영문 대문자의 획 정보와 후처리를 이용한 온라인 필기 단어 인식기 구현

윤인구, 김우생
광운대학교 컴퓨터과학과
전화 : (02) 940-5217 / 팩스 : (02) 909-0998

Developing an On-line Handwritten Word Recognition System Using Stroke Information and Post-processing Techniques

InKu Yoon, WooSaeng Kim
Department of Computer Science
Kwangwoon Univ.
E-mail : {ikyoon, woosaeng}@cs.kwangwoon.ac.kr

Abstract

This paper presents new on-line handwritten algorithm for continuous alphabet uppercase characters. The algorithm is based on the idea that alphabet uppercase character consists of at most 4 strokes. It tries to determine the maximum output for a recognition result among outputs of four recognizers which have the capacity to discriminate the character using from 1 through 4 stroke information. The recognition module has 4 neural network based recognizers, which can recognize from 1 through 4 stroke character. We also use specialized post-processing techniques for improving the recognition performance. Trained on 440 input data and choosing 390 uppercase words for a recognition test we reached a 92% recognition rate.

1. 서론

최근 들어 전자책, 화이트보드, 그래픽 툴, PDA, 워드프로세서 등과 같이 온라인 문자인식기술을 적용한 응용들이 계속 증가하고 있다. 이는 인류가 오랫동안 사용하여 친숙한 필기방식의 인터페이스를 컴퓨터에도 적용하여 보다 손쉽게 데이터 입력을 가능하게 하기 위함이다.

온라인 문자인식은 기본적으로 테블릿과 같은 입력장치를 통해 전달받은 신호형태의 입력데이터를 인식에 사용될 수 있는 데이터로 변환한 뒤 시스템의 사전에 포함되어 있는 단어 중의 하나로 입력데이터를 분류하는 것이라고 정의할 수 있다[1]. 실시간으로 데이터를 입력받기 때문에 필기자의 필기습관이나 환경에 따라서 다양한 형태의 데이터가 존재하게 된다. 영문 필기의 경우 정서체, Run-On 정서체, 흘림체, Run-On 정서체와 흘림체의 혼용의 형태로 필기형태를 구분할 수 있으며 이러한 필기형태에 따라 인식에 적용되는 방법 또한 달라지게 된다[2]. 이렇게 필기된 문자나 단어를 인식을 하기 위해서는 인식기에 입력할 단위(획, 문자 혹은 단어 등)로 분할하는 과정이 필수적이며 이것은 전체 인식성능에 크게 영향을 미친다[3].

본 논문에서는 알파벳 대문자는 최대 4개의 획으로 구성될 수 있다는 특성을 이용하여 분리와 인식문제를 효율적으로 해결할 수 있는 인식 시스템을 구현하였다. 인식율 향상을 위해서 문자의 획 정보뿐만 아니라 알파벳 대문자의 전역적인 정보도 함께 특징으로 사용하였으며 구조적 특성상 발생할 수 있는

출력값의 모호성을 특성화된 후처리기법을 사용하여 해결하였다.

본 논문의 구성은 다음과 같다. 2장에서는 획간 정보를 이용한 인식 방법에 대해서 설명하고, 3장에서는 개선된 인식 시스템의 구성, 4장에서는 성능평가, 그리고 마지막으로 5장에서는 결론과 앞으로의 연구 방향에 대해서 설명한다.

2. 획간 정보를 이용한 인식

획간 정보를 이용한 인식 알고리즘은 일반적인 경우 알파벳 대문자는 최대 4개의 획 이내의 조합으로 하나의 문자가 구성된다고 가정하고 이를 바탕으로 문자분할과 인식을 동시에 수행한다[4]. 표 1 은 인식 대상이 되는 알파벳 대문자를 구성된 획수별로 구분한 것이다. 문자를 인식하기 위해서 테블릿 상에 필기되는 연속적인 획들을 순서대로 1획, 2획, 3획, 4획 문자를 인식할 수 있는 인식기에 각각 입력한 뒤 이들의 출력값을 비교하여 가장 큰 값을 출력한 인식기의 결과를 최종인식문자를 판단한다. 이 때, 최대 값을 출력한 인식기에 입력한 획 데이터의 개수가 문자를 구성하고 있는 획수가 되기 때문에 분할 과정도 인식과정과 함께 수행되게 된다.

획	1	2	3	4	획	1	2	3	4
A		○	○		N	○	○	○	
B	○	○			O	○			
C	○				P	○	○		
D		○			Q	○	○		
E		○	○	○	R		○	○	
F		○	○		S	○			
G		○	○		T		○		
H			○		U	○			
I			○		V	○	○		
J		○			W	○	○		○
K		○	○		X		○		
L	○				Y		○	○	
M	○	○	○	○	Z	○			

표 1 인식 대상이 되는 알파벳 대문자
Table 1 Recognizable Alphabet Set

예를 들어 그림 1 처럼 1획 'S, 2획 'K', 3획 'Y'로 구성된 "SKY"라는 단어를 전자펜으로 필기했을 경우 1획 문자 인식기에는 'S', 2획 문자 인식기에는 'SI', 3획 문자 인식기에는 'SK', 4획 문자 인식기에는 'SK\'가 입력되게 된다. 입력된 값들에 의해서 각 인식기에서는 인식결과를 출력하는데 이 때 2획부터 4획까지 입력된 데이터들은 인식기에서 학습되어진 결과가 아니므로 그 출력값이 상대적으로 1획 문자 인식기의 결과보다 작게 된다. 따라서, 필기된 단어의 첫 번째 글자를 1획 'S'로 인식하게 된다. 마찬가지로 인식글자의 마지막 획의 다음 획부터 같은 방식을 적용하여 마지막 획이 포함된 글자가 인식될 때까지 수행하여 최종적인 결과를 얻게 된다.

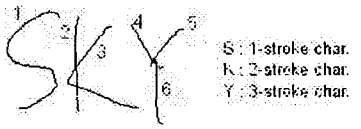

그림 1 전자펜을 이용해 필기한 영단어
Fig.1 Handwritten English Word

3. 인식 시스템

전체 시스템은 크게 전처리, 특징추출, 인식, 후처리의 4단계로 구성되어 있다.

3.1 전처리

전처리는 평활화, 훅제거, 거리여과 방법 등을 사용하였다.

3.2 특징추출

인식을 위해서 인식기에 입력할 적합한 특징정보가 필요하다. 본 논문에서는 인식할 대상의 특징으로 획의 형태를 나타내는 Sin, Cos 방향값과 각 문자를 구성하고 있는 획들이 갖는 전역적인 특징을 사용하였다.

1) 획 형태 정보

입력된 획의 형태를 표현하기 위해 8-방향코드 대신에 (1)과 같은 Sin, Cos 방향값을 사용하였다.

$$\cos\theta = \frac{(C_y - P_y)}{\sqrt{(C_x - P_x)^2 + (C_y - P_y)^2}}$$

$$\sin\theta = \frac{(C_x - P_x)}{\sqrt{(C_x - P_x)^2 + (C_y - P_y)^2}} \quad \cdots (1)$$

(단, 현재 위치가 (C_x, C_y), 이전 위치가 (P_x, P_y) 일 때)

2) 전역 정보

① 글자영역에서의 각 획의 위치

한 문자의 영역을 아래의 그림 2 와 같이 3X3 형태로 분할한 뒤 각 획의 시작점과 끝점의 위치를 그 영역에 매핑시켜서 한 글자 안에서 각 획의 위치를

표현할 수 있게끔 하였다.

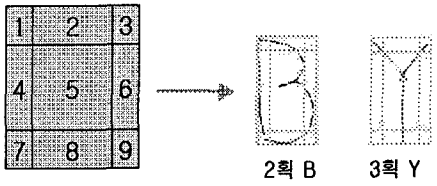

그림 2 한 글자영역 안에서 각 획들의 위치정보
Fig.2 Location Information of Every stroke in a Character Rectangle

② 각 획의 원시점들의 비율

원시점은 전처리가 수행된 후의 점들에 비해서 획의 형태정보를 나타내기에는 비효율적이지만 전처리가 끝난 후의 데이터가 갖지 못하는 필기속도나 필기점들의 밀도 등을 얻을 수 있다는 장점이 있다. 각 획의 원시점들의 비율은 (2)와 같은 식을 통하여 계산되어진다. 여기서 R_n은 n번째 획의 원시점 비율이고 OP_n은 n번째 획의 원시점 개수이다.

$$R_n = \frac{OP_n}{\sum_{t=1}^{k} OP_k} \quad \cdots \quad (2)$$

(단, $1 \le n \le k$, k : 문자를 구성하는 총 획수 ($1 \le k \le 4$))

③ 각 획의 필기속도

필기속도를 구하여 영문 대문자의 전역정보로 사용하고자 하였다. 각 획의 필기속도는 (3)과 같은 식을 이용하여 구하였다.

$$V_n = \frac{\sum_{t=2}^{k} D_n(k)}{T_n} \quad \cdots \quad (3)$$

(단, $1 \le n \le k$, k : n번째 획을 구성하는 원시점의 총개수 ($1 \le k \le 4$))

④ 펜의 이동정보

필기가 끝난 이전 획의 마지막 점(Pen-Up)부터 필기가 시작되는 현재 획의 시작점(Pen-Down) 사이의 이동방향과 거리 정보를 계산하여 이를 특징으로 사용한다. 방향정보는 획의 형태를 구할 때와 마찬가지로 이전 획의 마지막 점과 현재 획의 시작 점 사이의 방향값을 (1)을 이용하여 계산하고 거리정보는 현재 획의 길이에 대한 위의 두 점 사이의 거리비율을 이용하였다.

3.3 인식

필기된 문자를 인식하기 위해서 본 논문은 아래의 그림 3 과 같은 구조를 가진 인식기를 사용한다. 알파벳 대문자가 4획 이내로 구성되어 있다고 가정하였으므로 각 획 문자를 인식할 수 있게 인식기는 4개의 신경망으로 구성되어 있다. 신경망의 입력벡터는 획 형태를 표현하는 Sin, Cos 방향값 30개와 각 문자를 구성하는 획들의 전역적 특징으로 구성되어 있고 각 신경망의 입력노드는 각각 37개, 74개, 111개, 148개이고 신경망은 각각 12개(1획 문자), 18개(2획 문자), 11개(3획 문자), 3개(4획 문자)의 출력노드를 가지고 있다.

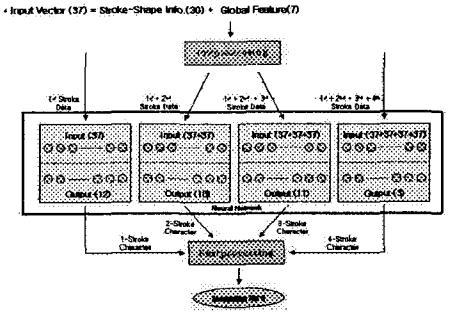

그림 3 인식기의 구조
Fig. 3 Structure of Recognition Module

3.4 후처리

1) 인식기 출력값 사이의 모호성 제거

본 논문에서 사용되어진 인식 알고리즘은 4개의 인식기 중에서 최대값을 가진 인식기의 결과를 최종인식 문자라고 결정한다. 이러한 구조적인 특성은 경우에 따라서 인식오류를 발생할 수 있는 원인이 되기도 한다. 그 이유는 4개의 인식기가 서로 독립적으로 각각의 출력값을 계산하기 때문에 그 값이 최대인 인식기의 결과를 최종문자로 결정하더라도 사용자가 기대한 인식결과가 아닐 수 있기 때문이다.

위와 같은 문제는 인식하고자하는 문자에 포함된 부분 획들이 학습된 알파벳 대문자와 유사한 경우에 발생한다. 따라서 문자에 포함된 부분 획이 학습시킨 알파벳 대문자와 유사할 경우 해당 인식기의 출력값을 강제적으로 0으로 할당시켜줌으로 해서 최종인식 문자 결정에서 모호성을 배제시켜주었다.

2) 글자를 이루는 획의 최대 개수 제한

현재 획 다음에 나올 수 있는 획형태에 관한 정보를 이용하여 한 글자를 구성하는 획수를 의도적으로 제한하여 인식율을 높이고자 하였다. 현재 획 다음에 나올 수 있는 획정보를 이용하여 이전 획에 연결될 수 없는 획이 올 경우 이전 획까지의 인식기 출력값

들만을 이용하여 인식문자를 결정하게 된다.

3) 1획 문자의 유효성 검사

획 단위로 인식을 시도하기 때문에 1획 인식기의 결과가 다음 획 인식기들에게 미칠 수 있다. 1획 문자의 경우 표 1에서 보는 바와 같이 곡선의 성격이 강한 문자들이기 때문에 1획 문자가 인식문자로 판단되어도 그 획의 형태가 직선일 경우에는 해당 출력값은 최종문자 결정과정에서 제외시켜서 나머지 인식기의 출력값들을 이용하여 인식문자를 결정하게 된다.

4. 구현 및 성능 평가

제안하는 방법의 성능 테스트를 위해 Pentium-II 시스템에서 Windows NT 4.0을 기반으로 하여 Visual C++ 6.0을 이용하여 프로그램을 작성하였다. 입력장치로는 Wacom사의 6x8 크기의 테블릿과 전자펜을 사용하였으며 인식기를 구성하는 신경망은 별도의 DLL로 제작하여 구현된 프로그램에 포함시켰다.

성능을 평가하기 위해 영단어를 임의로 130개를 선택한 다음 각각 3회씩 필기하여 총 390번의 필기를 통하여 인식율을 계산하였다. 1획 문자 인식기에는 120개의 입력패턴을, 2획 문자 인식기에는 180개의 입력패턴을, 3획 문자 인식기에는 110개의 입력패턴을, 4획 문자 인식기에는 30개의 입력패턴을 학습시켰다.

인식 알고리즘은 획 정보를 이용하여 인식과 분할을 동시에 수행하는 방식이기 때문에 한 글자를 구성하고 있는 획의 개수와 단어에 포함된 문자의 개수가 인식율에 많은 영향을 미치는 요인이 된다. 선택된 단어들은 최소 3개에서 최대 12개까지 문자를 포함하고 있으며 평균적으로 5개의 문자로 구성된 단어가 가장 많았다.

표 2 는 임의로 필기한 영어 단어를 인식 테스트한 결과를 나타내고 있는데 인식기만을 사용한 경우보다 후처리과정을 같이 수행했을 경우가 약 7%정도의 인식율이 높은 것을 알 수 있다. 이는 인식 알고리즘에 적용된 후처리기법이 각 인식기의 독립적인 출력값들을 잘 조절하여 인식문자를 결정하였으며, 인식에 영향을 미치는 그 외 인자들을 감소시켰기 때문이다.

인식방법	Correct	Error	인식율(%)
인식기	332	58	85.12
인식기 + 후처리	359	31	92.05

표 2 인식 결과
Table 2 Recognition Result

5. 결론

본 논문에서는 사용자가 알파벳 대문자로 필기한 단어를 온라인으로 인식하기 위한 인식 시스템을 구현하였다. 문자를 인식하고 분리하는 알고리즘은 연속 필기된 획의 조합을 이용한 방법을 사용하였고 알파벳 대문자를 구성하는 획 정보와 전역적인 글자 정보를 특징으로 사용하였으며, 인식율을 향상시키기 위해서 인식기의 출력값 사이의 모호성을 배제해주는 특성화된 후처리기법을 사용하였다.

앞으로의 과제는 보다 정확한 인식문자를 결정할 수 있는 방법에 대한 보완이 필요하다. 인식 알고리즘의 인식기들은 독립적으로 작동되기 때문에 기대한 출력값과는 다르게 인식기에서 결과가 출력될 수 있다. 따라서 각 인식기에서 얻어지는 출력값들을 보다 공평하게 평가하고 판정할 수 있는 방법을 후처리과정에 포함시킨다면 좀 더 나은 인식율을 얻을 수 있을 것으로 기대되어진다.

6. Reference

[1] G. Seni et al, "Large Vocabulary Recognition of On-Line Handwritten Cursive Words", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7, July 1996.

[2] C .C. Tappert, C. Y. Suen, T. Wakahara, "The State of the Art in On-line Handwriting Recognition", IEEE Transaction on Pattern Analysis and Machine Intelligence, 1990.

[3] Anil K, Jain, Robert P.W. Duin, Jianchang Mao, "Statistical Pattern Recognition: A Review", IEEE Transaction on Pattern Analysis and Machine Intelligence, Jan., 2000.

[4] 조현철, 김우생, "영문 대문자의 획간 조합 순서를 이용한 온라인 필기의 문자열 인식", 정보과학회 학술 발표 논문집, Vol. 2, 1999.