# 온라인 필기 한자인식을 위한 체인코드열과 구조코드열의 성능평가

김형태° 하진영
강원대학교 컴퓨터 정보통신 공학과
{ds2swd°, jyha}@kangwon.ac.kr

# Performance Evaluation of Chain-code Sequence and Structure-code Sequence for On-line Handwritten Chinese Character Recognition

Hyung-Tai Kim° Jin-Young Ha
Department of Computer, Information and Communications Engineering, Kangwon National University

요 약

본 논문에서는 보다 실용적인 온라인 한자인식기 개발을 위하여 한자 검정 능력 1급 쓰기 수준을 모두 포함하는 한자 필기 데이터로부터 16방향의 체인코드열과 부분획의 구조를 반영하는 구조코드열을 만들어 성능평가를 하였다. 인식 방법으로는 DP 매칭 방법과 HMM을 사용하여 2,362 종류의 한자에 대해 인식 실험을 하였다. 그 결과 체인코드열을 사용한 DP 매칭 방법에 의한 결과가 96.54%로 가장 높은 인식률을 보였으며, 구조코드열을 사용하여 HMM에 의한 인식실험 결과가 95.65%로 그 뒤를 이었다. 인식 속도면에서는 체인코드 보다 코드열의 길이가 짧은 구조코드열을 사용한 방법이 상대적으로 유리했고, 클래스 당 1개의 모델을 사용한 HMM에 의한 방법이 클래스 당 복수개의 모델을 사용한 DP 매칭 방법에 비해 모델의 개수가 훨씬 적기 때문에 속도 면에서 월등히 유리해 더 효율적인 인식 성능을 보인다는 결론을 내릴 수 있었다.

## 1. 서 론

최근 PDA와 태블릿(Tablet) PC등과 같은 펜 인식 기반의 정보 기기들이 널리 보급되면서 온라인 문자인식의 중요성은 점점 커지고 있다. 펜 인식 기반의 입력장치를 갖춘 기기를 사용하는 사용자들에게 있어 관심사는 자연스러운 인식 환경과 빠른 인식속도와 높은 인식률을 갖춘 인식 시스템이다. 현재 온라인 영어인식과 한글인식은 연구가 활발히 진행되어 이들 조건을 만족하는 우수한 성능을 갖춘 실용화된 시스템이 많이 있으나, 온라인 한자 인식 분야는 아직 부족한 실정이다.

한국을 비롯한 중국, 일본의 한자 문화권에서 사용하는 한자는 인식대상 글자 수가 많고, 서로 유사한 글자가 많으며, 글자를 구성하는 획의 변화가 매우 다양한 특징이 있다. 따라서 서로 다른 사용자로부터 입력된 필기 데이터에만 의존하게 되는 온라인 문자인식에서는 필기자의 습관으로 보다 다양한 종류의 획이 존재하게 된다. 이는 입력된 획에만 의존하여 대상 문자를 인식하는 온라인 한자 인식 시스템에서 인식률이 저하가 되는 원인이 된다. 이를 극복하기 위한 다양한 방법의 연구는 과거부터 계속 진행되어 왔으며, 현재 온라인 한자인식 방법은 대부분의 한자가 직선의 조합으로 이루어져 있다는 구조적인 정보를 이용한 인식 방법이 주를 이루고 있다. 그럼에도 불구하고 다양한 획의 변화와 많은 획의 수로 인한 계산 량의 증가는 문자 집합을 효율적으로 표현하지 못하게 되어 상당한 부가 작업을 요구하므로, 온라인 한자 인식을 위한 실용적인 시스템으로 발전시키기에는 큰 어려움이 있다[1].

본 논문에서 적은 연산량과 높은 인식률을 갖는 온라인 한자 인식기 개발을 위하여 실험 대상의 모든 한자(2,362자)에 대하여 사용자로부터 입력받은 필기 데이터의 획 방향 정보를 이용한 16방향의 체인 코드(chain-code)와 획의 구조 특성 열을 이용한 구조 코드(structure-code)를 만들고, 이를 DP 매칭(dynamic programming matching)과 HMM (hidden Markov model)을 사용한 온라인 한자인식 방법을 제안한다. 또한 보다 실용적인 온라인 한자 인식을 위하여 한자 검정 능력 1급수준(쓰기 배정한자 2,005자)을 모두 포함 하여 인식실험을 하였다.

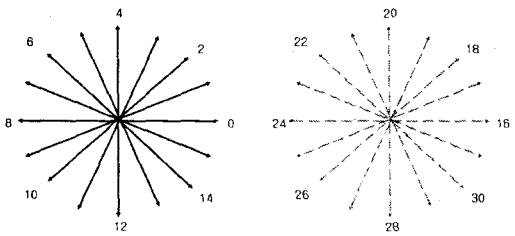
본 논문의 구성은 다음과 같다. 2절에서는 특징 추출의 방법으로 체인코드와 구조코드를 만드는 방법을 기술하고, 3절에서는 DP 매칭 방법과 HMM을 이용한 인식 방법에 대해 기술한다. 4절에서는 실험 및 결과 분석을 제시하고, 5절에서 결론을 맺는다.

## 2. 한자 인식을 위한 특징 추출

사용자로 하여금 입력된 필기 데이터로부터 특징 추출을 하기 위하여 전처리 과정은 반드시 필요하다. 이는 종이와 펜이 아닌 다른 환경으로부터 영향을 받으며, 필기자의 평소 글씨를 쓰는 습관과 속도, 필기자의 실수로 생기는 떨림, 훅(hook), 끊어진 획 등이 전처리 대상이 된다. 보다 높은 인식률을 위해서는 이를 보정해 줄 필요가 있다. 본 논문에서는 평활화(smoothing), 훅 제거(hook elimination), 난폭점 교정(wild point correction) 등의 전처리를 사용하였다.

### 2.1 체인 코드(Chain-code)

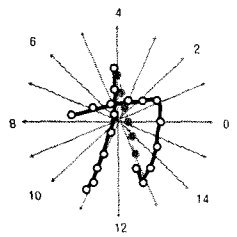
온라인 문자 인식에서 '획'이란 필기자로부터 펜을 떼지 않고 이동한 자취의 연속적인 점의 좌표를 말하며, 이를 이용하여 2차원 공간에서 획의 시작점 좌표와 끝점 좌표를 이용하여 방향 벡터를 구할 수 있다. 본 논문에서는 그림 1과 같이 360°방향을 22.5°씩 16개로 나누어 방향 코드로 표현하였고, 획과 획 사이의 가상 획을 갖는 16개의 방향 코드를 만들었다.




(a) 실제획(real) (b) 가상획(imaginary)

그림 1 체인 코드

모든 한자의 체인코드열은 연속된 방향코드의 조합으로 이루어져 있으며, 이 외에도 온라인 인식을 위하여 한자의 획수, 한자의 시작 획의 방향과 마지막 획의 방향, 체인코드열의 길이를 종합하여 그림 2와 같이 한자 인식 모델을 만들 수 있다[2].




力 (2 1 5 25)
1 2 2 2 2 15 15 12 12
11 …21 21 21 …5

/* 한자(획수,시작방향,끝방향, 코드열길이)
코드, 코드, 코드, ..... */

그림 2 한자 인식 모델

### 2.2 구조 코드(Structure-code)

구조 코드는 앞서 제시한 16방향 체인코드열의 한계를 극복하기 위한 새로운 모델로 한자 획의 구조적 특성 정보를 포함하는 새로운 코드열로 구성하였다. 구조 코드는 특징 점의 좌표를 추출 특징 벡터를 생성 후 이를 클러스터링 하여 새로운 구조 코드를 생성한다[3].

#### 2.2.1 특징 벡터의 생성

인식 대상 한자의 특징 점 좌표 사이의 부분 획으로부터 다음과 같은 정보를 추출한다.

- ▶ Distance
  : 특징점 사이의 누적 거리 (글자 높이정규화)
- ▶ Straightness
  : 부분 획의 곧은 정도: 100×( 직선거리/ 누적거리)
- ▶ Direction
  : 시작점에서 끝점으로 향하는 방향각
- ▶ Real
  : 실제 획과 가상획 (실제 획 =1, 가상 획 =0)
- ▶ Rotation
  : 부분 획의 굴곡 방향 (시계방향 =1, 반시계 =-1, 무방향 =0)

#### 2.2.1 특징 벡터의 클러스터링

부분 획에서 특징 벡터를 추출한 후 K-Means Clustering 알고리즘을 사용하여 총 64개의 클러스터를 생성하였다. 이때 부분 획의 길이와 방향에 따라 적절히 씨앗(seed)을 배정하였다. 그림 3은 클러스터링 된 구조 코드의 예이다.

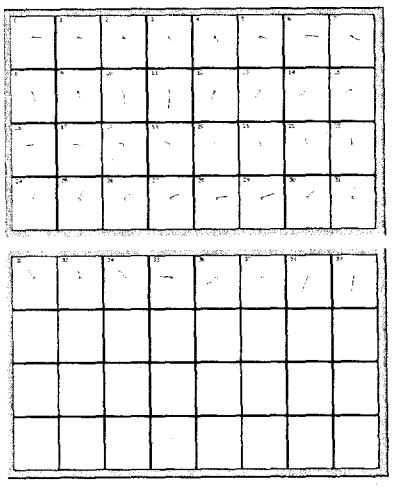


그림 3 구조 코드

클러스터 중심과 입력 벡터 사이의 거리는 식 (1)과 같이 가중치를 적용하여 계산하였다.

$$dist = \sum_{i=1}^{5} W_i \times |Input_i - Center_i| \quad (1)$$

where $Input_i$ = $i$th element of input vector
$Center_i$ = $i$th element of Cluster Center
$W_i$ = $i$th weight

2.2.2 구조 코드의 생성

구조코드는 그림 4와 같이 인식 대상 한자별 길이, 각도, 굽은 정도 등의 구조적 정보를 바탕으로 체인코드와 같이 획수, 시작 획의 방향, 마지막 획의 방향, 코드의 길이 정보와 더해져 생성된다.

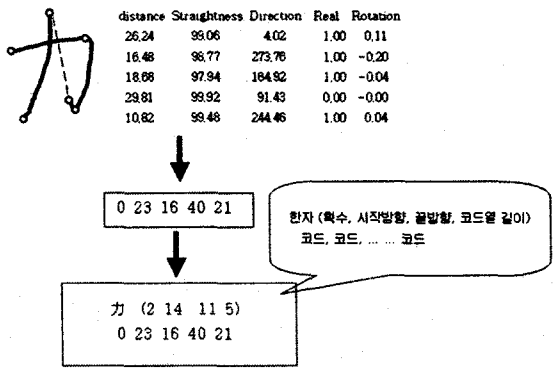



그림 4 구조코드 생성

## 3. 한자 인식 알고리즘

본 논문에서는 체인코드와 구조코드를 사용한 온라인 한자 인식의 인식률을 측정하기 위하여 DP 매칭에 의한 방법과 HMM을 사용하여 인식 실험을 하였다. 인식에 앞서 전처리는 2절의 코드 생성할 때 사용한 전처리를 그대로 사용하였으며, 체인코드와 구조코드를 각각의 방법을 적용하여 실험 하였다.

3.1 DP 매칭에 의한 인식

DP 매칭이란 동적계획법을 이용하여 두 요소간의 대응을 수행하여 유사도를 계산하는 방법이다. 유사도를 이용하면 복잡하게 변형된 패턴이 어떠한 패턴과 유사한 패턴이었는가를 판별할 수 있다. 따라서 기준이 되는 패턴과 왜곡 되거나 변형된 입력과의 매칭 정도를 평가할 수 있다. DP 매칭은 반복적 계산에 의해 이루어지며 구성이 간단하고, 획 변형과 획수가 다양한 한자인식에 있어서는 가장 안정적으로 인식할 수 있는 방법이다[4,5]. 따라서 체인코드와 구조코드를 사용하여 각각의 코드와 입력 패턴간의 특징 점들을 최적의 경로로 서로 대응시켜 패턴간의 거리가 최소가 되도록 한다. 앞서 만든 체인코드와 구조 코드를 사용하여 입력된 패턴과 미리 훈련되어 있는 패턴사이의 유사도를 측정하여 인식률을 구한다.

문자는 특징벡터의 열로써 특징을 추출하여 표현할 수 있는데 획수가 $I, J$인 문자 $A, B$의 특징방향은 다음과 같은 코드 열로써 나타낼 수 있다.

$$A = a_1, a_2, a_3, \cdots, a_I \quad (2)$$

$$B = b_1, b_2, b_3, \cdots, b_J$$

이들 열은 패턴 A의 순서로부터 패턴 B로 매칭을 하기 위한 하나의 함수로써 표현할 수 있다. 두 문자간의 유사도를 측정하는 문제에 대하여 생각하여보면 $A, B$를 $i, j$축에 놓을 때 서로 정합 시켜주는 점을 $c(k)$라 하면 정합 함수 $F$는 다음과 같다.

$$F = c(1), c(2), c(3), \cdots, c(k), \cdots, c(K) \quad (3)$$

여기서, $c(k)$는 $i, j$에서 두 패턴간의 차이이며 패턴간의 차이가 없을 때 정합 함수는 대각선 $i = j$와 일치한다. 두 특징벡터 $a_i$와 $b_i$사이의 거리는 다음과 같이 구한다.

$$d(c) = d(i, j) = \|a_i - b_i\| \quad (4)$$

그 다음 매칭 함수 $F$상에서의 가중치 합의 거리는 다음 식과 같다.

$$E(F) = \sum_{k=1}^{K} d(c(k)) w(k) \quad (5)$$

여기서, $k$는 정합 함수 $F$상에서의 점의 수를 나타낸다. 벡터 열 $A, B$를 정합 시키는 것은 두 패턴의 차이 값을 최소화 하는 것이다. $A, B$에서 정규화 된 거리는 다음 식과 같이 나타낼 수 있다.

$$D(A,B) = \min_{F}\left[\frac{\sum_{k=1}^{K} d(c(k))w(k)}{\sum_{k=1}^{K} w(k)}\right] \quad (6)$$

$w(k)$는 가중치 계수로 $E(F)$의 탄력 있는 특성을 유도하는데 도입되며 적절한 매칭 함수 $F$를 찾는데도 이용한다. 함수 $F$는 차이 값을 최적인 상태로 맞춤으로써 구할 수 있다. 분모 $\sum w(k)$는 정합 함수 $F$에서 점의 개수 $k$에 의한 영향을 보상하기 위해 사용된다. 이렇게 배열 상에 정렬된 두 벡터 열은 서로 비슷한 방향코드끼리 매칭 되어 진다고 볼 수 있다. 그러나 획의 경우 기울기의 제한 없이 지나치게 차이 나는 두 패턴을 매칭 시키면 실제적으로 맞지 않는 매칭이 될 가능성이 크기 때문에 거절(reject)할 수 있어야 한다. 한자에 있어서 방향코드는 큰 차이를 보이지 않기 때문에 이러한 제한을 두는 것이 인식률을 높일 수 있게 된다.

식 (6)을 계산하기 위한 기본적인 알고리즘은 다음과 같다.

$$g_k(c(k)) = \min_{c(k-1)}[g_{k-1}c(k-1) + d(c(k))w(k)] \quad (7)$$

앞에서 설명한 정합 함수에 제한을 두고 식(7)에 가중치 계수 $w(k)$로 대치하기 위해서는 몇 가지 실질적인 알고리즘을 유도해야 한다[2].

$$g(i,j) = \min\begin{cases} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{cases} \quad (8)$$

그림 5는 DP 매칭에 의한 예를 보여준다.

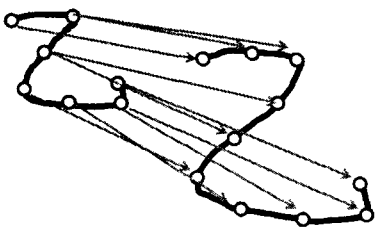


그림 5 DP 매칭

3.2 HMM을 사용한 인식

HMM은 음성인식과 문자 인식 등에서 많이 이용되는 대표적인 통계모델로서, 음성과 문자 등에서 발견되는 많은 변형들을 흡수할 수 있고, 시간에 따라 변해가는 특성을 지니는 자료(data)를 통계적 파라미터로 접근하여 잘 모델링 할 수 있다. 또한 다양한 글자체의 많은 변형을 적은 수의 모델로 표현이 가능하므로 한자인식과 같이 획의 수가 다양하고 문자집합이 방대할 경우 효과적으로 표현할 수 있다[1].

$N$을 모델의 상태수라 할 때, 각각의 상태는 $S=\{S_1, S_2, \cdots, S_n\}$으로 나타내고, $M$은 각 상태에서의 관측 심볼의 수를 의미한다. 이때, 각각의 심볼은 $V=\{v_1, v_2, \cdots, v_M\}$로 나타내고, 시간 $t$에서의 관측 심볼을 $O_t$라 한다.

$A=\{a_{ij}\}$ 는 $i$상태에서 $j$상태로 전이 확률 분포 이며, $a_{ij}$는 non-null transition이고 $a'_{ij}$는 null transition으로 $\sum_{j=1}^{N}(a_{ij}+a'_{ij})=1$ 이 된다.

$B=\{b_{ij}(x)\}$ 는 관측 심볼 확률 분포로 $i$상태에서 $j$상태로 전이 될 때 $x$번째 심볼을 관측 하게 될 확률을 말하며, $\sum_{k=1}^{M} b_{ij}(k)=1$을 만족 한다. $\pi=\{\pi_i\}$는 초기 상태의 확률 분포를 나타낸다.

각각의 $N$, $M$, $A$, $B$, $\pi$가 주어졌을 때 HMM은 $O=O_1O_2\cdots O_T$ 로 나타낼 수 있으며 모델 확률적 특성을 기술하는 파라미터 $\lambda=(A,B,\pi)$로 표현된다[6].

HMM은 단순한 마르코프 체인만으로 모델링하기 힘든 복잡한 실세계의 문제를 위와 같은 통계적 파라미터로 접근 할 수 있게 해주며, 순차적인 일련의 사건 발생을 모델링할 수 있는 상태 전이 파라미터와 각 사건의 특징을 유한개의 심볼로 대응시킬 수 있는 관찰 심볼 확률 분포의 두 가지 확률 과정의 결합으로 이루어져 있다.

그림 6는 본 실험에서 사용한 left-to-right HMM구조로서 각 상태와 상태 사이의 직선은 non-null transition이고 실선은 null transition이다.

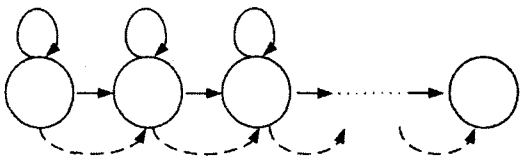


그림 6 left-to-right HMM

본 논문에서는 체인코드와 구조 코드를 이용하여 모델링을 한 후 네트워크상에서 최대 확률 값을 가지는 경로

를 찾아 인식 하게 된다.

## 4. 실험 및 결과 분석

한자 인식률을 측정하기 위하여 훈련 모델과 테스트 모델로 구분하고, 남·여 각각 15명씩 총 30명의 데이터를 PC에 연결된 태블릿을 이용하여 수집하였다. 실험에 사용된 총 한자 카테고리 수는 2,362개로, 수집된 모든 데이터는 한자능력 검증 1급 쓰기의 모든 한자를 포함하고 있다. 20명이 필기한 46,560 개의 데이터는 훈련 데이터로, 나머지 10명의 22,855 개의 데이터는 테스트 데이터로 선별하였다. 또한 수집 후 필기 오류가 포함된 데이터는 편집기로 삭제 하였다.[그림 7 참조]

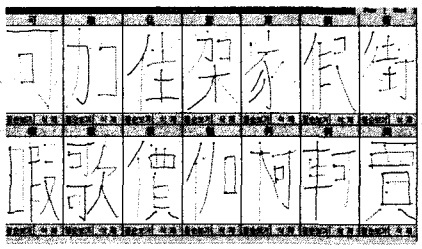


그림 7 한자 필기 데이터 편집기

인식 실험은 Pentium-4 (2.4GHz, RAM 256MB)의 PC 환경에서 수행 하였으며, 결과는 표 1과 같이 한자를 정확하게 인식한 정인식률과 10위 후보까지의 후보인식률로 나타내었다. 또한 각각의 인식방법에 따른 한 문자 당 평균 인식 시간을 나타내었다.

표 1 인식결과

| 인식구분 \ 모델 구분 / 인식률 | | Chain-code | Structure-code |
|---|---|---|---|
| DP matching | 정인식률 | 96.54% | 89.39% |
| | 10위 후보 인식률 | 99.02% | 95.21% |
| | 인식시간 | 831(ms/char) | 50(ms/char) |
| HMM | 정인식률 | 94.98% | 95.65% |
| | 10위 후보 인식률 | 98.36% | 99.07% |
| | 인식시간 | 267(ms/char) | 81(ms/char) |

# of training data = 46,560 (char)
# of test data = 22,855 (char)

인식 실험 결과 체인코드를 사용하여 DP 매칭방법이 96.54%의 인식률로 가장 좋은 인식률을 보였다. HMM은 체인 코드를 사용한 방법보다 구조 코드를 사용한 방법이 95.65%로 더 좋은 인식률을 보였다.

체인코드를 사용한 DP 매칭 방법은 모든 모델과의 유사도를 전부 계산해야 하므로 인식률은 좋으나, 속도가 느리다는 단점이 있다. 따라서 인식 속도를 높이기 위하여 20명의 훈련 데이터 중 3명을 기준으로 나머지 17명의 필기 데이터에 대하여 DP 매칭 방법에 의해 인식실험을 하였다. 그리고 오인식 된 데이터만 수집을 하여, 최초 3명의 훈련 데이터와 합쳐 새로운 훈련 데이터를 만들었다. 이는 최종 모델 데이터 수를 약 49.4% 줄일 수 있었다. 표 2는 최초 3명의 훈련 데이터에 나머지 오인식 된 데이터를 추가하여 생성된 모델을 사용한 DP 매칭 방법의 인식 결과이다.

표 2 오인식 된 데이터를 모델로 사용한 인식률

| 인식구분 \ 모델 구분 / 인식률 | | Chain-code (오인식 된 데이터) |
|---|---|---|
| DP matching | 정인식률 | 93.93% |
| | 10위 후보 인식률 | 98.96% |
| | 인식시간 | 398(ms/char) |

# of training data = 23,001 (char)
# of test data = 22,855 (char)

HMM을 사용한 방법은 인식 대상 한자 2,362개의 모델만 가지고 통계적 방법에 의한 인식을 하므로 속도면에서는 훨씬 유리하다. 또한 구조 코드열의 평균 길이가 체인코드보다 상대적으로 짧으므로 구조 코드열을 사용하여 HMM을 사용한 인식 방법이 인식기 성능면에 있어서는 가장 효과적인 것으로 분석된다.

## 5. 결론 및 향후과제

본 논문에서는 효율적인 온라인 한자 인식기 개발을 위하여 체인코드와 구조 코드를 만들고 이를 각각 DP 매칭 방법과 HMM을 사용한 방법을 제시했다. DP 매칭 방법이 인식률은 좋으나, 모든 모델의 유사도 모두 계산해야하는 단점을 가진다. HMM은 대상 한자의 최대 확률 값을 이용하여 빠른 시간과 DP 매칭방법과 비슷한 성능을 보였다. 따라서 두 가지 방법을 같이 사용 하거나 은닉마르코프 모델에서 오인식된 결과를 DP 매칭으

로 검증한다면 보다 좋은 성능의 온라인 한자 인식기를 기대할 수 있을 것이다.

Reference

[1] 김상균, 이종국, 김항준, "HMM과 레벨 빌딩 알고리즘을 이용한 흘림체 한자의 온라인 인식," *전자기술연구지*, vol. 17-2, pp.62-69, 1996.

[2] 윤병훈, 김형태, 박미나, 하진영, "문자 단위 매칭과 유닛 단위 매칭을 이용한 온라인 필기 한자인식," *정보통신 논문지*, Vol.10, pp.106-113, 2006.

[3] J.-Y. Ha, "Structure Code for HMM Network-Based Hangul Recognition," *18th International Conference on Computer Processing of Oriental Languages*, pp.165~170, ICCPOL 99, 1999.

[4] Cheng-Lin Liu, "Online Recognition of Chinese Characters : The State-of-the-Art," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 2, pp. 198-213, 2004.

[5] 진원, 김기두, "유닛 재구성 방법을 이용한 PDA용 온라인 필기체 한자인식," *정보공학회논문지*, 제 39권, SP편, 1호, pp. 97-107, 2002.

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