모델 그래프를 이용한 빠른 필기 인식 방법

오세창*

Fast Handwriting Recognition Using Model Graph

Se-Chang Oh*

요 약

많은 문자인식 문제에서 인식 속도를 높이기 위하여 대분류 방법이 사용된다. 이 경우 처음에 부류를 잘못 선택하면 대분류에 의해 돌이킬 수 없는 오류가 발생한다. 이러한 위험을 줄이기 위하여 각 모델을 여러 부류에 중복시키는 방법이 사용되지만, 이 또한 대분류 오류를 완전히 배제할 수는 없다. 본 논문에서는 오류를 증가시키지 않으면서도 전체 모델 중 일부를 선택적으로 정합함으로써 인식 속도를 높이기 위한 방법을 제안한다. 이 방법은 모델 간의 유사도를 이용하여 그래프를 구성하고, 이 그래프 상에서 특정 모델을 시점으로 그래프 탐색을 진행함으로써, 입력 패턴과 유사도가 떨어지는 모델들과의 불필요한 정합을 줄인다. 본 논문에서는 이 방법을 숫자와 영어 대소문자를 대상으로 하는 온라인 필기 문자 인식 문제에 적용하였다. 실험에서 입력 패턴을 모든 모델과 정합하는 기본적인 방법과 제안하는 방법을 비교해 보았다. 그 결과 모델 그래프의 진출 차수와 탐색 과정에서 유지하는 후보의 개수를 적절히 조절함으로써 기본적인 방법과 동일한 인식률을 얻었으며, 인식 속도는 2.45배의 증가를 보였다.

ABSTRACT

Rough classification methods are used to improving the recognition speed in many character recognition problems. In this case, some irreversible result can occur by an error in rough classification. Methods for duplicating each model in several classes are used in order to reduce this risk. But the errors by rough classfication can not be completely ruled out by these methods. In this paper, an recognition method is proposed to increase speed that matches models selectively without any increase in error. This method constructs a model graph using similarity between models. Then a search process begins from a particular point in the model graph. In this process, matching of unnecessary models are reduced that are not similar to the input pattern. In this paper, the proposed method is applied to the recognition problem of handwriting numbers and upper/lower cases of English alphabets. In the experiments, the proposed method was compared with the basic method that matches all models with input pattern. As a result, the same recognition rate, which has shown as the basic method, was obtained by controlling the out-degree of the model graph and the number of maintaining candidates during the search process thereby being increased the recognition speed to 2.45 times.

키워드

모바일 컴퓨팅, 패턴인식, 필기인식, 모델 그래프

Key word

Mobile Computing, Pattern Recognition, Handwriting Recognition, Model Graph

* 정회원 : 세종사이버대학교 정보통신학과 (scoh713@gmail.com)

접수일자 : 2011. 10. 06
심사완료일자 : 2012. 05. 01

Open Access http://dx.doi.org/10.6109/jkiice.2012.16.5.892

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ⅰ. 서 론

최근 스마트폰과 테블렛 PC 등 모바일 기기의 보급이 급격히 늘어나면서 수많은 응용 프로그램들이 개발되고 사용되고 있다. 이러한 모바일 기기에서 정보 입력 수단으로 사용되는 스크린 키보드는 작은 화면에서 키를 누르다 보니 많은 오타를 야기한다. 또한 키보드에 익숙지 않은 초보 사용자에게는 매우 부자연스러운 입력 수단일 수밖에 없다. 근본적으로 이러한 문제는 전자 펜을 사용하는 필기 인식 기술에 의해서 해결될 수 있다. 필기 인식 분야의 연구는 1990년대 말까지 활발히 진행되었다. 그러나 그 당시 모바일 기기의 낮은 성능으로 인해서 [1, 2]와 같은 고급 필기인식 기술의 적용이 어려웠다. 이러한 성능의 문제를 해결하기 위해 1990년대 중반에 개발된 Grffiti라는 인식기는 각각의 영문자를 한 획으로 쓰면서 서로 혼동되지 않도록 독특한 모양으로 디자인되었다. 이 방법을 사용하면 계산 자원을 적게 쓰면서도 거의 100%의 정확도를 얻을 수 있었다 [3]. 최근에는 이 기술을 적용한 스마트폰용 앱도 등장하였다 [4]. 그러나 이 방법은 사용자가 새로 디자인된 문자의 필기 방법을 익혀야 하므로 자연스럽지 않은 방법이다.

자연스러운 필기 방법을 보장하면서 인식 속도를 개선하기 위한 방법으로 문자 단위의 인식에서 대 분류 방법을 적용해 볼 수 있다. 즉, 기존의 방법에서는 입력된 패턴을 모든 문자 모델과 비교한다. 그러나 대 분류 방법을 적용하면 부류를 먼저 결정하고, 그 부류에 속한 모델들 만을 대상으로 비교함으로써 계산 량을 줄여 인식 속도를 획기적으로 개선할 수 있다. 문자 인식 분야에 적용된 대 분류 방법으로 가장 기본적인 방법은 먼저 군집화 알고리즘을 적용하여 문자 모델들을 군집화함으로써 각 모델이 하나의 부류에 속하도록 만든다 [5, 6]. 문자 인식 과정은 두 단계로 나뉘는데, 첫 번째 단계인 대 분류에서는 입력 패턴을 각 부류의 대표 특징 벡터와 비교하여 부류를 결정하고, 두 번째 단계인 소분류에서는 정해진 부류에 속한 모델들과 비교하여 최종적인 인식 결과를 얻는다. 이 방법의 근본적인 문제점은 첫 번째 단계인 대 분류에서 오류가 발생하면, 복구가 불가능하다는 점이다. 이러한 문제를 극복하기 위해서 비슷한 부류에 문자 모델을 중복해서 배치하는 방법이 사용된다. 예를 들어 [7]에서는 1차적으로 K-means 알고리즘을 사용하여 문자 모델의 중복 없이 부류를 정하고, 이 부류 정보를 사용해서 코호넨 네트워크를 학습시킴으로써 중복을 허용하도록 부류를 재조정 한다. 또한 [8]에서는 대 분류 과정이 트리 분류기를 사용하여 후보의 수가 충분히 줄어들 때까지 단계적으로 진행된다. 이 때 각 말단 노드는 하나의 부류를 나타내는데, 이들은 문자 모델들을 학습에 의해 중복하여 포함시킴으로써 대 분류에 의한 오류 문제를 보완한다. 그러나 [7, 8]과 같이 중복을 허용하여 대 분류를 하더라도 입력 패턴에 변형이 심해지면 대 분류 오류를 피할 수 없는 점이 이 방법의 근본적인 문제이다.

이러한 문제를 해결하기 위해서 본 논문에서는 모델 그래프를 이용한 인식 방법을 제안한다. 이 방법은 문자 모델 간의 유사도를 근거로 모델 그래프를 구성하는 방법과 이 모델 그래프 상에서 입력 패턴과 가장 비슷한 모델을 탐색하는 방법으로 구성된다.

즉, 이 방법은 문자 인식 문제를 풀기위해 대 분류 방법에서 사용하는 트리 상에서의 탐색 방법 대신에 그래프 상에서의 탐색 방법을 적용함으로써 비교 대상을 효과적으로 줄이면서도 탐색 오류를 배제할 수 있다. 따라서 이 방법을 사용하면 인식 속도를 향상시키면서도 모든 모델을 입력 패턴과 비교하는 기본적인 방법과 동일한 인식률을 얻을 수 있다.

Ⅱ. 모델 그래프를 이용한 문자 인식 방법

본 논문에서 제안하는 모델 그래프는 문자 모델을 노드로 하고 문자 모델들 간의 유사도를 간선 위에 표시하는 네트워크이다. 또한 문제를 단순화시키기 위해서 진출 차수가 d인 방향성 그래프로 모델 그래프를 구성하였다. 이 그래프에서는 각 모델로부터 유사도가 높은 d개의 모델을 선택하여 연결하고, 주어진 입력 패턴과 가장 유사도가 높은 모델을 이 그래프 상에서 탐색하는 문제로 문자 인식 문제를 형식화 한다. 이 때 탐색 과정이 점차 입력 패턴과 유사도가 높은 방향으로 진행되도록 조절하기 위해 후보 목록을 사용한 탐색 방법을 고안하였다.

이렇게 함으로써 대 분류 방식과 같이 일부 모델만 비교하고도 모든 모델을 다 비교하는 기본적인 인식 방법과 동일한 인식 결과를 얻을 수 있다. 또한 이 방법은 대

분류 방법과 달리 처음에 탐색의 시작 위치를 잘못 선택하더라도 인식률에 영향을 미치지 않는다. 이는 모델 그래프 상에 모든 모델들이 연결되어 있고 이 연결 구조를 따라가면 모든 모델에 도달할 수 있기 때문이다.

따라서 모델 그래프를 이용한 문자인식 방법은 모델 그래프를 구성하는 부분과 탐색의 시작점을 선택하는 부분 그리고 이 그래프를 이용해서 탐색하는 부분으로 나뉜다.

2.1. 모델 그래프의 구성

모델 그래프를 만들기 위해서는 훈련에 의해 구해진 문자 모델 m_i와 m_j 간의 유사도 $Sim(m_i, m_j)$를 계산해야 한다. 이 유사도의 계산은 입력 패턴을 모델과 비교하는 것과 동일한 방법을 사용하며, 3.4절에서 설명된 DTW 알고리즘에 의해서 계산된다. 각 모델로부터 유사도가 높은 d개의 모델을 선택하여 연결함으로써 식 (1)과 같은 모델 그래프가 완성된다. 이 식에서 N은 노드의 집합 즉, 모델 m_i로 이루어진 집합을 그리고 E는 간선 e_{ij} 즉, 모델 간의 연결선으로 이루어진 집합을 가리킨다. 또한 i, j는 모델의 인덱스이며, n은 전체 모델의 개수이다.

$$N = \{m_i \mid 1 \le i \le n\}$$
$$E = \{e_{ij} \mid m_i\text{를 기준으로 } Sim(m_i, m_j)\text{가 d번째안에드는 } m_j\} \quad (1)$$

이 때 d가 너무 크면 거의 모든 모델이 직접 연결되어 대분류를 하지 않은 것과 같이 탐색 효율이 떨어진다. 즉 d가 커짐에 따라 탐색 과정에서 입력 패턴과 정합되는 모델의 수가 늘어난다. 반면에 d가 너무 작으면 연결되지 않고 고립된 모델이 발생하여 탐색 과정에서 도달할 수 없는 모델이 발생하고 이는 인식률 저하를 초래한다.

2.2. 탐색의 시작점 선택

이렇게 구성된 모델 그래프에서 탐색을 하기 위해서는 탐색의 시작점 m_s를 결정해야 한다. 이 시작점을 어떻게 선택하는가에 따라 탐색 속도가 결정되는데, 모델 그래프 상에서 다른 모델들로의 도달 거리가 가장 짧은 모델로 정하는 것이 유리하다. 이는 그래프 탐색 과정에서 방문한 노드를 지나간 경로를 따라 연결하면 트리 형태가 되는데, 모든 단말 노드가 루트 노드로부터 같은 거리에 있는 균형 잡힌 트리가 평균적인 탐색 시간이 가장 짧은 것과 같은 원리이다. 이를 위해 본 논문에서는 식 (2)와 같이 전체 모델들 중 다른 모델들과의 유사도의 합이 가장 큰 모델 ms를 시작점으로 선택하였다. 이 식에서 i, j는 모델의 인덱스이다.

$$m_s = \arg\max_i \sum_j Sim(m_i, m_j) \quad (2)$$

2.3. 모델 그래프의 탐색

주어진 모델 그래프에서의 탐색은 그림 1과 같이 진행된다. 먼저 ①에서 시작점에 해당하는 모델 m_s를 방문하고 DTW 알고리즘을 사용해 입력 패턴 I와 정합한다. ②에서는 이 결과로 구해진 유사도 값과 함께 m_s를 후보 목록에 추가하여 목록을 갱신한다. 다음으로 ③은 모델 그래프 상에서 후보들을 한 단계 확장하여 보다 유사도가 높은 모델을 찾아 가는 과정인데, 이를 위해 먼저 후보 목록의 상위에 있는 후보부터 시작하여 직접 연결되어 있으면서 아직 방문하지 않은 모델을 찾는다. 그림 1과 같은 경우 가장 상위에 있는 후보 m_a와 직접 연결된 모델 m_b, m_c, m_d 중 이미 방문한 m_b를 제외한 m_c, m_d가 이에 해당한다. 여기에서 구한 새로운 모델들은 방문한 것으로 표시한 후 ④에서 입력 패턴 I와 정합하고, ②로 넘어간다. 만약 모든 후보에 대해서 이러한 모델들을 찾지 못한 경우에는 현재의 후보 목록을 반환하고 인식 과정을 마친다.

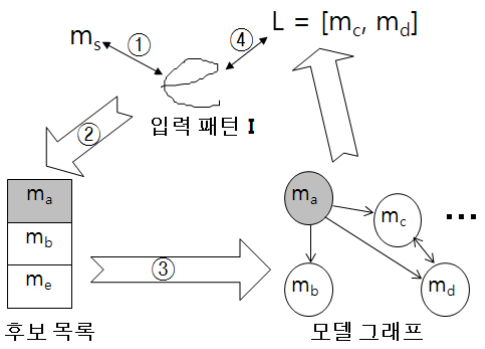

그림 1. 모델 그래프 탐색 과정
Fig. 1 Search Process on Model Graph

이 과정을 알고리즘으로 정리하면 다음과 같다.

Step 1. 모델 ms를 방문하고, 입력 패턴과 정합
Step 2. 전 단계에서 정합한 모델들을 추가하여 후보 목록 갱신
Step 3. 모델 그래프 상에서의 확장
  Step 3.1. 후보 목록의 인덱스 i를 0으로 설정
  Step 3.2. i번째 후보에 모델 mj가 있을 때 ejk가 존재하면서 아직 방문하지 않은 모델 mk를 각각 방문하고 이들을 모아 목록 L을 구함
  Step 3.3. 목록 L이 비어 있고 i가 마지막 후보의 인덱스가 아니면 i를 1 증가시키고 Step 3.2로 이동
Step 4. 목록 L이 비어있지 않으면 이에 포함된 각 모델에 대해서 입력 패턴과 정합하고 Step 2로 이동
Step 5. 후보 목록을 반환

Ⅲ. 문자 인식 과정

다음 그림은 문자 인식을 위한 전체 처리 과정을 나타낸다.

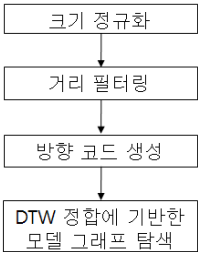

그림 2. 문자 인식 과정
Fig. 2 Character Recognition Process

3.1. 크기 정규화

스마트폰의 터치 스크린에서 들어온 입력 데이터는 점들의 리스트이다. 필기자에 따라 글자 크기가 다르기 때문에 모델과 정합하기 전에 먼저 크기 정규화를 해야 한다.

크기 정규화는 문자 단위로 이루어진다. 즉 하나의 문자를 이루는 획들의 모임으로부터 일정한 크기의 데이터를 얻고자 하는 것이 크기 정규화의 목적이다. 본 논문에서 사용한 방법은 그림 3과 같이 한 문자에 해당하는 부분의 바운딩 박스에서 대각선 길이를 일정한 비율로 나눈 값을 거리 필터링을 위한 단위 거리 delta로 정하는 방법을 사용하였다.

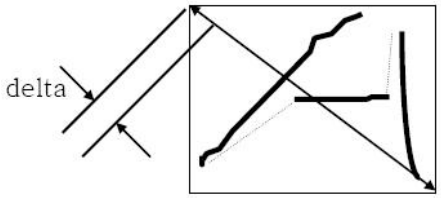

그림 3. 크기 정규화
Fig. 3 Size Normalization

3.2. 거리 필터링

획을 구성하는 점들은 필기 속도에 따라 배열 간격이 달라질 수 있다. 거리 필터링 과정에서는 이러한 점들의 간격을 일정하게 만들기 위해 등 간격으로 리샘플링 한다. 즉, 그림 4에서와 같이 한 점(P1)에서 일정한 거리 delta 안에 존재하는 점들은 제거하고, delta 만큼 떨어진 위치에 점이 없는 경우 새로운 점(P2)을 보간법을 사용하여 추가한다. 다시 P2를 기준으로 단위 거리 delta로 필터링 하고 이러한 과정을 획의 시작부터 끝까지 진행한다.

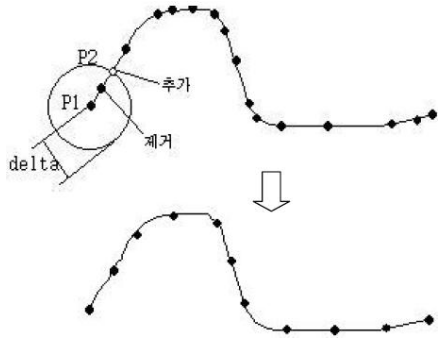

그림 4. 거리 필터링
Fig. 4 Distance Filtering

3.3. 방향 코드 생성

거리 필터링 과정을 통해 일정한 간격으로 배치된 점들을 사용하여 방향 코드를 생성한다. 즉 각 점에서 다음 점으로의 방향을 양자화하여 코드 값을 얻는다. 여기에서는 그림 5와 같이 펜을 그으면서 이동한 일반 획 성분과 펜을 들고 이동한 연결 획 [9] 성분에 대해 각각 36 방향으로 나누는 양자화 방법을 사용한다.

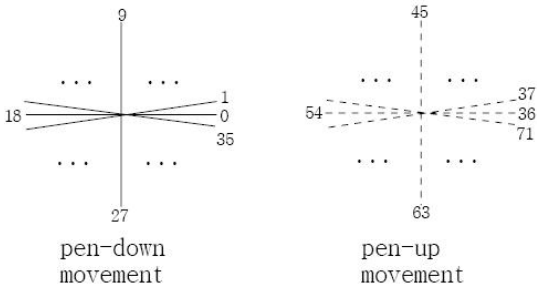

그림 5. 방향 코드
Fig. 5 Direction Code

3.4. DTW 정합에 기반한 모델 그래프 탐색

문자 정합은 방향 코드로 엔코딩된 입력 패턴과 역시 하나의 엔코딩된 데이터인 문자 모델 간의 탄력적 정합에 의해서 이루어진다. 여기에서 사용되는 정합 방법은 길이가 다르며 변형된 두 패턴을 정합하기 위한 DTW (Dynamic Time Warping) 알고리즘이다[10]. 이 방법은 비교하는 두 패턴을 각각 가로와 세로 방향으로 놓았을 때, 대응점을 대각선 방향으로 확장하면서 데이터를 이루는 각 점 간의 거리를 계산하고 대각선 방향으로 정합이 안 되고 조금씩 어긋나게 정합될 경우 일정한 벌점을 적용하면서 전체적으로 거리를 최소화하는 정합을 찾는 방법이다. 이 방법은 현재 다양한 문제에 응용되고 있다[11, 12].

입력 패턴과 모델의 유사도는 DPW 알고리즘에 의해서 구하고, 이 결과를 사용해서 앞에서 설명한 모델 그래프 탐색 방법으로 입력 패턴과 가장 비슷한 최적의 모델을 탐색한다.

Ⅳ. 실험 결과

실험은 본 논문에서 제안하는 방법이 입력 패턴을 모든 모델들과 비교하는 기본적인 방법에 비해 빠르고, 대분류 방법에서와 같은 인식률의 저하가 없다는 것을 입증하는 것을 목적으로 한다. 따라서 먼저 기본적인 인식 방법의 성능을 분석하고, 제안한 모델 그래프 탐색 방법의 성능을 분석해 비교한다. 이 때 모델 그래프 탐색 방법은 몇 가지 중요한 파라미터에 의해서 성능이 좌우되므로 실험을 통해 적당한 파라미터 값을 결정하는 것이 필요하다.

실험에서 사용된 문자 모델은 숫자와 영어 대소문자를 포함한 128개이다. 이 모델의 훈련은 5명으로부터 얻은 문자 단위의 필기 샘플 중 다양한 패턴이 포함되도록 선택하여 구성하였다. 또한 테스트 데이터는 앞에 5명을 포함해 7명으로부터 얻은 별도의 249개 문자 샘플들로 구성하였다. 사용된 모든 필기 샘플들은 감압식의 보급형 와콤 타블렛을 사용해 얻었다. 또한 인식을 위한 프로그램은 윈도우즈 환경에서 비주얼 C++로 구현되었으며, 이 프로그램을 사용해서 인식 결과는 물론이고 결과 분석에 필요한 각종 통계 자료도 같이 얻었다.

먼저 기본적인 방법에서는 평균 문자 정합 횟수와 후보 별 인식률을 비교한다. 결과는 표 1과 같은데, 문자 당 평균 정합 횟수는 항상 문자 모델의 개수인 128개이다. 즉, 모든 모델과 정합을 함으로써 인식 결과를 얻는다. 인식률은 첫 번째 후보가 88.76% 세 번째 후보까지가 100%이다.

표 1. 기본적인 인식 방법의 성능
Table. 1 Performance of Basic Method

평균 문자 정합 횟수	128
첫 번째 후보의 인식률	88.76%
세 번째 후보까지의 인식률	100%

본 논문에서 제안하는 그래프 탐색 방법은 탐색 과정에서 유지할 문자 후보의 개수와 그래프에서 각 노드의 진출 차수 d의 크기에 따라 성능이 달라진다. 따라서 인식률의 저하 없이 평균 문자 정합 횟수를 최소화 하도록 이 두 가지 값을 정하는 것이 필요하다. 실험에서는 이를 위해 후보의 개수를 1에서 5까지 변화시키고, d 값은 8부터 24까지 변화시키면서 문자정합 횟수와 인식률을 구해보았다. 그림 6은 평균 문자 정합 횟수를 나타내는 결과로 x, z 축은 각각 d 값, 문자 후보 수를 나타낸다. 이 그

래프에서 각 곡선에 표시된 숫자는 후보의 개수이다. 마찬가지로 그림 7은 첫 번째 후보의 인식률을 나타내고, 그림 8은 세 번째 후보까지의 인식률을 나타낸다.

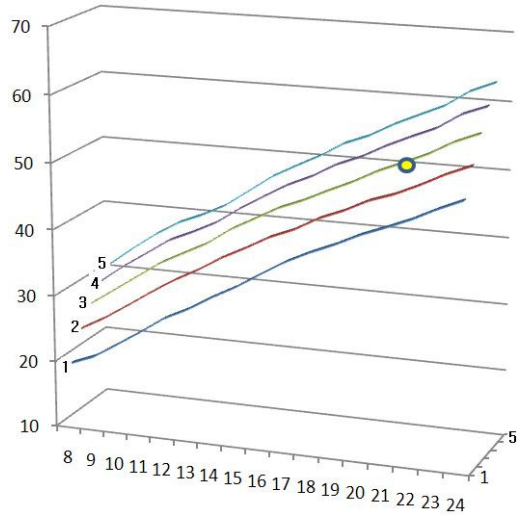

그림 6. 그래프 탐색에서 평균 문자 정합 횟수
Fig. 6 Average Numeber of Character Matchings in Graph Search

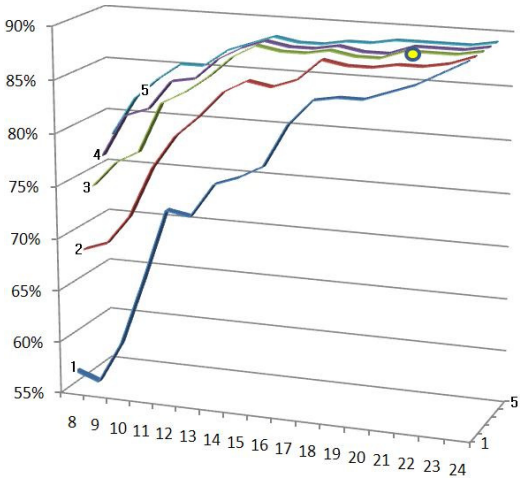

그림 7. 그래프 탐색에서 첫 번째 후보 인식률
Fig. 7 Correctness of Best One Candidate in Graph Search

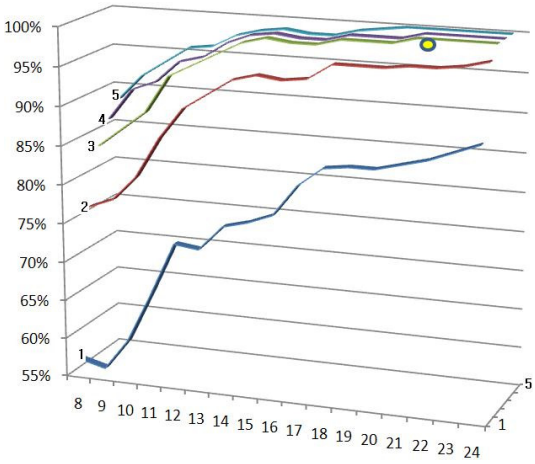

그림 8. 그래프 탐색에서 세 번째 후보까지의 인식률
Fig. 8 Correctness of Best Three Candidates in Graph Search

위의 결과를 보면 첫 번째 후보와 세 번째 후보까지의 인식률이 각각 88.76%와 100%로 그래프 탐색 방법을 적용하기 전의 기본적인 방법과 동일하면서 평균 문자 정합 횟수를 최소화하는 지점은 문자 후보 개수가 3이고 d 값이 21인 지점으로 그림 6, 7, 8에 각각 작은 원으로 표시하였다. 이 지점에서의 평균 문자 정합 횟수는 52.1회로 제안한 그래프 탐색 방법이 기본적인 인식 방법에 비해 인식 속도를 2.45배 증가시킨 결과를 보여준다.

정리하면 본 논문에서 제안한 방법에서 파라미터를 적절히 결정함으로써 기본적인 방법에 비해 2.45배의 속도를 얻었고, 대 분류 방법과 비교하면 대 분류 오류에 의한 인식률 저하를 배제할 수 있었다.

Ⅴ. 결론

본 논문에서는 필기 문자 인식에 있어서 모든 모델을 입력 패턴과 비교하는 기존 방법과 달리 일부 모델만 비교하고도 같은 인식률을 얻을 수 있는 모델 그래프 탐색 방법을 제안하였다. 즉, 이 방법은 기존 인식 방법에 비해 인식 속도가 2.45배 빠르며, 대 분류 방법과는 인식 속도의 증가에 따른 인식률 저하가 없다는 점이 차별화된다. 이 방법은 수많은 문자 조합을 처리해야 하는 단어 단위의 인식기를 스마트폰과 같이 처리 속도가 상대적

으로 느린 환경에서 구현하는데 매우 유리하다.

앞으로 남은 과제는 단어 단위의 인식기를 만드는 일이다. 실제로 사용자들은 현재 상용화되어 있는 문자 단위의 입력 방식 보다는 단어 단위의 입력 방식에서 훨씬 자유롭고 편리함을 느낄 것이다. 또한 단어를 이어서 흘려 쓰는 cursive script 형태나[13] 제약 없는 필기 형태는 [1] 복잡도가 너무 커서 아직은 시스템에 부담을 줄 것으로 보인다. 따라서 문자 단위로 끊어 쓰는 런온 (run-on) 형태의 단어 인식기를 만들고, 이를 현재 가장 많이 사용되는 종류의 스마트폰에서 실험해 보고자 한다.

참고문헌

[1] Se-Chang Oh, Jin-Young Ha, and Jin-H. Kim, "Context Dependent Search in Hidden Markov Models for unconstrained Handwriting Recognition", Int. Journal of Pattern Recognition, Elsevier Science Ltd., Vol. 28, No. 11, pp. 1693-1704, 1995.

[2] B.K. Sin, J.Y. Ha, and Se-Chang Oh, "Network-Based Approach to On-line Cursive Script Recognition", IEEE Transactions on Systems, Man and Cybernetics, Vol. 29, No. 2, pp. 321-328, April 1999.

[3] The Register, "PalmOne overturns Xerox Graffiti patent", 2004-05-24.

[4] "ACCESS Android Apps: Graffiti", http://www.access-company.com/products/androidapps, 2011년 10월.

[5] 강선미, 오근창, 황승욱, 양윤모, 김덕진, "문자인식을 위한 특징벡터의 부분 정보를 이용한 대분류 방법", 한국통신학회, 한국통신학회논문지, 제18권, 1호, pp. 32-38, 1993년 1월.

[6] 김백섭, 권희선, 전종홍, "적응학습과 대분류에 의한 온라인 한글 및 영문자 인식", 대한전자공학회 문자인식 워크샵, 제2권, pp. 105-115, 1994년 1월.

[7] 김광수, 양윤모, "코호넨 네트워크를 이용한 대규모 문자 세트의 대분류방법", 한국정보과학회, 한국정보과학회 1994년도 가을 학술발표논문집, 제21권, 제2호(A), pp. 753-756, 1994년 10월.

[8] 이승호, 조창제, 김일영, 조성배, 양현승, 김진형, "한글 문서 인식 시스템 SILNOON의 개발", 한국정보과학회, 한국정보과학회 1989년도 봄 학술발표논문집, 제16권, 1호, pp. 211-214, 1989년 4월.

[9] 김대환, 최현일, 이택헌, 김진형, "연결획 모델을 이용한 온라인 공간필기 인식", 2006 한국컴퓨터종합학술대회 논문집 Vol. 33, No. 1(B), pp.289-291, 2006.

[10] Donald J. Bemdt, James Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series", AAA1-94 Workshop on Knowledge Discovery in Databases, pp.359-370, 1994.

[11] 박태곤, 김경환, "DTW 알고리즘을 이용한 겹친 물체의 유사도 측정", 대한전자공학회 추계종합학술대회 학술발표논문집, pp. 747-748, 2008년 11월.

[12] 조정익, 이일병, "음성 정보와 DTW 알고리즘을 활용한 성인 컨텐츠 필터링", 한국지능시스템학회 춘계학술대회 학술발표논문집, 제18권 제1호, pp. 473-475, 2008년 4월.

[13] C.C. Tappert, "Cursive Script Recognition by Elastic Matching", IBM Journal of R&D, Vol. 26, No. 6, November 1982, pp. 765-771.

저자소개

오세창(Se-Chang Oh)

1997년 2월 KAIST 전산학과 공학박사
1995년 3월 ~ 1999년 1월 LG종합기술원 선임연구원
1999년 2월 ~ 2000년 2월 (주)인지소프트 이사
2000년 3월 ~ 2003년 8월 아주대학교 정보통신 전문대학원 조교수 대우
2004년 1월 ~ 현재 세종사이버대학교 정보통신학과 조교수
※관심분야 : 패턴인식, 데이터 마이닝, 소셜 네트워크