

은닉 마르코프 모델과 레벨 빌딩 알고리즘을 이용한 흘림체 한글의 온라인 인식

김 상 균[†] · 김 경 현^{††} · 이 종 국^{†††}
· 이 재 옥^{††††} · 김 항 준^{†††††}

요 약

자소 단위의 HMM을 설계하고 隸字 원리에 따라 연결한 한글 인식 네트워크에서 인식속도는 많은 경로수로 인해 상당히 느려진다. 본 논문에서는 탐색 속도를 개선하기 위해 최적의 네트워크 탐색 방법인 레벨 빌딩 알고리즘을 수정, 적용한 온라인 한글 인식 모델을 제안한다. 한글 인식을 위한 레벨 빌딩은 초성-중성-종성 순의 정해진 필기 한글 구조를 반영한 syntax-directed 레벨 빌딩 탐색 알고리즘으로, 전체 11,172개의 경로를 가지는 방대한 크기의 인식 네트워크 탐색에서 시간복잡도가 경로수에 비례적이고 노드 수 즉, 개별 자소 HMM의 수에만 의존하는 효율적인 탐색 방법이다. 제한된 방법의 효용성을 입증하기 위한 인식 실험에서 KAIST의 온라인 한글 데이터, 15,000자를 대상으로 한 자당 0.72초의 인식속도를 보였다.

On-line Recognition of Cursive Korean Characters Based on Hidden Markov Model and Level Building

Sang Kyoon Kim[†] · Kyung Hyun Kim^{††} · Jong Kook Lee^{†††}
· Jae-Uk Lee^{††††} · and Hang Joon Kim^{†††††}

ABSTRACT

In this paper, we propose a novel recognition model of on-line cursive Korean characters using HMM(Hidden Markov Model) and level building algorithm. The model is constructed as a form of recognition network with HMM for graphemes and Korean combination rules. Though the network is so flexible as to accomodate variability of input patterns, it has a problem of recognition speed caused by 11,172 search paths. To settle the problem, we modify the level building algorithm to be adapted directly to the Korean combination rules and apply it to the model. The modified algorithm is efficient network search procedure time complexity of which depends on the number of HMMs for each grapheme, not the number of paths in the extensive recognition network. A test with 15,000 handwritten characters shows recognition rate 90% and speed of 0.72 second per character.

† 준 회 원:인제대학교 전산학과 전임강사
†† 준 회 원:경북대학교 대학원 컴퓨터공학과
††† 정 회 원:안동대학교 컴퓨터공학과 전임강사
†††† 정 회 원:동서대학교 컴퓨터공학과 부교수
††††† 종신회원:경북대학교 컴퓨터공학과 교수
논문접수:1995년 12월 29일, 심사완료:1996년 3월 15일

1. 서 론

온라인 필기체 한글은 필기시 흘림 획의 허용 여부에 따라 정서체와 흘림체로 나눌 수 있다. 정서체는 “ㄱ”, “ㄴ”, “ㅇ”, “ㅡ”, “\”, “|”, “/” 등과 같은 7개 내지 10개의 기본획들로 필기되며, 필기 순서 또한 잘 지켜진 경우의 필기 형태이다. 흘림체는 정서체에서의 기본획뿐만 아니라 다양한 획들이 한 획으로 연결되어 자소 내의 흘림과 자소간의 흘림이 빈번한 필기 형태이다. 대부분의 필기자가 필기의 효율을 위하여 혹은 습관적으로 필기시 흘림을 가지므로, 실용적인 인식 시스템을 개발하기 위해서는 흘림체 한글의 인식 기능은 필수적이다.

흘림체 한글에서 발생하는 획수, 획간의 연결, 흘림에 의한 자소 변형을 수용하기 위해, 순차적으로 발생하는 신호의 다양한 시간적 변화에 유연하게 반응하는 통계적 모델인 은닉 마르코프 모델(Hidden Markov Model)이 인식 도구로 제안되었다[1, 2]. 이 모델은 표현하는 대상 패턴을 학습 과정을 통해 명확한 확률값으로 표현하는 확률 모델로서 학습 방법과 분석 과정이 수학적으로 명확히 표현되므로, 많은 데이터를 얻을 수 있는 음성 인식, 문자 인식에서 신뢰성 있는 확률값의 추정이 가능하다[1, 2, 3].

최근 제안된 한글 인식 시스템인 붕넷[4, 5]은 초성, 중성, 종성의 필순을 지켜야 하는 제약 외에 필기에 제한을 두지 않는 인식 시스템으로 초, 중, 종성의 기본 자소와 자소 사이의 위치 관계 및 흘림체 글자의 이어 쓰기를 표현할 수 있는 연결획의 설계에 은닉 마르코프 모델을 사용하였다. 붕넷은 글자 인식을 위한 여러 정보를 네트워크에 표현한 것으로 기본적으로 네트워크 자체가 표현하는 정적 글자 구조 정보와, 글꼴에 따라 다양하게 변하는 특징을 노드간 확률적 정의로 표현한 동적 정보를 갖고 있다. 이것은 다양한 글자체의 많은 변형을 적은 수의 모델로 표현할 수 있고 자소의 정확한 분리 과정에 비의존적으로 글자를 인식할 수 있다. 이 모델에서의 글자 인식 문제는 입력 패턴에 대해서 네트워크 상의 최대 확률값을 가지는 경로를 찾는 것으로 정의된다. 그러나 한글 인식 네트워크가 가질 수 있는 모든 경로의 수는 조합 가능한 모든 한글의 수 11,172개에 이르게 되므로, 이 모든 경로를 탐색하여 한글을 인식하는데는

많은 연산 시간이 요구된다.

본 논문에서는 HMM과 레벨 빌딩 탐색 알고리즘을 사용한 흘림체 한글의 인식 모델을 제안한다. 인식 모델은 자소 단위의 HMM과 자소간의 연결획 HMM에 의한 한글 인식 네트워크로 구성된다. 그리고 인식속도를 개선하기 위해 초성-연결획-중성-연결획-중성의 구성을 가지는 한글의 제자 원리에 입각한 순으로 네트워크를 탐색하는 수정된 레벨 빌딩 알고리즘을 적용한다. 한글을 이루는 모든 경로를 찾을 경우 발생하는 각 모델의 중복된 방문을 피하고, 한 방향으로만 진행하는 인식 네트워크의 보다 빠른 탐색을 위해 수정된 레벨 빌딩 알고리즘을 사용하였다. 이 방법은 네트워크 상의 최적 경로, 즉 인식 글자를 찾는 데 있어 각 HMM을 한번씩만 탐색하고 역추적 과정을 통해 적절한 자소 모델을 결정한다. 레벨 빌딩 알고리즘은 각 모델에 대해 비터비 알고리즘을 수행하여 생성된 최대 확률값을 사용하여 전체 모델 내에서 인식된 글자의 경로를 유지한다. 수정된 레벨 빌딩 알고리즘의 네트워크 탐색 속도는 단지 네트워크를 이루는 자소 HMM, 연결획 HMM의 수에 비례하는 효율적인 방법이다.

본 논문의 구성은 다음과 같다. 2장에서 HMM의 간략한 특성을 살펴보고, 3장에서 한글 인식기의 모형을 제시한다. 그리고 4장에서 온라인 한글 인식에서 레벨 빌딩 알고리즘의 적용 방법을 제시하고, 5장에서 실험 결과를 기술한다. 마지막으로 6장에서 결론을 맺는다.

2. 은닉 마르코프 모델

2.1 은닉 마르코프 모델의 구성

1960년대 말 처음 소개된 HMM은 시간에 따른 패턴의 변화를 적절히 모델링 하는 장점을 가지므로 주로 연속 자동 음성 인식 분야[1, 2, 3]에서 적용되어 왔으나, 최근 들어서는 문자인식 분야에서도 많은 적용 연구가 진행되고 있다[4, 5]. HMM은 마르코프 체인과 각 상태에서의 관측 심볼 확률 분포로 정의할 수 있으며, 일반적으로 HMM의 구성 요소는 아래와 같다.

- (1) N: 상태의 수
- (2) M: 관측 심볼의 수

- (3) T: 관측열의 길이
- (4) $Q = \{q_t\}$: 상태 집합, $q_t \in \{1, 2, \dots, N\}$, $t = 1, 2, \dots, T$
- (5) $V = \{V_k\}$: 관측 가능한 심볼들의 집합, $k = 1, 2, \dots, M$
- (6) $A = \{a_{ij}\}$: 상태 전이 확률 분포
 이는 마르코프 모델에서 상태 i 에서 상태 j 로의 전이 확률을 나타내는 파라미터로 $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, $1 \leq i, j \leq N$ 로 정의할 수 있다.
- (7) $B = \{b_j(v_k)\}$: 관측 심볼 확률 분포, $j = 1, 2, \dots, N$, $k = 1, 2, \dots, M$
 시간 t 에 상태 S_j 에서 k 번째 심볼 V_k 가 나타나는 확률을 의미한다.
- (8) $\pi = \{\pi_i\}$: 초기 상태 확률 분포, $i = 1, 2, \dots, N$
 $\pi_i = P[q_1 = i]$: i 상태의 초기 확률값.

HMM은 이러한 확률 매개변수들을 이용해서 주어진 관측열의 집합을 표현한다. 즉 $O = O_1, O_2, \dots, O_T$ 라는 관측열이 주어졌을 때, 외부에서 그 상태 전이 과정을 직접적으로 관측할 수는 없어도 자체의 확률 매개변수를 사용하여 마르코프 과정의 확률 함수로 모델화할 수 있다.

2.2 HMM 파라미터의 초기 추정치

HMM에서 고려해야 할 한 요소로 A, B 매트릭스와 초기 확률값 결정을 들 수 있다. 이론적으로 추정식은 우도 함수의 지역 최대값(local maximum)에 상응하는 HMM 파라미터의 값을 주어야 한다. 훈련 과정을 통해 추정된 확률값 P는 지역 최대값이므로 초기값의 변화에 따라 P값의 변이가 생길 수 있다. 본 연구에서는 식 (1), (2)를 만족하는 난수를 사용하여 초기값을 설정하였다[1].

$$\sum_{j=1}^N a_{ij} = 1, i = 1, 2, \dots, N \quad (1)$$

$$\sum_{k=1}^M b_j(k) = 1, k = 1, 2, \dots, M \quad (2)$$

2.3 은닉 마르코프 모델의 형태

HMM의 형태는 한 상태에서 다른 모든 상태로 전이 확률값을 가지는 ergodic 모델을 비롯하여 시간이 증가함에 따라 상태의 인덱스도 같이 증가하는 left-right 모델(Bakis model) 등 다양한 마르코프 모델이 있다.

이 중 본 연구에서 사용한 left-right형의 HMM은 음성 인식이나 온라인 문자인식에서처럼 시간에 따라 순차적으로 변하는 특성을 지닌 데이터를 모델링 하는데 적합한 아래와 같은 특징을 가진다.

- (1) 한번 방문한 노드를 다시 방문할 수 없다.

$$a_{ij} = 0, j < i$$

- (2) 상태열은 항상 첫 노드에서 시작하고 마지막 노드에서 종료한다.

$$\pi_i = \begin{cases} 0 & i \neq 1 \\ 1 & i = 1 \end{cases}$$

- (3) 상태 사이의 전이에서 인덱스의 큰 변화는 발생하지 않는다.

$$a_{ij} = 0, |j - i| \geq \Delta$$

이러한 left-right 형의 모델을 사용할 경우 고려해야 할 두 가지 사항이 있다[3]. 첫째, 하나의 '연속적인 관측값은 이 값이 마지막 상태 q_N 에 도달하게 되면 나머지 관측값은 더이상 이전 상태에 영향을 미치지 못하게 된다. 따라서 이 모델을 훈련시키기 위해서는 첫 상태 q_1 에서 시작되는 관측값들의 집합이 요구되어진다. 둘째, 전향-후향(forward-backward) 확률의 각 개별적 집합에 대한 scale factor가 다르므로 한 패턴에 대한 여러 관측값들을 이용한 추정 과정에서 scaling에 주의하여야 한다. 이런 문제점을 해결한 다중 관측값을 사용하는 수정된 훈련 방법은 2.4절에서 설명한다.

2.4 다중 관측값

Left-right형 모델의 파라미터에 영향을 미치는 중요한 요소 중의 하나로 훈련을 위한 관측 샘플을 들 수 있다. 본 연구에서는 여러 필기자의 글씨체에 독립적인 모델을 구성하기 위해 여러 사람의 필기 샘플 $O = [O^{(1)}, O^{(2)}, \dots, O^{(k)}]$ 를 훈련 샘플로 사용한다. $O^{(k)}$ 는 k 번째 훈련 샘플 $[O_1^{(k)}, O_2^{(k)}, \dots, O_{T_k}^{(k)}]$ 을 나타낸다. 이러한 다중 관측값(Multiple Observation Sequence)을 사

용하여 모델 파라미터를 추정하는 것은 훈련 샘플들이 서로 독립이라는 가정하에,

$$P(O|\lambda) = \sum_{k=1}^K P(O^{(k)}|\lambda) = \sum_{k=1}^K P_k \tag{3}$$

식 (3)을 최대로 하는 모델 λ 의 파라미터를 추정하는 것이다. 파라미터 A, B를 추정하는 식은 다양한 관측값의 발생 빈도수에 기초를 둔 것이므로 다중 관측값을 위한 추정식도 각 개별적 관측값들의 발생 빈도수를 추가하여 쉽게 확장할 수 있다[1]. 수정된 식은

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_{i-1}} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}(j)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_{i-1}} \alpha_t^k(i) \beta_t^k(i)} \tag{4}$$

와

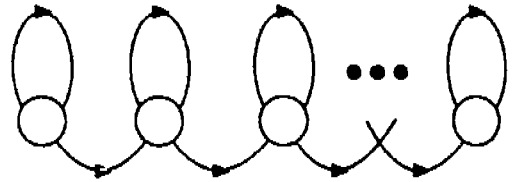
$$\bar{b}_j(l) = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1, s.t. O_t=ul}^{T_{i-1}} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_{i-1}} \alpha_t^k(i) \beta_t^k(i)} \tag{5}$$

3. 한글 인식 모델

3.1 자소-모형과 연결획 모형

HMM은 훈련을 통해 얻어지는 확률 파라미터 외에도 기본 모델의 구조, 상태의 수, 관측 심볼의 종류 등을 미리 결정해야 한다. 이들은 주어진 문제의 특성에 따라 가변적이므로 설계자의 경험에 의존할 수밖에 없다. 본 연구에서는 한글을 이루는 초성 19자, 중성 21자, 종성 27자를 기본 자소 모형 단위로 하고, 이들 각각에 대해서 (그림 1)에 나타나는 현재의 상태에서 자기 자신으로의 전이와 다음 상태로의 전이만을 갖는 left-right형의 모델을 사용하여 표현하였다.

온라인 필기는 첫 pen-down에서 시작하여 마지막 pen-up에 이르기까지의 곡선 궤적으로 샘플링 되어 일련의 획으로 나타난다. 보통은 이와 같이 pen-down된 부분만을 획으로 간주한다. 이 획들 간의 공간적 위치 정보는 보통 pen-up에 의해 구분되어지나, 글씨

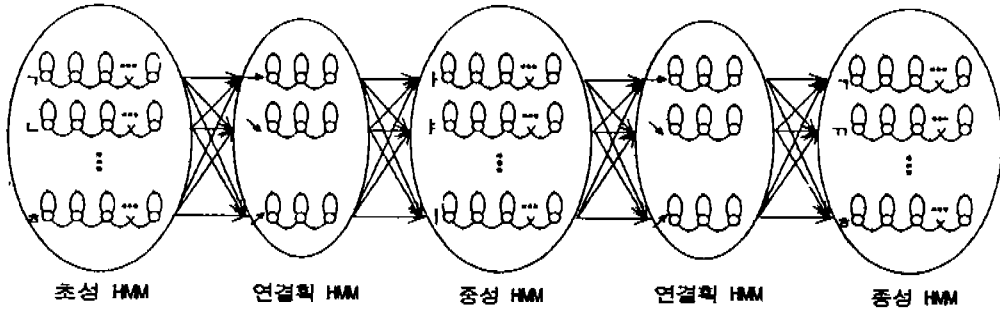


(그림 1) 실험에 사용된 left-right 모델
(Fig. 1) A left-to-right HMM

를 흘려 쓸 경우에는 연속된 여러 획이 연결되어 하나의 획으로 나타날 수 있다. 이러한 획과 획사이를 연결하는 가상선을 연결획(ligature)이라 한다[4]. 자소 내에 포함되는 연결획은 각 자소 HMM에 의해 적절히 표현될 수 있고, 자소간 연결획은 따로 연결획 HMM을 사용하여 표현하였다. 자소 모형의 상태의 수는 초, 중, 종성에 대해 각각 10, 15, 15개를, 연결획 모형에는 3 상태를 할당하였다. 그리고 각 상태에서 관측 심볼은 방향에 따라 구분한 8방향 코드를 사용하였다.

이 모형을 초성-연결획-중성 또는 초성-연결획-중성-연결획-중성 순으로 연결하면 완전한 한 글자를 표현할 수 있다. 이러한 방법으로 모든 한글 음절 각각에 대해 하나씩의 HMM을 구성할 수 있다. 제한된 문제 안에서 인식하려는 글자 수가 많지 않으면 이와 같은 방법이 문제되지 않지만 일반적인 한글 인식에 대해서는 조합 가능한 한글의 수가 11,172자에 이르므로 이 모든 모형을 유지하기 위해 많은 메모리와 이에 따른 방대한 계산 시간이 요구되므로 실시간 응용에는 무리가 따른다.

메모리 요구에 대한 문제는 구조적 공유(structural tying) 기법[4, 5]에 의해 해결할 수 있다. 동일한 자소가 같은 HMM을 공유하기 위해 자소당 하나 또는 여러 개의 HMM을 설계하여 이를 필순에 따라 연결하고, 한글의 제자 원리에 따라 결합에서 일반적인 한글 음절 모형을 구성한다. (그림 2)는 전체 모형을 도시화한 것으로, 적은 수의 모델로 전체 한글 모형을 구성하므로 메모리 문제를 해결한다. 네트워크 상의 모든 경로의 탐색으로 발생하는 계산 시간에 대한 문제는 레벨 빌딩 알고리즘으로 해결할 수 있다. 글자의 입력 패턴 중의 일부를 각 자소 모델과 연결획 모델에 적합 시키는데는 비터비 알고리즘을 사용하고



(그림 2) HMM에 기반을 둔 한글 인식 네트워크
 (Fig. 2) A recognition network for Korean characters based on HMM

실제 글자를 표현하는 전체 모델 내에서 인식된 글자의 경로를 유지하는데는 레벨 빌딩 알고리즘을 사용한다.

4. 인식

4.1 비터비(Viterbi) 알고리즘

비터비 알고리즘은 최적원리(principle of optimality)에 입각한 동적 프로그래밍 기법의 하나로, 임의의 상태에 이르는 경로 비용 또는 확률은 이전 상태의 비용과 거기서 현재 상태로의 전이 비용을 합하는 방식으로 순환 계산하는 기법이다. 이 점은 마르코프 모델의 가정과 그것에 따르는 시간적 제약에 밀접한 관계를 가지므로 마르코프 모델 내의 최적 경로를 찾는 데 적용할 수 있다. 자소나 연결획 모델중 하나의 HMM에 적용한 비터비 알고리즘은 다음과 같다[1].

Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), 1 \leq i \leq N \quad (6)$$

$$\Psi_1(i) = 0. \quad (7)$$

Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad (8)$$

$$2 \leq t \leq T, 1 \leq j \leq N$$

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad (9)$$

$$2 \leq t \leq T, 1 \leq j \leq N.$$

Termination:

$$p^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (10)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]. \quad (11)$$

State Sequence backtracking:

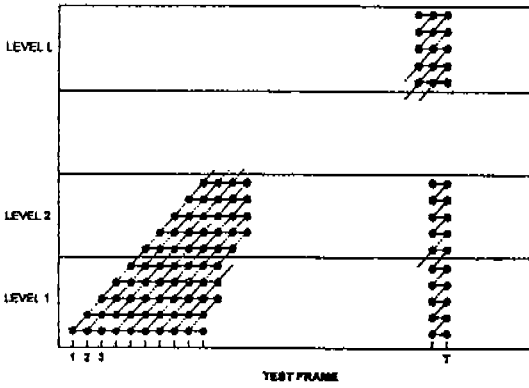
$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \dots, 1 \quad (12)$$

하나의 모델에 대한 순환 계산과 역추적 계산 과정은 다음절의 레벨 빌딩 알고리즘에서 수정되어 적용한다.

4.2 레벨 빌딩을 사용한 최우 적합

디지털타이저에서 입력된 글자, 즉 일련의 관측값(8방향 코드) $O_t, t = 1, 2, \dots, T$ 와 (그림 2)의 글자를 구성하는 개별적 자소 및 연결획 HMM 집합 $\{W_j\}$ 가 주어졌을 때 인식의 단계에서는 관측값 O_t 를 사용하여 일련의 적합된 모델 $\{W_{1t}, W_{2t}, \dots, W_{10t}\}$ 을 결정한다. 이는 관측값과 모델 상태의 결합확률을 최고로 하는 매칭 문제로 정의될 수 있다.

(그림 3)은 전체 모델에서 최고 확률값을 출력하는 일련의 모델을 찾기 위해 적용한 레벨 빌딩 알고리즘을 설명한다. 이 그림에서는 각 개별 모델은 5 상태를 가지며 연속 모델 $W_{1t}, W_{2t}, i < j$ 에서는 어떠한 제약 없이 모든 모델이 될 수 있다고 가정한다. 즉, 한 모델 다음에 어떤 다른 모델도 연속으로 배열될 수 있다. 먼저 레벨 $l=11$ (초기 레벨)의 단계의 프레임 1에서



(그림 3) 5상태를 가지는 HMM에 기초를 둔 level-building의 수행
 (Fig. 3) Implementation of level building based on five-state HMMs

시작하여 모델 $q(W_q)$ 를 관측값 O 에 매칭시킨다. 이는 아래의 수정된 비터비 알고리즘[12]에 의해 수행된다.

1) Initialization

$$\delta_1(1) = [b_1^q(O_1)], \Pr[O_1, O_2, \dots, O_T] \quad (13)$$

$$\text{and } S_1, S_2, \dots, S_{t-1}, j \in \{A, B\}$$

부분적 상태들과 관측값의 결합확률을 보여준다.

$$\delta_1(j) = 0, j = 2, 3, \dots, N \quad (14)$$

2) Recursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) * [a_{ij}^q]] * [b_j^q(O_t)] \quad (15)$$

$$2 \leq t \leq T, 1 \leq j \leq N$$

3) Termination

$$P(l, t, q) = \delta_t(N), 1 \leq t \leq T \quad (16)$$

$$B(l, t, q) = 0 \quad (17)$$

첫 레벨에서는 모든 모델에 대해 이 알고리즘을 적용하고 계산이 끝난 후 아래의 새로운 배열 $\hat{P}, \hat{B}, \hat{W}$ 을 생성한다.

$$\hat{P}(l, t) = \max_q [P(l, t, q)] \quad (18)$$

$$\hat{B}(l, t) = B[l, t, \arg \max_q P(l, t, q)] \quad (19)$$

$$\hat{W}(l, t) = \arg \max_q P(l, t, q) \quad (20)$$

위의 식에서 \hat{P} 은 각 레벨의 최고 확률값을 나타내고, \hat{B} 은 이전 모델에서의 프레임을 가리키는 역추적 포인터이고, \hat{W} 은 최고 확률을 출력하는 모델을 나타낸다.

레벨 2 이상에서는 초기값을 설정하는 과정에서 위와 차이점을 가진다. 이 레벨에서는 이전 단계의 결과를 초기값으로 설정하므로 아래의 초기화 과정을 수행한다.

$$\delta_1(1) = 0, \alpha_1(1) = 1 \quad (21)$$

$$\delta_t(1) = \max [\hat{P}(l-1, t-1), a_{11}^q \delta_{t-1}(1)] \quad (22)$$

$$* [b_1^q(O_t)], 2 \leq t \leq T$$

$$\alpha_t(1) = \begin{cases} t-1 & \text{if } \hat{P}(l-1, t-1) > \delta_{t-1}(1) \\ * a_{11}^q & \\ \alpha_{t-1}(1) & \text{otherwise} \end{cases} \quad (23)$$

식 (21)은 $\delta_1(1)$ 의 값을 0으로 하고 식 (22)는 이전 레벨로부터 가장 적절한-가장 큰 결합확률을 생성하는 위치를 기록한다. 이전 모델이 끝나는 전 레벨에서의 프레임의 기록을 위해 식 (23)에서 가장 적절한 초기 역추적 포인터 배열을 생성한다. 순환(Recursion)계산 동안 역추적 포인터는

$$\alpha_t(j) = \alpha_{t-1}[\arg \max_{1 \leq i \leq N} (\delta_{t-1}(i) * a_{ij}^q)] \quad (24)$$

로 계산되어지며, 레벨의 끝에서 확률과 역추적 포인터 배열은

$$P(l, t, q) = \delta_t(N), 1 \leq t \leq T \quad (25)$$

$$B(l, t, q) = \delta_t(N), 1 \leq t \leq T \quad (26)$$

로 계산되어진다. 그리고 $\hat{P}, \hat{B}, \hat{W}$ 을 결정 한 후 다음 레벨로 진행된다.

이 모든 과정은 주어진 최고 레벨 L에 다다른 끝나게 된다. 그리고 레벨 수가 L일 때, 확률 $\hat{P}(L, T)$ 을 가지는 적합한 모델 $\{W_{[1]}, W_{[2]}, \dots, W_{[L]}\}$ 를 역추적 포인터 배열 $\hat{B}(l, t)$ 를 사용하여 다음의 알고리즘을 통해

구한다.

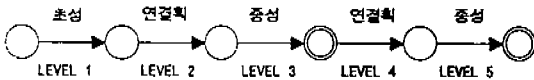
Algorithm(Back tracking)

```

Begin
    level = L;
    back = T;
    while (level > 0) do
        begin
            W|level| =  $\hat{W}$ (level, back);
            back =  $\hat{B}$ (level, back);
            level = level - 1;
        end
    end
End
    
```

4.3 레벨 빌딩을 이용한 한글 인식

위의 레벨 빌딩은 한 모델 다음에 다른 어떤 모델도 연속할 수 있다는 가정 아래 아무런 제약 없는 알고리즘이다. 그러나 (그림 2)의 한글 인식 네트워크를 살펴보면 (그림 4)에서 보이는 바와 은 일반적인 필기 순서인 초성-연결획-중성-연결획-중성의 구조를 지닌다. 이처럼 인식 네트워크가 한글의 필기 형식을 반영하므로 이런 구조를 수용하는 제약 조건-레벨 빌딩 알고리즘에서 *l*번째 레벨에서의 선택 가능한 후보는 (*l* + 1)번째 레벨에서의 연속적으로 후보가 될 필요는 없다-을 가지는 레벨 빌딩 알고리즘을 필요로 한다. 즉 (그림 4)에서 보이는 바와 같이 초성 HMM 다음에는 꼭 연결획 HMM이 나타나야 하고 그 다음에는 중성, 연결획, 중성 HMM이 순서대로 나타나야 한다.



(그림 4) 한글을 위한 수정된 level-building
(Fig. 4) The modified level building for Korean

수정된 레벨 빌딩 알고리즘은 각 레벨에서는 시간 *t*에서의 한글을 이루는 연속적인 자소 HMM과 연결획 HMM 중에서 적합도가 가장 높은 모델을 출력하게 된다. 레벨 5까지의 계산이 모두 끝난 후에 역추적 과정을 통해 적합한 자소 HMM을 찾아 완전한 한글 인식에 이른다. 제한 조건을 가지는 레벨 빌딩은 각

레벨에서 네트워크 내의 전체 모델에 대해 비터비 계산을 수행하지 않고, 레벨별로 제한된 모델 내에서 비터비 계산을 수행하게 되므로 효율적인 인식 알고리즘이 된다. 레벨의 수가 *L*인 전체 시스템을 구성하는 개별 HMM의 수를 *M*이라 두고 이들 각 HMM의 상태의 수를 *N*이라 가정하면, 식 (13)에서 (26)까지의 레벨 빌딩 알고리즘의 시간복잡도는

$$\begin{aligned}
 O &= O(L(M(N + TN^2 + T) + TM)) \\
 &= O(LM(N + TN^2 + 2T)) \\
 &= O(LM(N + T(N^2 + 2))) \\
 &\approx O(LM(N + TN^2)) \\
 &\approx O(LMTN^2)
 \end{aligned}$$

이 된다. 그러나 각 레벨에서의 모델로 나타날 수 있는 HMM의 집합을 {*Q*₁, *Q*₂, ..., *Q*_{*L*}}로 두면, 수정된 알고리즘에서 전체 레벨에 통해 계산되어지는 모델의 수는

$$M = \sum_{i=1}^L Q_i$$

이며 시간복잡도는

$$\begin{aligned}
 O &= O(\sum_{i=1}^L Q_i TN^2) \\
 &= O(MTN^2)
 \end{aligned}$$

이 된다. 그래서 이 알고리즘은 레벨의 수와는 무관하고 단지 모델의 수에만 의존하는 보다 빠른 네트워크 탐색 방법이 된다.

Algorithm(The modified level building)

```

Begin
    for l = 1 to 5 do
        begin
            for q = 1 to Q[l] do
                begin
                    Initialization(l, q);
                    Recursion(l, q);
                    Termination(l, q);
                end
            end
        end
    end
for t = 1 to T do
    
```

```

begin
 $\hat{P}(l, t) = \max_q [P(l, t, q)]$ 
 $\hat{B}(l, t) = B[l, t, \arg \max_q P(l, t, q)]$ 
 $\hat{W}(l, t) = \arg \max_q [P(l, t, q)]$ 
end
End

```

Algorithm(Back Tracking)

```

Begin
if ( $\hat{P}(3, T) > \hat{P}(5, T)$ ) then level = 3;
else level = 5;
back = T;
while (level > 0) do
begin
 $W_{[level]} = \hat{W}(level, back)$ ;
 $back = \hat{B}(level, back)$ ;
level = level - 1;
end
End

```

위의 알고리즘의 레벨 1에서는 초성에 해당되는 각 모델 W_q 에 대해서 위의 수정된 비터비 알고리즘을 적용한 후 초성 HMM중에 각 프레임(관측값, $O=O_1, O_2, \dots, O_T$)별 최고 확률값 $\hat{P}(1, t)$ 에 저장하고, 최고 확률을 출력하는 초성 모델의 인덱스를 ($\hat{W}(1, t)$)에 저장한다. 첫 레벨이므로 역추적 포인터는 어떤 모델도 가리키지 않는다. 이들은 각 시간 t 에서 최적화 모델을 결정하여 다음 레벨의 비터비 계산에 초기값으로 사용된다. 다음 레벨에서는 초성과 중성을 연결하는 연결책 모형을 프레임별로 적합 시키게 된다. 식 (22), (23)에 의해 레벨 1에서 초성 모델 최고 확률값과 레벨 2에서 연결책 모델의 첫 상태에 머무를 확률값을 비교하여 이전 레벨에서의 연결 위치와 프레임을 기록하는 초기화 과정을 수행한다. 초기화한 후 비터비 알고리즘의 순환 과정을 수행하면서 각 모델의 각 상태에서 가장 큰 확률값을 프레임에 기록하여 역추적 포인터로 사용한다. 레벨 2의 과정을 중성, 연결책, 중성 HMM을 결정하는 레벨 3, 4, 5에도 동일하게 적용한다.

모든 레벨에 대한 비터비 계산이 끝난 후 적합 모델을 결정하는 역추적 과정을 수행한다. 실제 한글 한 글자는 (그림 4)의 상태 3 혹은 상태 5에서 완전한 글자를 이루게 된다. 이들 두 상태의 출력 확률 $\hat{P}(3, T)$ 와 $\hat{P}(5, T)$ 의 크기를 비교하여 중성의 유무를 판별할 수 있으며 위의 Back Tracking 알고리즘에 의해 연속적인 모델 $\{W_{[1]}, W_{[2]}, W_{[3]}\}$ 또는 $\{W_{[1]}, W_{[2]}, \dots, W_{[5]}\}$ 가 결정된다.

5. 실험 및 결과

제안된 레벨 빌딩을 이용한 흘림체 한글 온라인 인식 시스템은 IBM Pentium 60 PC 상의 MS-Windows 3.1 환경에서 구현하였다. 프로그램은 Borland C++ 4.0을 사용하여 구현하였으며, 실험 대상은 KAIST 한글 흘림체 데이터를 사용하였다. 이 온라인 흘림체 한글 데이터는 WACOM SD-510C 디지털타이저로 50명이 입력하였으며, 필기시 제약 조건을 가하지 않았다.

5.1 데이터 코딩과 모형 훈련

디지털타이저에서 글자를 입력받은 후, 펜 궤적 샘플링 도중 발생하는 잡음을 줄여 주는 억제거, 평활화의 전처리 과정[8]을 수행한다. 전처리에 이어 일정치 않은 필기 속도에 의해 발생하는 불균등한 샘플링 점들의 분포를 균일하게 하고 자소와 자소내, 자소간의 공백을 연결하는 획정보를 얻기 위해 등간격으로 재구성한다(resample). 글자 크기의 비율에 따른 적절한 간격으로 재구성하면 동일한 글자는 글씨의 크기와 상관없이 비슷한 수의 점으로 변환할 수 있다. 이때 재구성의 간격은 글씨의 원형을 왜곡시키지 않는 범위 내에서 결정해야 한다. 본 실험에서는 80개로 재구성하고, 그 후 샘플링된 점과 점사이의 방향 정보를 사용하여 연속적인 8방향 코드를 생성하게 된다.

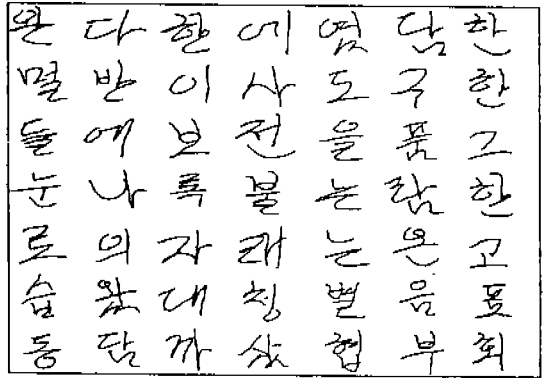
이상에서 생성된 데이터를 사용하여 개별 자소 HMM과 연결책 HMM을 훈련시킨다. HMM은 많은 확률 파라미터로 인하여 대량의 샘플 데이터를 필요로 한다. 이 데이터는 하나의 완전한 흘림체 글자이므로 자소 사이를 구분시키는 경계점을 찾기가 어렵기 때문에 이를 사용하여 자소 모형과 연결책 모형을 훈련시키는 것은 부적절하다. 따라서 개별적 모델의 훈련을 위해 수작업으로 글자 내에서 자소간의 경계점

을 표시해 주어야 한다. 이 경계점은 해당 자소가 기본꼴에 가장 충실하도록 선택되어야 한다. 이것은 각 자소 모형이 이를 포함하는 모든 글자에서의 대표적 자소 모형이 되기 위해 기본꼴과의 편차를 최소로 해야 하는데 있다.

5.2 글자 인식

글자 인식의 단계에서도 모형 훈련과 마찬가지로 동일한 방법으로 데이터를 얻는다. 이를 레벨 빌딩 알고리즘의 입력으로 사용하며 (그림 2)의 네트워크 상에서 하나의 글자를 인식하게 된다. 인식 네트워크에서 레벨 빌딩 알고리즘의 수행 속도는 한 모형 내에서의 비터비 알고리즘 수행 속도와 네트워크 내의 전체 HMM 수의 곱으로 계산되어지며, Pentium 60 컴퓨터에서 한 글자를 인식하는데 0.72 초의 시간이 소요되었다. 그리고 시험용 흘림체 데이터 15,000에 대한 정인식율은 90%였다. (그림 6)은 실험에서 사용한 흘림체 데이터의 예이며 <표 1>은 구체적인 인식 결과이다. 인식 오류는 비정상적으로 필기된 문자, 모

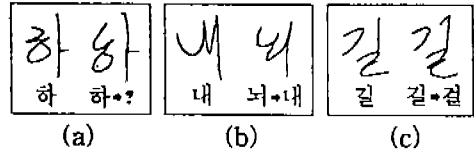
호한 문자, 과도한 훅(hook)을 포함하는 문자에 대해서 발생하였다. (그림 7)은 각 오류에 대한 예이며 <표 2>는 오류 유형별 비율이다.



(그림 6) 실험에 사용된 글자
(Fig. 6) Sample characters for test

<표 1> 인식율
(Table 1) Recognition rate

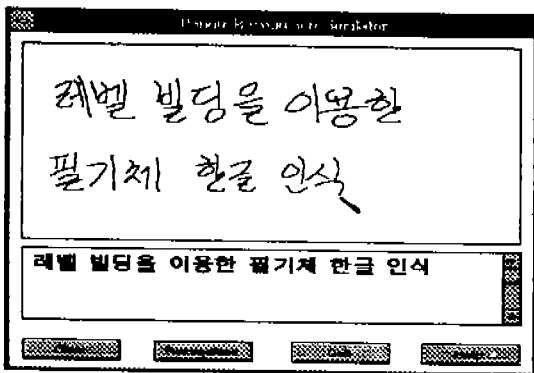
실험 데이터 문자수	오인식		미인식		정인식율	
	#	%	#	%	#	%
학습 데이터	5,000	94	1.9	0	00	4906 98.1
시험 데이터	15,000	1205	8.0	268	1.8	13527 90.2



(그림 7) 인식 오류의 예
(Fig. 7) Examples of the recognition error

<표 2> 인식 오류의 유형별 비율
(Table 2) Error ratio according to error types

오류 형태	오류 문자수	비율 (%)
비정상적 필기	919	62.4
모호성	417	28.3
과도한 훅	137	9.3
합계	1473	100.0



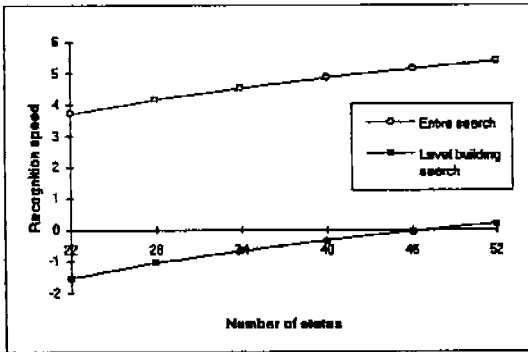
(그림 5) 인식 시스템의 구현
(Fig. 5) System layout

<표 3>은 각 자소 모형에 할당되어진 HMM의 상태의 수에 따른 인식속도와 인식율을 보여준다. 이 표에서 나타나는 바와 같이 연결확 모형을 세 상태로 고정하였을 경우 네 번째가 한글 인식에 가장 적합함을 알 수 있다. (그림 8)은 인식 네트워크 전체를 탐색하는 방법과 레벨 빌딩을 이용한 탐색 방법에서의 인

식속도의 차이를 나타낸다. 그래프의 x축은 <표 3>의 상태 수를 나타내고 인식속도는 log-scale 되었다. 레벨 빌딩 알고리즘은 네트워크 경로수에 의존하는 전체 탐색 방법에 비해 전체 네트워크를 구성하는 HMM의 수에 의존하므로 보다 빠른 인식속도를 보여준다.

<표 3> 인식속도와 인식율
<Table 3> Recognition speed and rate

상태수			인식 속도 (sec/char)	인식율(%)
초성	중성	종성		
4	9	9	0.2178	81.45
6	11	11	0.3528	83.56
8	13	13	0.5202	88.61
10	15	15	0.7231	90.18
12	17	17	0.9522	89.12
14	19	19	1.2168	88.94



(그림 8) 두 가지 탐색 방법의 인식속도
(Fig. 8) Recognition speed of two search method

그림 6은 한글 “다”자를 제안된 레벨 빌딩을 이용하여 인식하는 과정은 보여준다. 여기에서 “Level”은 전체 인식 네트워크 내에서 자소 HMM 또는 연결획 HMM의 위치를 나타내고, \hat{B} 은 역추적 포인터를 나타내고, \hat{W} 은 적합한 자소 또는 연결획 모델을 나타낸다. 그리고 \hat{P} 은 현재 프레임까지의 각 레벨에 대한 적합한 확률을 나타내고 가장 큰 확률값을 출력하는 모델이 적합 모델로 선택된다. 프레임 $T(t=80)$ 에서 레벨 3과 레벨 5의 적합 확률을 비교하면 $3.162e-28$ 과 $3.783e-41$ 으로 레벨 3의 적합 확률이 크므로 인식 네

트워크에 적합한 한글 모형은 초성-연결획-중성 구조를 가진다는 것을 알 수 있다. 레벨 3에서의 적합한 중성 모형은 $\hat{W}(3, T)=0/“ㄷ”$ 이 되고 이전 레벨의 프레임에 가리키는 역추적 포인터의 값은 $\hat{B}(3, T)=48$ 이 된다. 레벨 2에서 48번째 프레임에 적합한 모형은 $\hat{W}(2, 48)$ 의 값 3으로 “ㄱ”으로 방향을 가리키는 연결획 모형이 되고 이전 레벨의 프레임 번호는 34가 된다. 이와 같은 방법으로 레벨 1까지 반복하면 초성은 4/“ㄷ”로 선택되어 완전한 글자 “다”로 인식된다.

각 레벨은 각 자소와 자소를 연결하는 연결획을 가리키고 \hat{B} 은 일련의 관측값에서 각 레벨이 차지하는 범위를 포함한다. 이러한 정보를 이용하여 전체 입력 관측값에서 자소 및 연결획의 위치를 찾아 입력된 글자를 개별 자소로 분할할 수 있다. (그림 9)에서 레벨 3에서 1까지 \hat{B} 값을 추적하면 각각 48, 34, 0의 값을 가지고 있고, 입력 관측값 중 1에서 33까지는 초성 “ㄷ”, 49에서 80까지는 중성 “ㄷ”가 됨을 알 수 있다.

6. 결 론

본 논문에서는 인식속도를 개선하기 위해 레벨 빌딩 탐색 알고리즘을 적용한 홀림체 한글의 온라인 인식 시스템을 제안하였다. 한글은 2차원 상에서 초성-중성-종성의 순으로 자소가 결합되어 하나의 글자를 형성한다. 이러한 한글의 제자 원리에 의한 인식 네트워크를 자소 단위의 HMM과 연결획 HMM으로 구성하고 최적의 네트워크 탐색 알고리즘인 레벨 빌딩을 수정하여 적용하였다.

레벨 빌딩 알고리즘은 각 레벨별 계산 과정에서 자소 또는 연결획 모형을 결정지어 가므로 각 레벨에서 후보로 주어진 모형에만 비터비 알고리즘이 적용되어진다. 따라서 11,172개의 경로를 가지는 인식 네트워크 탐색에서 경로수에 비의존적이고 개별 자소 HMM과 연결획 HMM의 수에만 의존하는 시간복잡도를 가진다.

인식 실험을 통하여 제안된 시스템은 홀림체 한글의 모델링에 탁월한 HMM의 장점을 수용하고, 더불어 인식속도의 문제점을 개선하여, 자소간의 홀림이 강하고 실시간 응답을 필요로 하는 온라인 한글 인식에 적합함을 확인하였다. 제안된 시스템은 우수한 성능을 보였지만, 보다 강력한 혹 제거 알고리즘, 레벨

Level = 1									
$\hat{P}(1, t)$									
0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	6.153e-21
6.638e-22	1.415e-18	2.176e-15	2.218e-14	2.016e-14	1.832e-14	1.664e-14	1.512e-14	5.672e-18	3.915e-21
2.463e-20	4.537e-20	2.695e-17	1.189e-17	5.246e-18	1.736e-18	1.280e-15	5.902e-12	3.798e-12	9.142e-13
5.488e-13	3.295e-13	1.978e-13	4.441e-14	2.666e-14	4.676e-15	8.200e-16	1.438e-16	2.522e-17	4.423e-18
7.756e-19	1.360e-19	2.385e-20	4.183e-21	7.335e-22	1.286e-22	2.256e-23	3.956e-24	6.938e-25	3.469e-28
1.734e-31	8.672e-35	4.336e-38	2.168e-41	1.084e-44	5.420e-48	2.710e-51	1.355e-54	6.775e-58	3.388e-61
1.694e-64	8.469e-68	4.235e-71	2.117e-74	1.059e-77	5.293e-81	2.647e-84	4.641e-85	8.139e-86	1.427e-86
2.503e-87	4.390e-88	7.698e-89	1.350e-89	2.368e-90	1.421e-90	8.533e-91	5.123e-91	3.076e-91	1.847e-91
$\hat{B}(1, t)$					$\hat{W}(1, t)$				
0	0	0	0	0	0	0	0	0	3
0	0	0	0	0	0	0	0	0	5
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	4
Level = 2									
$\hat{P}(2, t)$									
0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
0.000e+00	0.000e+00	5.847e-22	5.649e-22	1.345e-19	2.068e-16	2.108e-15	2.036e-15	4.579e-18	4.161e-18
3.400e-17	3.399e-17	3.397e-17	3.396e-17	3.393e-17	1.011e-20	8.918e-21	7.869e-21	3.472e-21	2.416e-18
3.031e-14	1.364e-14	6.139e-15	3.372e-15	1.692e-15	1.484e-16	2.221e-17	6.666e-18	1.169e-18	2.289e-19
1.881e-19	1.546e-19	1.271e-19	1.045e-19	8.587e-20	7.058e-20	5.801e-20	4.768e-20	3.919e-20	1.123e-22
4.637e-26	1.560e-27	1.559e-27	1.558e-27	1.557e-27	1.557e-27	1.556e-27	1.555e-27	1.554e-27	1.554e-27
1.553e-27	1.552e-27	1.551e-27	1.550e-27	1.550e-27	1.549e-27	1.548e-27	1.161e-30	8.708e-34	6.531e-37
4.896e-40	3.674e-43	2.755e-46	2.066e-49	1.734e-51	3.469e-52	6.938e-53	1.388e-53	2.775e-54	5.550e-55
$\hat{B}(2, t)$					$\hat{W}(2, t)$				
0	0	0	0	0	0	0	0	0	0
0	9	9	9	11	12	13	13	15	16
17	17	17	17	17	22	23	24	25	26
27	27	27	27	31	32	33	34	35	34
34	34	34	34	34	34	34	34	34	34
34	48	48	48	48	48	48	48	48	48
48	48	48	48	48	48	48	48	48	48
48	48	48	48	48	48	48	48	48	48
Level = 3									
$\hat{P}(3, t)$									
0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
0.000e+00	0.000e+00	6.482e-40	1.128e-37	1.698e-33	1.114e-30	2.422e-30	1.589e-25	1.980e-22	2.052e-23
1.636e-23	3.319e-23	3.190e-23	1.077e-24	8.580e-25	1.413e-27	7.480e-30	3.960e-32	2.097e-34	6.109e-30
1.740e-27	1.454e-27	5.581e-28	2.143e-28	8.226e-29	3.158e-29	1.213e-29	4.655e-30	1.787e-30	7.101e-28
2.207e-26	2.175e-26	4.245e-25	1.286e-22	1.226e-22	1.168e-22	1.113e-22	1.061e-22	1.011e-22	9.634e-23
9.181e-23	8.749e-23	8.338e-23	7.946e-23	7.572e-23	7.216e-23	6.877e-23	5.157e-26	3.868e-29	5.656e-32
4.576e-29	2.163e-29	1.022e-29	4.831e-30	2.283e-30	7.771e-28	6.194e-28	4.936e-28	3.934e-28	3.136e-28
$\hat{B}(3, t)$					$\hat{W}(3, t)$				
0	0	0	0	0	0	0	0	0	0
0	0	0	0	12	13	14	15	16	17
17	17	12	13	14	15	16	17	17	17
17	17	17	17	17	17	17	17	29	29
30	30	30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30	30	48
48	48	48	48	48	48	48	48	48	48
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	2	2	9	9	9	9	0	0
0	4	4	0	0	4	4	4	4	13
3	13	13	13	13	13	13	13	13	13
3	3	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

(그림 9) Level-building을 이용한 한글 "다"의 인식
 (Fig. 9) Recognition of Korean character '다'(da) using level building

빌딩을 통한 후보 생성, 무엇보다도 학습 데이터의 증가를 통하여 성능 향상을 기대할 수 있을 것이다. 더불어 한글과 유사한 구조를 가지는 한자에 대한 제안된 방법의 응용이 연구 과제로 남아 있다.

참 고 문 헌

[1] L. R. Rabiner, "Tutorial on Hidden Markov Model and Selected Application in Speech Recognition," in Proc. of the IEEE Vol. 77, No. 2, pp. 257-285, Feb. 1989.

[2] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition," The Bell System Technical Journal, Vol. 62, No. 4, pp. 1075-1105, April 1983.

[3] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Function of a Markov Process to Automatic Speech Recognition," The Bell System Technical Journal, Vol. 62, No. 4, pp. 1035-1074, April 1983.

[4] 조성배, 김진형, "온라인 필기문자의 인식을 위한 은닉 마르코프 모델과 신경망 분류기의 결합," 한국정보과학회논문지, 제20권, 제3호, pp. 328-337, 1993. 3.

[5] 신봉기, 김진형, "은닉 마르코프 모델을 이용한 온라인 한글 인식," 제1회 문자 인식 워크숍 논문집, pp. 189-194, 1993. 5.

[6] 신봉기, 김진형, "은닉 마르코프 모델 네트워크에 의한 온라인 흘림 필기 한글 인식," 한국정보과학회논문지, 제21권, 제9호, pp. 1737-1745, 1994. 1.

[7] 김병기, 양종렬, 배진학, 김향준, "흘림획을 수용하는 온라인 한글인식," 제1회 문자 인식 워크숍 논문집, pp. 185-188, 1993. 5.

[8] P. K. Kim, J. Y. Yang, and H. J. Kim, "On-line Cursive Korean Character Recognition using Extended Primitive Strokes," in Proc. of 3rd PRICAI(Pacific Rim International Conference on A. I.), pp. 816-821, Aug. 1994.

[9] Chinmoy B. Bose and Shyh-Shiaw Kuo, "Connected and Degraded Text Recognition using Hidden Markov Model," Pattern Recognition, Vol. 27, No. 10, pp. 1345-1363, 1994.

[10] Cory S. Myers and L. R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," IEEE Trans. on Acoust., Speech, Signal Processing, Vol. ASSP-29, No. 2, pp. 284-297, April 1981.

[11] Cory S. Myers and S. E. Levinson, "Speaker Independent Connected Word Recognition Using a Syntax-Directed Dynamic Programming Procedure," IEEE Trans. on Acoust., Speech, Signal Processing, Vol. ASSP-30, No. 4, pp. 561-565, Aug. 1982.

[12] L. R. Rabiner and S. E. Levinson, "A Speaker-Independent, Syntax-Directed, Connected Word Recognition System Based on Hidden Markov Models and Level Building," IEEE Trans. on Acoust., Speech, Signal Processing, Vol. ASSP-33, pp. 561-573, June 1985.



김 상 균

1991년 경북대학교 통계학과 졸업(학사)
 1994년 경북대학교 대학원 컴퓨터공학과(석사)
 1994년~현재 경북대학교 대학원 컴퓨터공학과 (박사과정)

관심분야: 패턴인식



김 경 현

1995년 경북대학교 통계학과 졸업(학사)
 1995년~현재 경북대학교 대학원 컴퓨터공학과 (석사과정)

관심분야: 패턴인식



이 종 국

- 1989년 경북대학교 통계학과 졸업(학사)
- 1991년 경북대학교 대학원 컴퓨터공학과(석사)
- 1995년 경북대학교 대학원 컴퓨터공학과(박사)
- 1993년~1996년 동서대학교 컴퓨터공학과 전임강사

1996년~현재 안동대학교 컴퓨터공학과 전임강사
관심분야: 패턴인식, 게임이론, 한국어정보처리



김 항 준

- 1977년 서울대학교 전기공학과 졸업(학사)
- 1979년 KAIST 전기 및 전자(석사)
- 1980년~현재 경북대학교 컴퓨터공학과 교수

관심분야: 인공지능, 문자인식, 음성인식, 병렬처리, 컴퓨터 구조 등



이 재 욱

- 1975년 경북대학교 전자공학과 졸업(학사)
- 1981년 동아대학교 전자공학과(석사)
- 1989년 동아대학교 전자공학과(박사)
- 1992년~현재 동서대학교 컴퓨터공학과 부교수

관심분야: 최적화, 유전자이론, 퍼지이론 등