HMM 네트워크 기반의 한글 인식기를 위한 구조 특성열의 적용

하 진 영

강원대학교 컴퓨터공학과
강원도 춘천시 효자2동 192-1 우: 200-701
jyha@cc.kangwon.ac.kr

Application of Structure Code Sequence for HMM Network-Based Hangul Recognizer

Jin-Young Ha
Department of Computer Engineering,
Kangwon National University

요약

온라인 필기 한글 인식 연구 중 HMM 네트워크를 기반으로 한 방법이 흘려 쓴 한글 인식에 있어서 우수한 성능을 보여주고 있다. 하지만, 또박또박 쓴 정서체 한글 인식에 대해서는 때때로 예측하지 못한 결과를 출력하기도 한다. 필기자가 정성 들여 필기했을 경우 보다 일관성 있는 인식 결과를 출력할 수 있는 것이 중요하다. 또한 계산 능력이 떨어질 수밖에 없는 휴대용 컴퓨터에서의 활용을 위해 인식 속도의 향상도 필요하다. 따라서 본 논문에서는 정서체 인식률 및 인식 속도 개선을 위해 16-방향 체인코드 대신 구조적 정보를 포함하는 새로운 코딩 방식을 제안하고자 한다.

1 서론[1)]

온라인 필기 문자 인식에 대한 연구는 지난 30여 년 동안의 많은 연구의 결실로 실용화에 근접하게 되었다. 특히 인간과 컴퓨터의 상호작용에 대한 여러 학문 분야에서의 연구 진척은 보다 자연스럽고 편리한 입력 방법에 대한 요구를 증대시켰고, PDA와 휴대용 PC의 소량화, 경량화에 많은 장애가 되어 온 키보드를 대체할 입력수단에 대한 요구가 지속되고 있다[1-2].

해외에서의 온라인 필기 인식에 대한 많은 연구는 주로 한자와 일본어의 인식을 위한 것이었다. 1980년대부터 1990년대에 이르기까지 다수의 시제품과 상업용 시스템이 소개되었는데, 상당수의 시스템들이 필기에 많은 제약을 가해 필기에 융통성을 기하기 어려웠다[1-3].

한글 인식에 대한 연구는 1960년대 말부터 주로 일본과 국내의 대학을 중심으로 진행되어 왔으나, 실용적인 문서 인식 시스템이나 펜컴퓨터 등의 개발을 염두에 두고 진행된 것은 1990년대 이후부터이다. 온라인 필기 인식은 얼마나 자유로운 필기를 허용하느냐에 따라 인식의 난이도가 결정된다. 연구 초기에는 자소가 분리된 문자만을 인식 대상으로 삼았지만, 최근의 연구에서는 무제약 필기 인식에 대한 연구가 활발하다.

한글 인식 초기에는 획 정합 방법을 주로 사용한 구문론적 인식 방법에 근간을 두고 있었는데[3], 최근에는 신경망을 이용한 방법 및 HMM(hidden Markov model)을 이용한 확률 통계적 방법이 우수한 성능을 보인다고 알려져 있다. 특히 HMM은 음성인식 분야에서 많이 활용되어 변형이 심한 대량의 데이터의 모델링에 뛰어난 성능을 보이고 있어서 흘려 쓴 문자 인식 분야에 중요한 기여를 하고 있다[4-8].

국내의 연구 중 주목할 만한 것은 한국과학기술원 인공지능연구센터를 중심으로 대학, 연구소, 기업의 컨소시엄 형태로 진행된 노트패드 프로젝트의 연구 결과이다. 이 연구에서는 온라인 한글 인식 연구에 한 획을 긋는 중요한 업적을 남겼다. BongNet라고 명명된 인식 방법론은, 그 이전의 대부분의 연구에서 해결하지 못했던 흘려 쓴 한글 인식을 위한 HMM기반의 새로운 통계적 방법론을 제안하였는데, 흘림이 심한 한글 자소는 물론 한글의 초성, 중성, 종성이 모두 연결된 필기 문자의 인식도 가능한 방법론이었다[4-8]. 그 후 계속된 연구에서 한글 자소의 구조적 특성을 추가한 BongNet+와 연속 한글 인식을 위한 Circular BongNet, 그리고 한글 및 영숫자 인식도 모드 전환 없이 인식이 가능한 Unified BongNet 등의 연구가 계속되어 왔고, 한글 사전의 적용도

1) 본 연구는 한국전자통신연구원의 지원과 정보통신부의 정보통신연구관리단의 지원에 의한 강원대학교 멀티미디어 특화연구센터의 지원을 일부 받았음을 밝힙니다.

추가되었다[4-9].

BongNet의 여러 장점에도 불구하고 몇 가지 문제점이 있는데, 첫째는 HMM Network의 입력으로 16-방향 체인코드를 사용한 것이다. 흘림이 심한 필기 문자의 인식에는 상대적으로 뛰어난 성능을 보였지만, 정서체의 필기 문자의 인식 시에는 오히려 다른 방법론에 비해 장점을 발휘할 수 없었다. 특히 필기자가 정성 들여 필기했을 경우에도 때때로 예측하지 못한 결과를 출력하는 경우도 발생하여, 일관성 있는 인식 결과의 출력이 요구된다. 둘째는 인식 속도의 향상이 필요하다는 것이다. CPU의 성능이 우수하지 못한 소형의 휴대용 단말기 또는 PDA, 펜 컴퓨터 등에서 사용하려면 상당한 인식 속도의 향상이 필요하다.

2 BongNet

BongNet은 각 자소 모델과 연결획 모델을 기반으로, 이를 한글의 글자 조합 원리를 이용하여 연결함으로써 한글 필기를 모델링한 네트워크 구조이다. 한글에는 초성 19자, 중성 21자, 종성 27자 등 총 67개의 자소가 있으며, 각각의 자소에 발생하는 필기 습관, 필기 상태, 전체 자소의 결합 형태에 따른 다양한 변형을 흡수하기 위하여 각각을 은닉 마르코프 모델로 모델링 하였다. 또한 한글 필기 시에 발생하는 자소 간의 흘림을 위한 연결획(ligature) 모델 개념을 도입하여, 연결획의 시작부분과 마침 부분의 상대적인 위치에 따라 모델을 나누어 각각을 은닉 마르코프 모델로 모델링 하였다.

한글을 필기할 때에는 초성, 중성, 종성의 순서로 필기한다. 추가적으로 연결획을 고려하면, 한글 한 음절은

(초성)+(연결획)+(중성)

(초성)+(연결획)+(중성)+(연결획)+(종성)

과 같은 확장된 자소열로 표현할 수 있다. 이러한 방법으로 각각의 자소 모델과 연결획 모델을 연결한 것이 BongNet이다. 네트워크의 시작 노드에서 종료 노드로의 각 경로는 하나의 글자에 해당한다. 인식은 주어진 글씨에 대해서 그 글씨가 표현하고자 하는 글자의 경로를 찾는 작업이라고 할 수 있다. 즉 통계적으로 가장 유사한 경로를 찾는 문제로, Viterbi 알고리즘을 사용하여 효율적으로 구할 수 있다[5].

봉네트의 특징은 아래와 같이 정리할 수 있다[7].

- HMM으로 다양한 필기를 모델링 하였다.
- 연결획 모델의 도입으로 자소 간에 흘려 쓴 글씨도 수용하였다.
- 통계 모델로서, 인식에 관련된 여러 가지 정보의 결합이 용이하다.
- 네트워크상의 각 경로는 한 글자에 대응되고, 입력 코드열에 비선형 배열된다.
- 네트워크 디코딩을 통하여 인식과 자소 분할이 동시에 일어난다.
- 일관된 표현 구조, 일관된 계산 구조를 갖고 있다.

3 구조적 정보를 포함하는 코드열

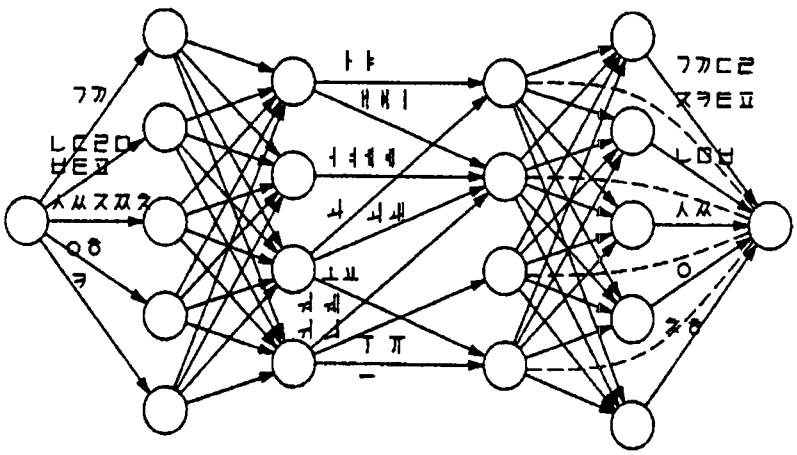
3.1 특징점의 추출

그래픽 테블렛으로부터 입력되는 데이터는 먼저 거친점 제거(wild point reduction)와 평활화(smoothing) 과정을 거친 후 계산량을 줄이기 위해 일정한 간격으로 다시 샘플링한다. 각 획의 시작점으로부터 끝점까지의 각 점 사이의 각도의 변화에 따라 <그림 2>와 같이 특징점을 추출한다.

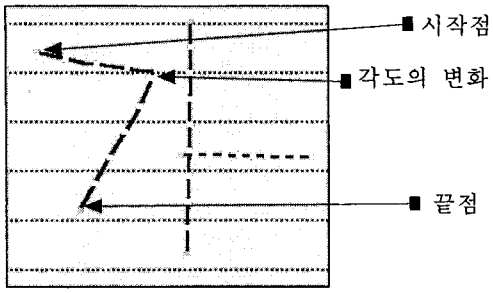
이와 같은 특징점 추출 방법은 정서체 뿐만 아니라 흘려 쓴 대부분의 필기 한글의 자소 간 경계점을 누락시키지 않고 찾아낼 수 있다. <그림 3>은 필기 한글의 특징점 추출 예를 보여주고 있다.

3.2 특징벡터의 생성

구조적 정보를 포함하는 코드열을 생성하기



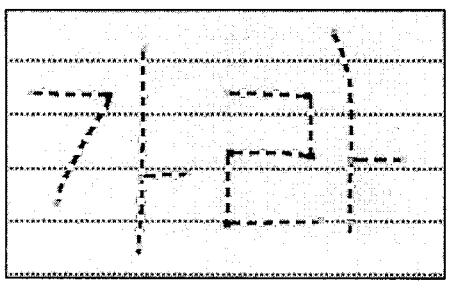
<그림 1> BongNet의 구조



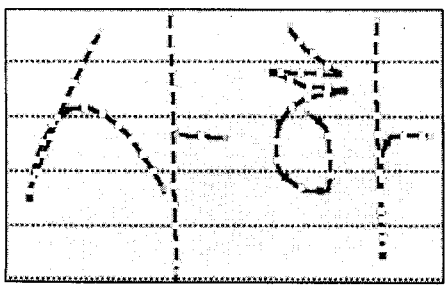
<그림 2> 특징점의 추출

위해, 각도의 변화에 따른 특징점 사이의 각 부분획으로부터 다음과 같은 5차원의 벡터를 생성한다.

- Distance: 특징점 사이의 각 점 간 거리의 합 (입력문자의 높이를 100으로 정규화)
- Straightness: 부분획의 곧은 정도로서 시작점과 끝점 사이의 직선 거리를 누적거리로 나눈 비율(완전한 직선일 경우 100%)
- Direction: 시작점에서 끝점으로 향하는 각도(0도~360도)
- Real: 실제획이면 1, 가상획이면 0
- Rotation: 부분획의 굴곡 방향을 시계방향(1), 반시계방향(-1), 그리고 무방향(0)의 3가지로 분류 (<그림 4> 참조)

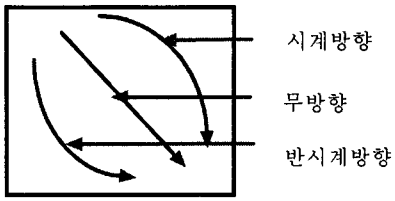


(a)



(b)

<그림 3> 특징점 추출의 예



<그림 4> rotation

<표 1 > 특징 벡터의 예

Dis-tance	Straight-ness	Direc-tion	Real	Rota-tion
53.755	95.529	113.876	1	-1
35.638	94.922	353.055	1	-1
35.784	97.420	290.420	1	-1
51.686	89.838	198.337	1	-1
62.849	100.000	353.956	0	0
100.159	99.909	87.894	1	0

3.3 특징벡터의 클러스터링(clustering)

K-menas clustering algorithm은 입력 데이터를 정해진 개수의 클러스트로 만들 때 널리 사용되는 것으로 그 알고리즘은 다음과 같다[10].

Algorithm: K-menas clustering

1. Choose the nubmer of classes, K.
2. Choose m_1, m_2,...,m_K. These are initial guesses.
3. Classify each x_k.
4. Recompute the estimates for m_i using the results of 3.
5. If the m_i are consistent, STOP; otherwise go to step 1,2, or 3.

추출된 특징 벡터들을 모두 모아 클러스터링해서 64개의 클러스터 중심(cluster center)을 생성하는데, 부분획에 대해 여러 각도와 길이를 잘 반영하도록 하기 위해 씨앗(seed)을 다음과 같이 적절히 할당한다.

① 보통 길이 실제획의 각 방향에 대해(24도 각도 간격으로 15개)

② 짧은 길이 실제획의 각 방향에 대해(24도 각도 간격으로 15개)

③ 가상획의 각 방향에 대해(15개)

④ 기타(19개)

클러스터 중심과 입력 벡터와의 거리 계산 시 각 벡터의 요소에 가중치를 두어 중요한 요소의 반영 비율을 크게하고, 또한 상대적으로 값 차이가 적은 요소를 보정해 주도록 하였다. 본 연구에

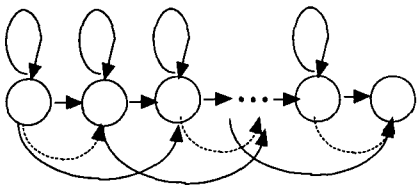
서 사용된 가중치는 (1.0, 15.0, 10.0, 100000.0, 100.0)이다. 4번째 가중치는 실제획과 가상획을 구분하기 위해 사용되는 값인데 크게 하여 다른 요소에 의해 거리 순서가 바뀌지 않게 하였고, 마지막 가중치는 rotation을 나타내는 값인데 값의 범위가 -1 ~ 1 사이이기 때문에 다른 요소들에 비해 상대적 차이가 작은 것을 보정해 주기 위해 큰 값을 사용했다. 또한 length나 straightness보다는 direction의 가중치를 크게 하여 부분획의 각도의 차이를 크게 반영하게 하였다.

3.4 구조 코드의 생성

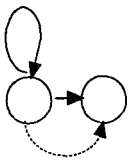
특징점에 의해 분할된 부분획으로부터 추출한 각 특징 벡터와 3.3절에서 설명한 것 같이 미리 만들어 놓은 클러스터 중심을 비교하여 해당 클러스터 중심의 번호를 구조 코드로 사용한다. 이렇게 생성된 구조 코드는 체인 코드와는 달리 구조적 정보를 포함하게 된다.

4 HMM 훈련과 인식

한글 각 자소와 자소간의 연결획 데이터를 구조 코드로 변환한 다음 각각을 HMM으로 모델링하였다. 모델 훈련 후 <그림 1>과 같이 HMM 네트워크를 구성한 후 입력에 대해 Viterbi 알고리즘을 이용하여 가장 높은 확률을 갖는 경로를 찾아 한글을 인식했다. 모델 훈련과 인식 방법은 논문 [4]와 동일한 방법을 이용했다.



(a) 자소 HMM



(b) 연결획 HMM

<그림 5> HMM 구조

4.1 HMM의 훈련

한글 자소 모델과 연결획 모델의 구조는 다음과 같다. <그림 5>에서 파선은 null transition을 나타낸다. 자소 HMM의 상태(state) 수는 자소마다 달리 했고, 연결획 HMM의 상태 수는 2로 했다. 자소 HMM 중 가장 적은 상태 수를 갖는 모델은 중성 '이'로 3개이고, 가장 많은 상태 수를 갖는 것은 초성 'Q'으로 20개이다. 각각의 자소에 대해, 2 ~ 20개의 상태 수를 갖는 HMM을 만든 후 입력 데이터에 대해 가장 높은 확률값을 모델링 하는 상태 수를 구했다. 연결획에 대해서는 2 ~ 5개의 상태 수에 대해 확률값을 계산하였다.

4.2 인식

인식 과정은 다음과 같다. 앞 절에서 기술한 것과 같이 필기 입력 데이터를 구조 코드열로 변환한 후 HMM의 네트워크인 BongNet에서 가장 높은 확률값을 갖는 경로를 구함으로써 인식 결과 및 각 자소의 분할 정보를 동시에 얻을 수 있다. 자세한 내용은 논문 [9]를 참조하기 바란다.

5 실험 및 결과 분석

한글 모델의 훈련과 인식 실험을 위해 KAIST에서 구축해 놓은 문자 DB를 사용하였다. 필기 문자의 획득은 WACOM SD-510C 디지타이저를 이용하였으며 총 41명이 필기한 것을 한글 자소 모델의 훈련을 위해 자소와 연결획 데이터로 분리해 놓은 것을 이용하였다. 한글 자소 훈련을 위해 총 214,176 개의 자소 데이터를 사용하였고, 연결획 모델의 훈련을 위해 총 150,341 개의 연결획 데이터를 사용하였다.

Sun Ultra-2 200MHz 워크스테이션 상에서 정서체 한글에 대해 인식률을 측정하였는데, 8명이 필기한 6,308자의 한글에 대해 95.89%의 정인식률과 28.67문자/초의 인식속도를 얻었다. 이 결과는 16-방향 체인코드를 사용할 때보다 인식률이 0.51% 높아졌고, 인식 속도는 초당 15문자 이상을 더 인식한 것으로서 2.15배의 인식속도를 보였다. 여기에서 인식속도는 CPU 시간이 아닌 사용자 시간으로 측정한 것이다.

<표 2> 정서체 필기에 대한 실험 결과

	16-방향 체인코드	구조 코드	증감
한글의 구조적 정보 미사용	95.38 % 13.33문자/초	95.89 % 28.67문자/초	+0.51% +15.34 문자/초
한글의 구조적 정보 사용	97.46 % 14.14문자/초	97.26 % 28.80문자/초	-0.20% +14.66 문자/초

BongNet에 한글의 구조적 정보를 포함해서 개선한 BongNet+에 대해서도 같은 실험을 수행한 결과 16-방향 체인코드에 비해 구조 코드의 인식률이 약간 낮게 나타났다. 이것은 한글의 구조적 정보 중 많은 부분이 16-방향 체인코드에 대해 적용되게 되어 있기 때문인 것으로 분석된다.

흘려 쓴 문자에 대해서도 인식률을 측정하였다. <표 3>에서의 결과는 본 논문에서 제안하는 구조 코드가 흘려 쓴 한글에는 적합하지 않다는

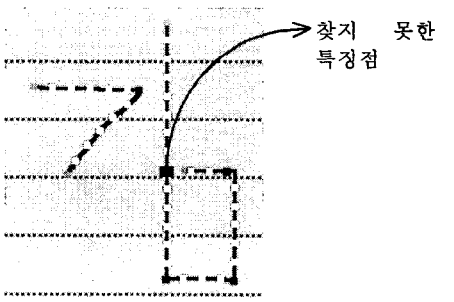
것을 보여주고 있다. 이와 같은 결과는 흘림이 심할 경우 각도의 변화에 의한 특징점 추출에 어려움이 있고, 또한 부분획이 지나치게 작은 조각들로 분할되어 일관성 있는 구조 코드열을 얻을 수 없는 것으로 분석된다.

<표 3> 흘려 쓴 한글에 대한 인식률

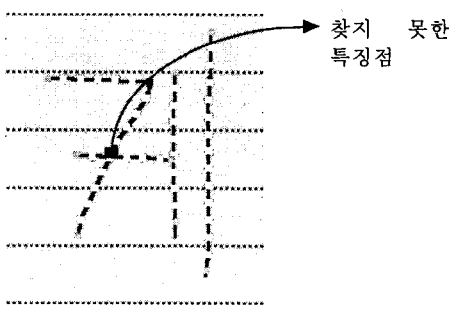
	16-방향 체인코드	구조 코드	증감
1순위 정인식률	92.25 %	85.05 %	-7.20 %
상위 5순위 인식률	95.81 %	93.73 %	-2.08 %

<그림 6> 은 각도 변화에 의한 특징점 추출이 실패한 경우를 보여주고 있다. 이러한 필기 유형일 경우에는 체인 코드를 사용하는 것이 더 적절하다.

실험에 사용된 필기 데이터의 일부가 <그림 7>과 <그림 8>에 나타나 있다. 정서체와 흘려 쓴 데이터에 대한 인식률의 차이는 이러한 데이터의 품질의 차이에서 기인하는 것이 크다고 분석된다.



(a) 중성에서 종성으로의 흘림으로 특징점을 찾지 못하는 경우



(b) 초성과 중성 사이의 특징점을 찾지 못하는 경우

<그림 6> 특징점 추출의 실패 예

6 결론

본 논문에서는 온라인 한글 인식을 위해 새로운 구조 코드열을 제안하였다. 이 구조 코드열은 체인 코드에 비해 코드열의 길이가 상당히 작아짐에 따라 높은 인식 속도를 이룩할 수 있었고, 또한 정서체로 필기할 경우 일관성 있는 높은 인식률을 얻을 수 있었다.

감사의 글

본 연구를 수행하는데 한국과학기술원의 김진형교수님, 이재준, 조성정의 도움과 강원대학교의 김재륜, 이상철, 이재성의 도움에 감사를 드립니다.

참고문헌

[1] 이성환, 문자인식 - 이론과 실제, 홍릉과학출판사, 1993.

[2] C.C Tappert, C.Y. Suen, and T. Wakahara, "The state of the art in on-line handwriting recognition," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, pp. 787-808, 1982.

[3] 김태균, 이은주, "한글에 적합한 핵 해석에 의한 연속 필기 한글의 on-line 인식에 관한 연구," 한국정보과학회 논문지, 제 15권 제 3호, pp. 171-181, 1988.

[4] B.-K. Shin and J.H. Kim, "Ligature Modeling for online cursive script recognition," IEEE Trans. Pattern Recognition and Machine Intelligence, Vol. 19, No. 6, pp. 623-633, 1997.

[5] L.R. Labiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, Vol. 77, pp. 257-285, 1989.

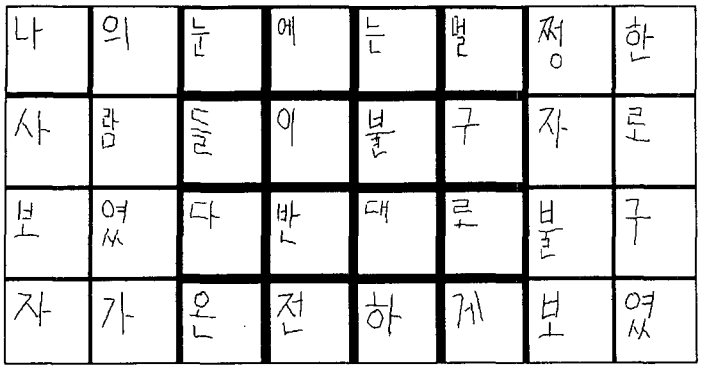
[6] 신봉기, 김진형, "봉네트 - 그 후 일 년," 제 5회 한글 및 한국어 정보처리 학술발표논문집, pp. 503-518, 1993.

[7] 이재준, 권재욱, 신봉기, 김진형, "개선된 봉네트," 제 6회 한글 및 한국어 정보처리 학술발표논문집, pp. 189-194, 1994.

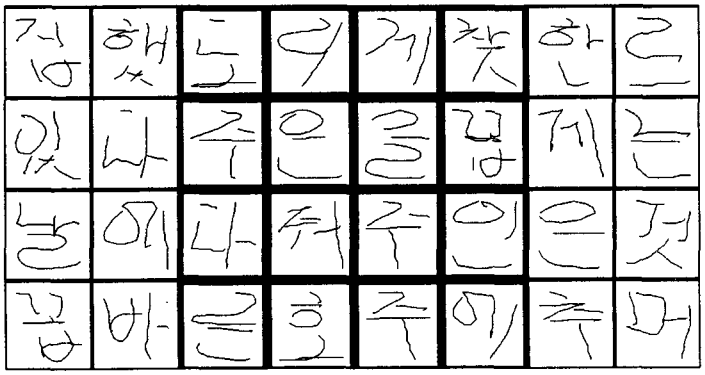
[8] B.-K. Sin, J.-Y. Ha, S.-C. Oh, and J.H. Kim, "Network-based approach to on-line cursive script recognition," IEEE Trans. Systems, Man, and Cybernetics, Vol. 28B, No. 5, 1998 (To Appear).

[9] 이재준, 김진형, "상호 연결된 HMM을 이용한 한/영 혼용필기의 온라인 인식," 인공지능, 신경망 및 퍼지시스템 춘계종합 학술대회 논문집, pp. 375-378, 1994.

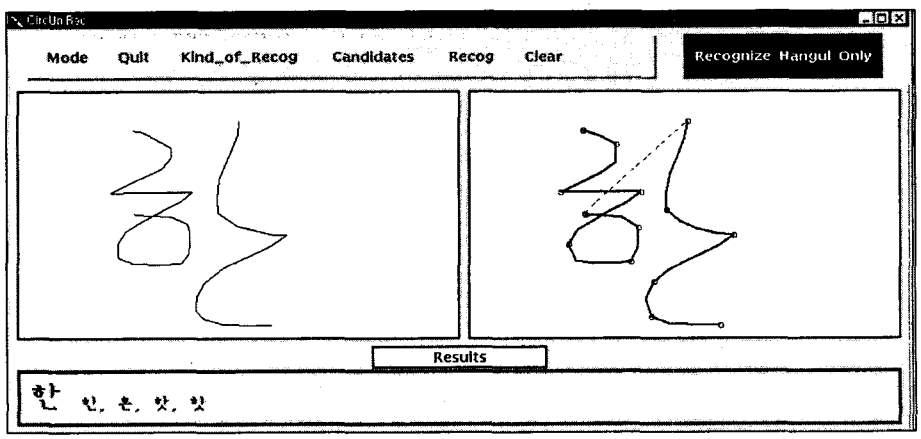
[10] R. Schalkoff, Pattern Recognition - Statistical, Structural, and Neural Approaches, John Wiley & Sons, Inc., 1992.



<그림 7> 정서체 한글의 예



<그림 8> 흘려 쓴 한글의 예



<그림 9> 필기 "한"의 인식 예

<표 4> 필기 "한"에 대한 특성 벡터와 구조 코드열

=* Distance	Straightness	Direction	Real	Rotation
18.010	99.535	24.256	1	0
39.693	90.791	139.283	1	1
39.532	99.989	359.358	1	0
45.010	98.329	144.269	1	0
35.790	89.382	15.784	1	-1
17.684	97.459	281.220	1	-1
27.311	97.044	194.411	1	-1
68.198	100.000	318.013	0	0
45.539	98.831	103.127	1	0
35.958	97.302	20.739	1	-1
45.532	99.516	149.494	1	0
18.160	93.099	94.067	1	-1
33.959	99.048	6.682	1	0

1 37 28 15 5 23 18 56 12 5 15 39 0: 구조 코드열