ij}, t_{ij}) \rangle, 1 \leq i \leq m_i \quad (식1)$$

식 1에서 S_i 는 사용자가 입력한 i 번째 획을 의미하고 원소 x_{ij}, y_{ij}, t_{ij} 는 각각 S_i 의 j 번째 점의 x 좌표 값, y 좌표 값, 그리고 점이 입력되어진 시간 순서를 나타내며, M_i 는 S_i 를 구성하는 점들의 개수를 의미한다. 펜으로 데이터를 입력할 때 동일한 사람이 동일한 글자를 입력해도 두 번 이상 쓸 때 똑같이 쓸 수 없기 때문에 잉크 데이터의 정보는 달라지게 된다. 그러므로 잉크 데이터 인식이 정확히 일치하는 한글 데이터 인식은 불가능하다. 따라서 잉크 데이터 인식은 잉크 데이터의 대략적인 정합(approximate matching)을 통한 인식이라고 표현하는 것이 적절할 것이다. 전자 잉크 데이터의 인식 문제는 두 개의 전자 잉크 데이터 P(pattern), T(text)가 주어질 때 P가 T 내에 포함되어 있는지, 포함되어 있으면 어느 위치에 있는지를 결정해 주는 것이다. 고전적인 스트링 정합과 다른 점은 P와 T 사이에 완벽한 정합이 발생하지 않는다는 것이다. 펜으로 입력한 필기체는 동일한 글자라 하더라도 사용자에게 따라 필기하는 형태와 획 순서가 다를 수 있다. 한글 정합 알고리즘[7]은 데이터의 특성을 고려하여 필기자 종속(writer-dependant)인식 방법으로 시스템을 개발하였다. 필기자 종속 정합 알고리즘의 경우 필체가 다르면 동일한 글자에 대해서도 다른 모양의 객체로 간주하기 때문에 필체가 다르면 데이터 베이스 내의 동일한 글자를 쿼리(query)로 주더라도 원하는 정보를 찾을 수 없다. Pocket PC와 같은 휴대용 컴퓨터의 경우 개인전용 단말기라는 특성을 가지고 있으므로 필기자 종속 정합 알고리즘이 적절하며 이는 필기 형태가 다른 사람들에게 대해 보안의 기능도 제공할 수 있다. 이런 기능의 특성과 단점을 보완하여 필기자의 데이터 정보를 CF 메모리상에 데이터 값을 저장하여 이 값들을 학습 시키고 가공하면 하드웨어적인 단점을 보완하며 이들 시스템들의 효율적인 속도를 보장하고 효율적인 한

글 인식이 가능하다고 본다.

III. 제안된 한글 정합 시스템

3.1 한글 정합 시스템

본 연구에서는 한글 위주의 스크립트를 전자 잉크 데이터 형태로 Pocket Pc에서 사용하기 위한 한글 정합 알고리즘을 이용하여 시스템을 구현하였다. 한글 정합 알고리즘을 이용하여 잉크 데이터를 기본획 단위로 나눈 후 동적 프로그래밍 기법을 적용한다. Pocket PC의 하드웨어적인 제약을 고려하여 효율적인 속도를 보장하면서 인식률을 높일 수 있도록 CF 메모리상에 한글 수치 데이터를 저장하도록 고안되었다. 적용된 시스템은 Pocket PC상의 다양한 전자 잉크 데이터를 이용하여 한글 데이터 값으로 변환하여 주었을 때 원하는 CF 메모리상의 기존 한글 데이터 값을 검색하여 이를 한글로 정합해주는 시스템이다.

필기할 때 손의 떨림으로 인한 굴곡, 획의 끝 부분에서 삐침 등이 생길 수 있다. 이런 오류를 평활화, 삐침제거, 크기 정규화 등의 전처리 과정을 거쳐 보정한다.

$$X_i = \frac{(X_{i-3} + 3X_{i-2} + 6X_{i-1} + 7X_i + 6X_{i+1} + 3X_{i+2} + X_{i+3})}{27} \quad (\text{식 } 2)$$

식 (2)과 같이 전후 7점에 대해 가중치를 적용한 평균값으로 현재 점을 대체하여, 획의 굴곡을 완만하게 하는 평활화를 수행하였다. 필기속도에 의한 불규칙한 점들의 간격은 DDA(Dot Density Algorithm)와 같은 여과과정을 통해 일정하게 할 수 있었다. 본 연구에서는 방향벡터를 특징으로 사용했기에 그 외의 전처리 과정은 생략할 수 있었다.

3.2 특징 추출 과정

특징 추출 과정은 개선된 획들의 좌표 데이터로부터 자소 인식을 위한 정보로써 특징벡터, 가상벡터[11,12] 등을 추출한다. 입력 획은 필기체의 특성상 정형화되지 않은 길이와 각도를 가지며 여러 가지 형태의 홀림이 가미될 수 있다. 그러므로 입력 획의 좌표 데이터 열에서 불필요한 정보의 양을 줄이기 위해 특징점을 인식의 기본단위로 이용한다[8-10]. 그러나 필기 속도나 각도 변화에 따라 입력 획의 특정부분에 여러 개의 특징점이 존재할 수 있으므로 고리, 장식선 제거 및 인접한 특징점 제거 과정에 의해 주어진 거리 임계치 이내에 군집된 불

필요한 특징점들을 제거한다. 특징벡터와 가상벡터는 한글데이터베이스와의 정합을 위한 정보로써 특징벡터는 입력 획에 대한 순서 및 방향정보이며 가상벡터는 자소 내 획간의 위치관계나 자소 간 위치관계에 대한 정보이다[12,13]. 그러나 자소 인식 시 후보 자소들이 많이 발생하지 않으면 빠른 처리를 위해 가상벡터는 Tail-Head Vector 만을 이용한다. 또한 자소 분리와 인식을 위해서 획간의 위치 정보인 포함관계와 위치관계를 추출한다. 포함관계는 현재 획을 포함하는 최소외접 사각형과 이웃하는 획들의 최소외접 사각형간의 인접한 정도를 나타내는 정보로써 포함, 겹침, 분리로 나누어진다. (그림 1)는 입력문자 '강'에 대해서 포함관계를 추출하는 예를 보여준다. 점선사각형은 획을 포함하는 최소외접사각형이며 번호는 점선사각형안에 있는 획번호이다. 즉, 점선사각형 1은 획 'ㄱ'에 대한 전체 크기를 나타낸다. 점선사각형 2는 점선사각형 1에 포함되며 점선사각형 1,3,5는 서로 분리되며 점선사각형 3과 4는 겹쳐있음을 알 수 있다. 위치관계는 현재 획에 대한 무게중심과 모든 입력 획의 무게중심 간의 방향정보로써 자소내, 자소간 위치관계정보로 이용한다. (그림1)는 입력문자 '기'의 위치관계를 추출하는 예로써 추출된 위치관계정보는 문자를 구성하는 각 자소 간의 무게중심 들간의 방향정보 이다. 이러한 획간의 위치관계는, 입력된 획들을 하나의 곡선으로 생각할 때 획을 구성하는 점들에 대한 곡률을 계산하면 기본획으로 분리시킬 위치를 찾을 수 있다. 획을 구성하는 임의의 점에 대해 임계치 이상의 곡률을 가지면 그 점에서 획을 분리하였다.

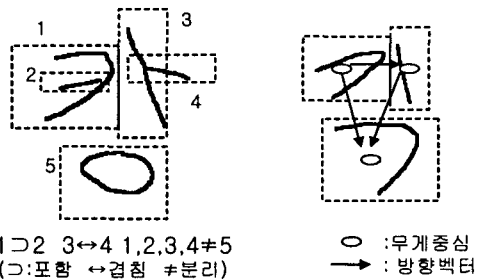


그림 1. 획간의 위치 관계 추출

3.3 자소 분리와 인식

자소 분리는 자소별 인식을 위해 반드시 필요한 과정으로, 본 연구에서는 자소 분리와 인식을 병행하여 처리한다. 자소 인식은 자소 분리에 의해 추출한 각 자소별 획 정보와 한글 데이터베이스에 저

장된 자소별 획 정보간의 정합에 의해 자소를 인식하며 인식된 각 자소들의 정보를 전자 잉크 데이터의 지정된 값으로 변경하여 문자를 생성한다.

자소 분리는 획간의 위치관계를 이용한 순차적 자소 분리와 CF 메모리상의 한글 데이터 값과의 정합에 의해 각 자소를 구성하는 획 수를 변경함으로써 인식을 수행하는 백트래킹 자소 분리와 CF 메모리상의 한글 데이터 값과의 정합에 의해 각 자소를 구성하는 획 수를 변경함으로써 인식을 수행하는 백트래킹 자소 분리를 이용한다. 순차적 자소 분리는 획간의 포함관계가 분리일 경우 다른 자소의 획이라고 인식하며 겹침이나 포함일 경우 같은 자소 내의 획으로 인식한다. 그러나 자소 내 획간의 포함관계가 분리일 경우 순차적 자소 분리는 정확한 자소 분리능력이 미흡하므로 오인식이나 미인식이 발생할 경우 백트래킹 자소 분리를 수행한다. 기존의 백트래킹 방법은 모든 과정을 재처리해야 하는 단점을 가진다. 그러나 제안된 백트래킹 자소 분리는 기존에 추출한 획 정보를 기반으로 입력문자가 가지는 전체 획의 개수를 각 자소별 인식 여부에 따라 최적의 획 개수로 분리하여 인식을 수행한다. (그림 2)은 이러한 변형된 백트래킹 자소 분리에 대한 알고리즘을 나타내며 (그림 3)은 자소 분리와 인식이 CF 메모리상의 한글 데이터 값과 병행하여 처리되는 과정을 보여준다. 본 연구는 입력문자나 필체의 형태 정보에 따라 차별적으로 자소 분리를 수행함으로써 빠른 처리속도로 Pocket PC 상의 제한적 하드웨어에 적합한 특성을 가지고 있다.

```

Step 1. 순차적 자소 분리
Step 2. 초성인식
if (인식 성공) step 3
    if CF 메모리상의 한글 데이터 값 비교
        else Step 2.
else 백트래킹 적용, 초성 재인식
Step 3. 중성 인식
if (인식 성공)
    if (중성이 없는 경우) 자소 분리 종료, 문자인식
    if CF 메모리상의 한글 데이터 값 비교
        else Step 2.
else step 3
else 백트래킹 적용, step 4
if (재인식성공) step 5
else 백트래킹 적용, step 3
Step 4. 중성인식
    if (인식 성공) 자소 분리 종료 문자 인식
    
```

```

else 백트래킹 적용, step 4
if(중성 재인식 성공) 자소 분리 종료,
if (중성 재인식 성공) 자소 분리 종료, 문자 인식
if CF 메모리상의 한글 데이터 값 비교
else Step 2.
else 백트래킹 적용, step 3
else 백트래킹 적용, step 3
    
```

그림 2. 변형된 백트래킹 자소 분리 알고리즘

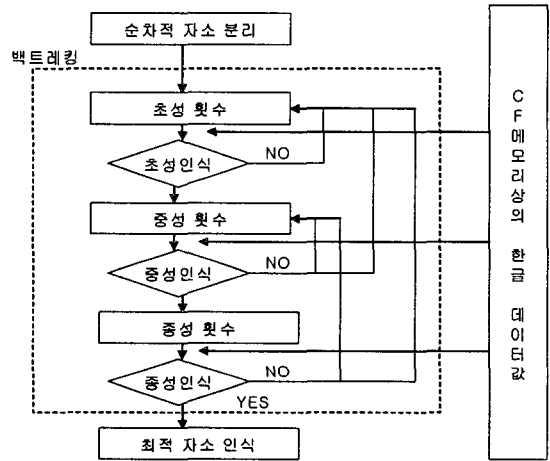


그림 3. 자소 분리와 인식 처리도

입력된 데이터들을 획의 곡률을 이용하여 기본 획 단위로 분리하는 과정을 거친다. 기본획은 Pocket PC에서 빠른 정합 속도를 보장하기 위해 한글의 기하학적인 특성을 고려하여 일곱 개를 지정하여 사용하였다. 다음 단계로 곡률에 의해 분리된 기본획들의 종류를 결정하여 획 특징 벡터 (stroke feature vector)를 생성한다. 획 특징 벡터는 입력된 데이터를 기본획 단위로 분리한 정보를 가지고 있는 벡터이다. 획 특징 벡터가 생성되면 동적 프로그래밍에 의해 쿼리의 획 특징 벡터와 데이터 베이스 내 잉크 데이터들에 대한 획 특징 벡터들 사이의 거리값을 계산하여 가장 적을 거리값을 가지는 데이터를 CF 메모리상의 데이터 수치값과 비교하여 정합 결과로 돌려준다.

자소 인식은 CF메모리상의 한글데이터베이스의 수치값의 동기화 과정으로 이루어지며 CF메모리상의 한글의 구조적 유형정보와 각 자소에 대한 다양한 사용자들의 획 정보 등을 지닌다. 한글의 구조적 유형정보는 중성 인식에서 사용되는 자소간 결합정보, 모음정보, 획간의 위치관계, 유형에 따른 자소 분류정보 등을 포함한다. 또한 자소별 인식을 위한 자소 모델은 각 자소에 대해서 일반적으로 사용하는 필체 정보 등을 지닌다. 하지만 심한 곡선

이나 자소간 홀림이 가미된 자소에 대해서는 정확한 정보를 추출하기 힘들기 때문에 직선, 완만한 곡선 정보, 다양한 방향의 필체 유형만을 이용한다. (그림 4)은 한글 데이터 잉크에 대해 계층적으로 인식하는 처리과정을 보여준다.

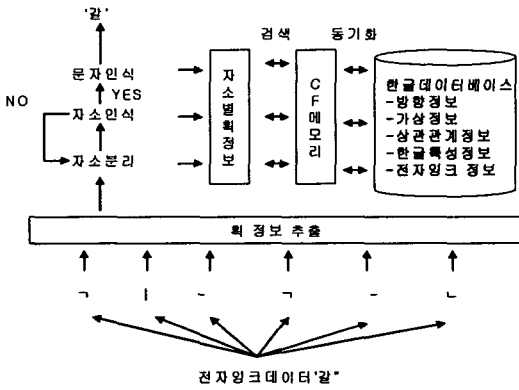


그림 4. 계층적 인식 과정

IV. 구현 및 고찰

4.1 실험 데이터 및 실험 모델

실험을 위한 데이터는 무작위로 추출한 전자 잉크 데이터 집합을 사용하였고 입력 형태는 한글 스크립트 형태를 사용하였다. 또한, CF메모리상의 한글 데이터를 만들기 위해 Visual C# 과 MS-SQL을 이용하였다. Pocket PC 상의 프로그램은 Visual Studio 2003과, Embedded Visual C++로 작성했으며 실험용 Pocket PC(Compaq ipaq 3550), SMART PONE 에뮬레이터를 이용해서 수행하였다.

(그림 5)는 CF 메모리상의 한글 데이터 값과 기존의 가공된 한글 데이터 베이스와의 한글 정보와의 비교를 통한 새로운 한글 데이터의 비교 및 가공을 통한 CF 메모리의 한글데이터 값의 동기화 모드이다.

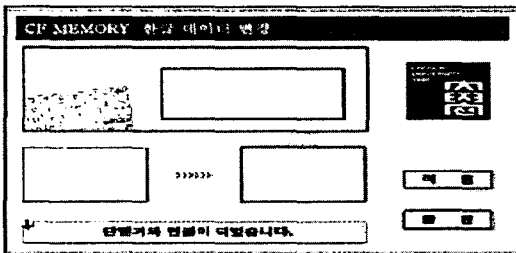


그림 5 CF 메모리 데이터 변경 실행 모드

세 가지 요소를 사용하여 정합률 R을 측정하였으며 그 모델은 다음과 같다.

$$R = m(n, t, r)$$

여기서 n은 CF 메모리상의 데이터 베이스에 있는 한글 스크립트의 값, t는 쿼리를 입력한 횟수, r 은 정합 결과의 순위(rank)를 의미한다. 쿼리 데이터 입력 횟수 t를 모델의 요소로 사용한 것은 원하는 정합 결과를 얻지 못할 때 다시 검색을 시도하도록 하기 위해서이다. M(900,2,1)은 900개의 한글 자소 데이터를 가진 한글 데이터 베이스에서 찾고자 하는 쿼리를 두 번 입력했을 때 원하는 결과를 1순위로 찾아줄 비율을 의미한다.

실험 모델에서 가장 이상적인 결과는 찾고자 하는 쿼리를 한번 입력했을 때 1순위로 찾아주는 것이다. 하지만 어떤 사람도 동일한 글자를 두 번 이상 쓸 때 똑같이 쓸 수 없다. 이런 펜 글씨의 특징으로 인해 대략적인 정합이 이루어지며 따라서 항상 1순위로 쿼리를 정합해 준다고 보장할 수 없다. 구현한 시스템은 한 페이지에 4순위까지의 정합 결과를 보여 주므로 찾고자 하는 데이터가 1순위에서 4순위까지의 순서를 가지면 정합이 성공적으로 이루어졌다고 여긴다. CF 메모리상의 데이터 베이스 내의 k번째 한글 데이터의 값에 대한 정합 성공 여부는 아래와 같다.

$$Hit(k, l) = \begin{cases} 1, & \text{if matchrank is from } l \text{ to } 5 \text{ for } k \\ 0, & \text{otherwise} \end{cases}$$

식 (3)
식 (4)

식 3에서 Hit(k, l)의 값이 0일 때는 정합이 실패한 경우이므로 원하는 데이터를 찾기 위해 쿼리를 다시 입력할 수 있다. 동일한 쿼리 데이터를 찾기 위해 t번 입력한다면 이때의 정합 성공 여부는 식 4와 같다.

4.2 데이터의 개수와 정합률

CF 메모리상의 데이터 베이스의 크기에 따른 정합률(R)을 실험하기 위해 데이터 개수가 100, 200, 300, 400, 500, 700, 900인 데이터 집합을 가공 준비하였다. 모든 데이터에 대해 실험하기 힘들므로 데이터 집합에서 각각 50%, 70%에 해당하는 샘플을 랜덤하게 추출하여 세 개의 실험 데이터들의 집합을 만들었다.

표 1은 세 개의 데이터 집합에 대한 정합률을 보

여주며 그림 6은 샘플 크기가 전체 데이터의 50% 일 때의 평균 정합률을 보여준다. 실험 모델에서 t 값이 1일 때와 3일 때, 즉 찾고자 하는 쿼리를 한번 입력했을 때와 정합이 실패했을 경우 세 번까지 동일한 쿼리를 입력했을 때 나타난 정합 결과를 보여 준다. 데이터 베이스의 개수가 300개 이하일 경우 t의 값에 관계없이 100%의 정합률을 보여 주었다. 하지만 개수가 많아지면 정합률이 조금씩 떨어지는 것을 볼 수 있다. 정합률이 t=1일 때와 t=3일 때를 비교해 보면 평균적으로 약 1% 차이가 났다. 900개의 데이터 집합을 기준으로 볼 때 한번 쿼리를 입력했을 때 정합이 성공할 비율은 96.3%이고 정합이 실패할 경우 세 번까지 쿼리를 입력하면 97.7%로 향상되었다. 이는 입력할 때마다 조금씩 달라지는 펜 데이터의 특징 때문이다.

표 1의 데이터 집합 종류에 따른 정합률을 비교해 보면 t=1일 때 데이터 2인 경우가 데이터 1인 경우에 비해 정합률이 상대적으로 떨어진다. 이것은 샘플로 선택된 데이터 중에서 사용자의 일반적인 필체와 다른 데이터를 상대적으로 많이 포함하고 있기 때문이다. 따라서 CF 메모리상의 데이터 베이스에 데이터들을 저장할 때 사용자의 일반적인 필기 형태로 주의를 기울여 입력한다면 정합률이 그림 6에서 보여주는 것보다 더 높아질 것이다. 그림 7은 샘플 크기가 70%일 때의 결과이다. 샘플 크기가 50%인 데이터와 비교해 볼 때 정합률은 차이가 나지 않는다. 이것은 표본의 크기가 정합률에 영향을 미치지 않음을 보여준다. 따라서 전체 데이터에 대한 정합률이 샘플 데이터에 대한 정합률과 비슷할 것으로 짐작할 수 있다.

표 1. 세 종류의 데이터 집합에 대한 정합률과 평균 정합률

데이터집합	100	200	300	400	500	600	700
T=1	데이터 1	100	100	100	99	98	98
	데이터 2	100	100	98	98	97	96
	데이터 3	100	100	100	98	99	97
	평균	100	100	99.3	98.3	98	97
T=3	데이터 1	100	100	100	100	100	99
	데이터 2	100	100	100	99	97	97
	데이터 3	100	100	100	100	100	98
	평균	100	100	100	99.7	98.7	98

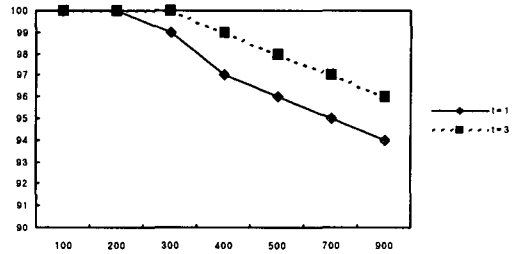


그림 6. 데이터 개수와 쿼리 입력 횟수에 따른 평균 정합률(샘플 크기 50%)

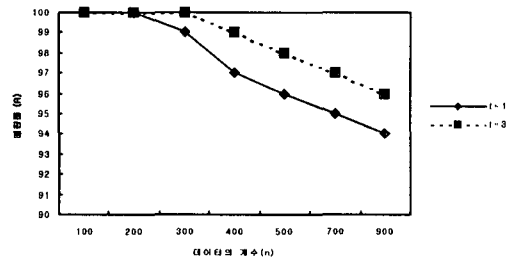


그림 7. 데이터 개수와 쿼리 입력 횟수에 따른 평균 정합률(샘플크기:70%)

V. 결 론

본 연구에서는 Pocket PC에서 펜으로 입력한 스크립트 데이터를 이용하여 전자 잉크 형태로 저장 및 인식하기 위한 한글 정합 시스템을 제안하고 구현하였다. 제안된 한글 정합 시스템은 전처리 과정과 함께 획의 곡률을 계산하여 기본 획으로 분리하는 과정을 거친다. 그리고 기본획 종류 결정에 의해 획 특징 벡터를 생성하고 이를 이용한 동적 프로그래밍 기법에 의해 거리값을 계산한다. 거리값을 계산할 때 사용한 편집 연산은 삭제, 삽입, 교환, 연산 외 한글의 특징을 고려한 결합, 분리 연산을 사용하였다. 여기서 만들어진 값을 이용하여 CF 메모리상의 한글 수치 데이터 값을 비교하여 한글을 인식하여 이를 시스템에 적용하였다.

본 연구에서 사용한 정합 알고리즘은 데이터베이스의 개수가 300개일 때에도 한글 스크립트인 경우 98%에 가까운 정합률을 보여 주었다. 또한, 데이터베이스의 개수가 300개일 때 평균 정합 속도가 0.8초를 보였는데, 이는 데스크 탑 컴퓨터와 비교하여 하드웨어적인 제약을 가지고 있는

Pocket PC의 환경을 고려할 때 매우 빠른 속도라 할 수 있다. 다음으로, 필기자 종속 정합 알고리즘이 가지는 장점으로써 필체가 다른 사람에 대해 보안 기능을 제공해 줄 수 있다.

향후 연구 과제는 첫째, CF 메모리상의 한글 데이터베이스의 효율적인 구축을 위해 데이터의 가공의 문제점을 어떻게 효율화 시켜 시스템에 적용하는 부분이다. 둘째, Pocket PC 상의 사용자의 전자 잉크 데이터를 가지고 와서 각각의 사용자의 특성을 파악하고 이를 얼마나 효율적으로 CF 메모리상의 데이터베이스 값으로 동기화 시키는 부분이다. 마지막으로 스크립트를 입력할 때 문자를 구분하지 않고 입력할 때 정합률을 향상시키기 위한 방법을 고안하는 것이다.

- [7] 조미경, 조환규, "PDA를 위한 한글 스크립트 매칭 알고리즘," 한국정보과학회 논문지 제 28권 10호, 2002.
- [8] 이성환, 문자인식 이론과 실제(I)(II), 홍릉과학출판사, 1993.
- [9] 김대수, 신경망 이론과응용(I)(II), 하이테크정보, 1993.
- [10] 양종원, "TDNN 신경망을 이용한 온라인 한글 필기체 인식에 관한 연구", 전남대학교 석사학위논문, 1997.
- [11] 김항미, "순환신경망을 이용한 온라인 문자 인식", 연세대학교 석사학위논문, 1995.
- [12] 정기철, 김상균, 이종국, 김항준, "자소 단위의 온라인 흘림체 한글 인식", 전자공학회 논문지 제33권, 제9호, pp.124-134, 1996.

참고문헌

- [1] W. Aref, D. Barbara, D. Lopresti. and A. Tomkins. "Ink as a first-class datatype in Multimedia databases," Multimedia Database, Springer-Verlag,1995.
- [2] Walid G. Aref, Ibrahim Kamel, and Daniel P. Lopresti, "On Handling Electronic Ink," ACM Computing Surveys, Vol. 27, No 4. pp. 564-567,1995.
- [3] Lopresti, D. and Tomkins, A "On the searchability of electronic ink," In Proceedings of the International Workshop Front. in Handwriting Recognition, pp. 156-166, 1994.
- [4] D.P. Lopresti and A. Tomkins, "Approximate matching of hand-drawn Pictogram," In Proceeding of the Third International Workshop on Frontiers in Handwriting Recognition, pp. 102-111, 1993
- [5] Walid Aref and Daniel Barbara, "Supporting Electronic Ink Database," Information Systems, Vol. 24, No. 4, pp. 303-326, 1999.
- [6] Ibrahim Kamel and Daniel Barbara, "Retrieving Electronic Ink by Content," IEEE Proceedings of International Workshop on Multimedia Database Management Systems, pp. 54-61, 1996.

저자소개

박종민(Jong-Min Park)



1988년 조선대학교 전자계산 전공(공학석사)
 2004년 조선대학교 컴퓨터공학 전공(박사과정수료)
 ※관심분야 : 생체인식, 패턴인식, 인공지능, 정보보호 및 보안

조범준(Beom-Joon Cho)



1980년 조선대학교 전기공학과 (공학사)
 1988년 한양대학교 전기공학 과 (공학박사)
 2004년 한국과학기술원 전자전산학과 전산학전공(공학박사)
 1980년~현재 조선대학교 전자정보공과대학 컴퓨터공학부 교수
 1993년~1997년 조선대학교 전자계산소장
 2002년~현재 한국멀티미디어학회 부회장
 2000년~2002년 조선대학교 전자정보공과대학 학장
 ※ 관심분야 : 인공지능, 패턴인식, 뉴로컴퓨터