

논문 2004-41SP-3-23

# 핵심어 검출을 위한 단일 끝점 DTW 알고리즘

## A Single-End-Point DTW Algorithm for Keyword Spotting

최 용 선\*, 오 상 훈\*\*, 이 수 영\*\*

(Yong-Sun Choi, Sang-Hoon Oh, and Soo-Young Lee)

### 요 약

본 논문에서는 핵심어 검출 시스템을 실시간 적용이 가능한 하드웨어로 구현하기 위해 연산량이 적고 구조가 간단한 단일 끝점 DTW 방법을 제안한다. 제안된 알고리즘은 일반적 DTW가 양쪽 끝점을 요구하는데 비하여 단지 한쪽 끝점만 필요하므로 이용하기에 편리하며, 국부 검색의 연속이 전역 경로를 이루게 되므로 매우 적은 연산량을 가진다. 그리고, 제안한 단일 끝점 DTW가 보다 나은 성능을 지니도록 하기 위해 새로운 경사 가중치와 거리 측정법을 가지도록 하였다. 이외에도, 단일 끝점 DTW는 특징벡터 정규화를 적용하여 특징벡터 각각의 차원에서 데이터들이 같은 표준편차를 가지게 하며 모든 프레임이 같은 에너지를 가지도록 정규화 되었다. 또한, 주어진 학습 패턴들에 클러스터링을 적용한 후, 각 클러스터 내에서 평균을 계산하여 구한 패턴을 해당 핵심어를 대표하는 여러 개의 기준패턴으로 삼았다. 이러한 기준패턴들과 입력 음성의 특징벡터가 이미 정해진 문턱값 보다 작은 거리 내에 있을 때 핵심어는 검출된다. 제안된 알고리즘을 고립단어 음성인식과 핵심어 검출 실험에 적용하여 다른 방법을 이용한 결과보다 성능이 뛰어난 것을 확인하였다.

### Abstract

In order to implement a real time hardware for keyword spotting, we propose a Single-End-Point DTW(SEP-DTW) algorithm, which is simple and less complex for computation. The SEP-DTW algorithm only needs a single end point which enables efficient applications, and it has a small amount of computations because the global search area is divided into successive local search areas. Also, we adopt new local constraints and a new distance measure for a better performance of the SEP-DTW algorithm. Besides, we make a normalization of feature frame vectors so that they have the same variance in each frequency bin, and each frame has the same energy levels. To construct several reference patterns for each keyword, we use a clustering algorithm for all training patterns, and mean vectors in every cluster are taken as reference patterns. In order to detect a key word for input streams of speech, we measure the distances between reference patterns and input pattern, and we make a decision whether the distances are smaller than a pre-defined threshold value. With isolated speech recognition and keyword spotting experiments, we verify that the proposed algorithm has a better performance than other methods.

**Keyword :** 동적 시간 정합, 핵심어 검출, 연속 음성 인식, 단일 끝점 DTW

## I. 서 론

연속 음성 인식 시스템은 실제로 구현하는 데에 있어서 많은 어려움을 가지고 있으며, 그것의 주된 원인 중 하나는 학습 의 단어나 말을 더듬는 현상 등으로 인하여 인식기를 구현하는 것이 매우 힘들 다는 것이다. 이러한 어려움을 해결하며 효율적인 연속 음성 인식을 하기 위해서는 발음된 모든 단어를 빠짐없이 알아내는

\* 학생회원, \*\*정회원, 한국과학기술원 바이오시스템학과 및 뇌과학연구소

(Department of BioSystems and Brain Science Research Center, Korea Advanced Institute of Science and Technology)

\*\*\* 정회원, 목원대학교 정보통신전파공학부

(Division of Information Communication and Radio Engineering, Mokwon University)

※ 본 연구는 과학기술부의 뇌신경정보학 연구사업으로부터 지원 받았습니다.

접수일자: 2002년5월2일, 수정완료일: 2004년4월22일

것보다는 인식에 직접적으로 필요한 단어를 검출하는 것이 보다 나은 방법이다. 실제 이용 가능한 연속 음성 인식 시스템을 구축하게 되면 입력되는 모든 음성을 인식하는 것은 불가능에 가까우므로 몇 가지 지정된 핵심어만을 검출하는 방법은 시스템의 자원을 절약하고 성능을 향상시키는 효과적인 방법이라 할 수 있다.

이러한 이유로 핵심어 검출은 근래에 들어 활발하게 연구되고 있으며 그에 알맞은 인식기가 제안되고 있다. 그 중 대표적인 것으로는 은닉 마코프 모델<sup>1), 2)</sup>(HMM : Hidden Markov Model), 신경 회로망<sup>3)</sup>(ANN : Artificial Neural Network), 동적 시간 정합<sup>4)</sup>(DTW : Dynamic Time Warping)등이 있다. 그러나 실제로 하드웨어를 구성하기 위해서는 구조가 간단하며 연산량이 적어야 하는 등 많은 제약이 따르며, 이를 만족하는 것으로 DTW가 적당하다고 할 수 있다. DTW는 적은 양의 데이터로도 학습이 가능하므로 데이터 수집이 간단하며, 메모리 또한 크게 줄일 수 있다. 게다가 연산의 대부분이 덧셈과 곱셈의 반복으로 구성되어 있으므로 실제적 구현에 알맞다. 다만 DTW를 적용하는 경우 핵심어의 수가 늘어나는 것과 비례하여 연산량이 증가되며<sup>5)</sup>, 끝점의 추출이 선행되어야 하는 문제점이 있다. 즉, 끝점을 추출하지 않고 DTW를 적용할 수 있거나 연산량을 크게 줄일 수 있다면 하드웨어로 구현하는 데에 많은 도움이 된다.

따라서 본 논문의 II장에서는 핵심어 검출을 위하여 빠른 연산이 가능하며 끝점의 추출이 선행될 필요가 없는 단일 끝점 DTW를 제안한다. 제안된 알고리즘은 실생활에 응용되어 직접 하드웨어로 제작 될 수 있도록 최대한 간단한 형태를 띠고 있으며 매우 적은 연산량을 가지면서도 그 성능이 적절한 것을 목표로 한다. 또한, 일반적인 DTW 과정에서 필수적이던 끝점 추출 과정을 개선하여 더욱 효율을 높인다. III장에서는 단일 끝점 DTW를 이용하여 실제적인 핵심어 검출을 하는 과정에 대하여 설명한다. IV장에서는 실제 실험을 위하여 기준 패턴을 만드는 방법에 대하여 설명하고, 제안된 단일 끝점 DTW가 일반적인 DTW에 비하여 어떠한 성능을 보이는지를 비교 실험을 통하여 검증한다. 또한 실제 음성 데이터베이스를 이용하여 핵심어 검출 실험을 수행하고 간단한 HMM과의 비교분석으로 제안된 핵심어 검출 방법이 타당함을 보인다. 마지막으로 V장에서는 본 논문을 요약, 정리하여 결론을 짓는다.

## II. 단일 끝점 DTW

음성 인식에서의 가장 커다란 문제점 중 하나는 같은 발음이라도 말하는 사람에 따라 그 길이가 달라진다는 것이다. 이러한 문제점을 해결하기 위하여 DTW가 널리 이용되고 있으며, 이는 저장되어 있는 기준 패턴과 입력으로 들어오는 테스트 패턴 사이에 서로 대응되는 프레임의 찾아내어 시간왜곡을 보정하는 방법이다. DTW는 좋은 성능을 보이면서도 그 구조가 간단하여 하드웨어 등으로 구현하기에 용이한 장점을 가지고 있다. 하지만 기준 패턴과 테스트 패턴간의 프레임을 대응시키는 모든 경우에 대하여 최적의 결과를 얻고자 하므로 연산량이 많아 실시간 처리를 하는데 어려움이 따른다. 또한 적용을 위하여 먼저 끝점 추출과정이 선행되어야 하는데 연속 음성에서 끝점을 추출하는 것은 매우 어려우며 정확성 또한 낮으므로 가능한 모든 점을 끝점으로 가정하고 인식을 수행하는 방법이 많이 이용되고, 이로 인하여 연산량은 더욱 증가하게 된다<sup>6)</sup>. 따라서 이러한 문제를 해결하기 위해 DTW의 효율적인 적용 방법이 다각도로 연구되어 왔다<sup>7), 8), 9)</sup>.

그 중 연산량을 크게 낮추는 방법으로 두 개의 끝점 중 하나를 검출하지 않는 상태로 DTW를 수행하는 방법이 있다<sup>8)</sup>. 이것은 모든 가능한 시작점에서 끝점의 추출없이 DTW를 시작하고 전역 경로를 연산 중 재설정하는 방법으로, 경로 추적의 시작점에서 위, 아래로 R만큼의 범위 내에 있는 경로만을 대상으로 경로상의 다음 위치점을 선정하고, 새로이 선정된 점에서도 같은 기준으로 다음 위치점을 선정하는 방식을 적용하였다. 그렇지만 이 방법에서는 R의 값이 커질수록 연산량이 원래의 DTW검색과 유사해지며, R의 값 또한 어느 정도의 값 이상이 되어야 하므로 더욱 연산량을 줄일 수 있는 알고리즘이 필요하다.

### 1. 단일 끝점 DTW

본 논문에서는 연산량을 더욱 줄이기 위하여 새로운 DTW 방식을 제안한다. 제안된 구조는 그림 1과 같으며 기본적으로 타입V를 변형하여 구성하였다. 다른 형태를 이용하는 방법도 가능하지만 본 논문에서는 타입V를 기준으로 DTW를 구성하고 실험을 수행하였다.

기존의 DTW는 현재점이 과거의 어떠한 점에서 오는 것이 최적인지를 비교하는 데에 비하여 단일 끝점 DTW에서는 현재점을 기준으로 다음의 5개의 검색점에 대하여 최적의 점을 찾고, 그 점이 바로 다음 연산의 시

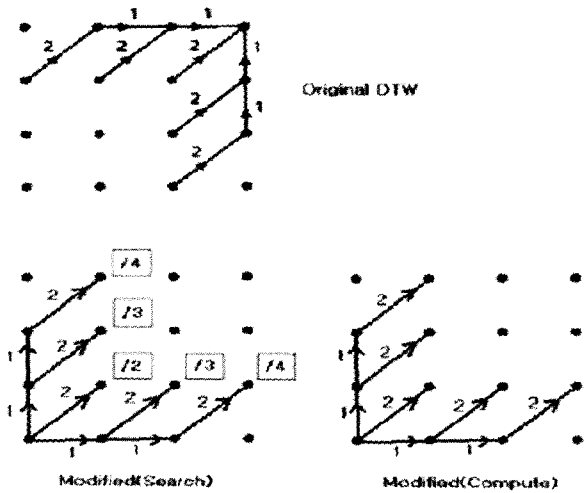


그림 1. 단일 끝점 DTW의 구조 및 가중치  
Fig. 1. Structures and Weights of Single-End-Point DTW

작점이 된다. 이 때 단일 끝점 DTW는 검색과 누적거리 계산에서 경사 가중치를 다르게 주고 있다. 만일 검색의 경우에 원래의 DTW와 마찬가지로의 경사 가중치를 적용한다면 각각의 경로에 대한 가중치의 합이 모두 다르게 되므로 가중치의 합이 가장 작은 경로를 선택하게 될 확률이 높아진다. 이러한 경우를 해결하기 위하여 그림의 검색 부분에서와 같이 국부 오차의 합을 가중치의 합으로 나누어 모든 경로의 가중치 비율을 일정하게 유지시켰다. 이것은 모든 경로가 선택될 확률을 일정하게 유지시켜주며 국부 오차가 가장 적은 방향으로 DTW가 진행되도록 하는 효과를 가져온다. 이러한 형태의 연결이 전체 경로를 구성하므로 전역 경로를 매우 좁게 제한하는 효과를 가져온다. 결과적으로 단 하나의 경로만을 제공하게 되며 이러한 과정을 그림 2에 나타내었다. 한번 검색에서 나타난 최적의 점이 흰색 원으로 표시되어 있다.

일반적인 DTW방식은 테스트 패턴의 양쪽 끝점을 규정하고 검색을 시작하는 데에 비해 단일 끝점 DTW방식은 Myers등의 방식과 같이 한쪽 끝점의 위치를 찾지 않고 검색을 시작하는 것을 알 수 있다. 즉, DTW를 수행하는 중간과정에 따라 그 끝점은 변할 수 있으며, 테스트 패턴의 끝점은 단일 끝점 DTW를 적용하는 과정 내에서 기준 패턴의 끝과 맞닿은 지점으로 정의한다.

이 방법은 일반적 DTW방식이 모든 가능한 경우에 대하여 최적화된 경로를 탐색하는데 비하여, 모든 경로를 추적하지 않은 채 국부적 최적경로의 연속점을 이용하므로 고립단어 인식 등의 실험에서 인식률이 어느 정도 저하될 가능성이 있다. 하지만 고립단어 인식실험이 아닌 핵심어 검출실험 등을 하는데 있어서 필수적인 실

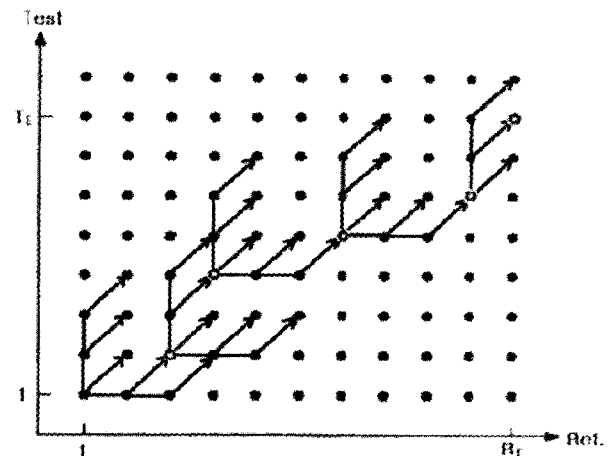


그림 2. 단일 끝점 DTW의 적용 예  
Fig. 2. An example of the Single-End-Point DTW application

시간 연산이 가능할 정도의 연산량 감소의 효과가 있으며, 연속음성 인식실험에서 일반적 DTW가 해결하기 힘든 끝점 추출과정이 자동으로 이루어지고 있다는 장점이 있다.

## 2. 패턴간 거리의 정규화

DTW를 적용하여 나온 패턴간의 거리값은 경로를 따라가며 누적되기 때문에 길게 발음한 단어일 수록 크게 나오는 것이 일반적이다. 그러나 기준 패턴이 여러 가지이며 길이가 고정되어 있지 않은 경우에는 각 기준 패턴들의 효과를 모두 같게 해 주기 위하여 전체 거리를 가중치의 합으로 나누어 정규화하는 방법이 많이 쓰인다<sup>[8]</sup>. 기준 패턴 R(n)과 테스트 패턴 T(m) 사이의 최적화된 경로가  $(i(k), j(k)) \ 1 \leq k \leq K$  로 이루어지며 k는 파라미터(parameter)화된 경로의 표현값, i(k)는 시간 k에서의 기준 패턴의 프레임, j(k)는 시간 k에서의 테스트 패턴의 프레임이라 할 때, 두 패턴간의 전체거리 D는 다음과 같이 표현된다.

$$D(i(K), j(K)) = \frac{\sum_{k=1}^K d(i(k), j(k)) W(k)}{N(W)} \quad (1)$$

여 때,  $d(i(k), j(k))$ 는 기준 패턴의 프레임 벡터 i(k)와 테스트 패턴의 프레임 벡터 j(k)간의 국부 거리를 나타내며  $W(k)$ 는 파라미터화된 경로에 따른 경사 가중치이고,  $N(W)$ 는 전체 경로에서의 가중치의 합이다.

단일 끝점 DTW의 경우에는 끝점이 DTW를 적용하는 과정에서 자동적으로 찾아지기 때문에 하나의 요소

를 더 첨가하여 보다 나은 거리 계산식을 구하였다. 이것은 비슷한 단어나 발음끼리는 길이가 비슷할 가능성이 높으므로 기준 패턴의 길이와 찾아진 테스트 패턴의 길이의 차이를 거리 계산에 첨가 할 수 있다는 것에 기반한다.

$$D'(i(K), j(K)) = D(i(K), j(K)) \times \exp\left[\alpha \times \frac{|i(K) - j(K)|}{i(K) - 1}\right] \quad (2)$$

$\alpha$ 는 길이의 차이에 대한 가중치를 얼마로 할 것인가를 결정하는 요소이며 일종의 파라미터이다. 본 논문에서는 실험적인 결과에 의해  $\alpha=0.2$ 로 두고 이용하였다.

### 3. 연산량 비교

일반적인 DTW방식 및 Myers의 방식과 비교하여 제안한 단일 끝점 DTW방식이 연산량에 있어서 어느 정도의 이득을 가지는지 계산을 통하여 비교해 볼 수 있다. 각각의 경우에 대하여 기준 패턴의 끝점을  $R_E$ , 테스트 패턴의 끝점을  $T_E$ 라고 두고, 한 번 연산이 수행될 때의 연산량을  $N_C$ , 전체 연산량을  $T_C$ 라고 하자. 일반적인 DTW의 경우 전역 경로 제약으로 인하여 검색경로가 전체 영역의  $1/A1$ 로 줄어든다고 하면, 검색이 필요한 영역의 모든 위치에 대하여 검색이 수행될 때, 전체 계산량을 계산할 수 있다. 국부 연속성 제약 타입V를 이용할 경우  $A1=2$ 정도로 두는 것이 보통이므로 연산량은 다음과 같다.

$$T_C = 1/A1 \times N_C \times R_E \times T_E \quad (3)$$

$$\simeq \frac{1}{2} N_C R_E T_E$$

또한, Myers 방식에서는 기준 패턴 매 프레임마다 새로운 검색 구역이 설정되는 것이 일반적이며, 그 때 테스트 패턴  $A2$ 프레임 길이 만큼을 검색구역으로 두고 전체 연산량을 구할 수 있다. 최적의 검색점을 기준으로  $R$ 만큼의 여유를 두고 검색을 수행하므로  $A2=2R+1$ 로 다시 둘 수 있고  $R=4$ 정도의 값을 최소 값으로 Myers는 제안하였다. 이 경우의 연산량은 다음과 같다.

$$T_C = A2 \times N_C \times R_E$$

$$= (2R + 1) \times N_C \times R_E \quad (4)$$

$$\simeq 9N_C R_E$$

이에 비하여 단일 끝점 DTW는 검색을 수행할 때, 한번의 검색으로 인하여 기준 패턴 방향으로 진행되는 프

레이프 수가 매 검색 시기마다 다르다. 이 값을 평균적으로  $A3$ 라고 둘 수 있으며, 검색 시 모든 점이 선택될 확률이 모두 같다고 가정하면  $A3=8/5$ 로 둘 수 있다. 이 때의 연산량은 다음과 같이 나타낼 수 있다.

$$T_C = 1/A3 \times N_C \times R_E \quad (5)$$

$$\simeq \frac{5}{8} N_C R_E$$

즉, 일반적인 DTW방식이  $R_E T_E$ 에 비례하는 연산량을 가지는데 비하여 제안된 방식은  $R_E$ 에 연산량이 비례하므로 매우 효율적이다. 또한 Myers의 방법보다도 약 15배 이상의 연산량 절약의 효과가 있음을 알 수 있다.

## III. 단일 끝점 DTW를 이용한 핵심어 검출

핵심어 검출은 입력되고 있는 음성이 이미 저장되어 있는 기준 패턴과 얼마나 비슷한지를 판단하는 과정이다. 현재의 시점을 모든 핵심어의 끝점이라 가정하고 시작점을 향하여 단일 끝점 DTW를 적용하게 되면 각 핵심어의 기준패턴에 대해서 현재의 테스트 패턴이 어느 정도의 거리값을 가지고 있는지 알 수 있다.

### 1. 끝점 가정

단일 끝점 DTW에서는 시작점이나 끝점 중 하나만을 알고 있으면 된다. 시작점을 알고 있을 경우에는 계산과정에서 끝점을 알아내기 위하여 미래의 입력을 기다리고 있어야 하지만 끝점을 알고 있을 경우에는 이미 입력된 과거의 값을 바탕으로 시작점을 계산할 수 있으므로 보다 효율적이다. 본 논문의 핵심어 검출 방법은 끝점에서 시작하여 시작점을 향하여 계산되는 단일 끝점 DTW를 적용하고, 연속 음성내의 모든 프레임은 핵심어의 끝점이 될 가능성이 있으므로 입력되는 모든 프레임에 대하여 끝점이라는 가정을 적용한다.

### 2. 문턱값의 도입

$k$ 번째 핵심어의  $c$ 번째 클래스에 대한 기준 패턴  $R_k^c$ 을 현재의 음성 프레임  $m$ 에 적용시켜 얻어진 거리  $D_k^c(m)$ 은 기준 패턴과 비슷할수록 그 값이 작고 다를수록 그 값은 커진다. 핵심어 이외의 단어를 거부하기 위하여 어느 정도로 작은 값을 가질 경우 핵심어로 결정

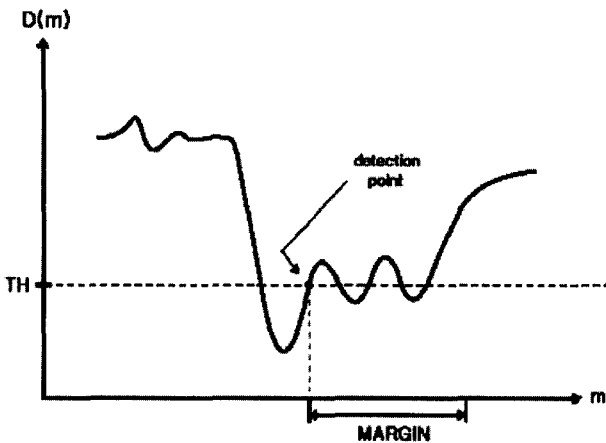


그림 3. 문턱값의 적용과 검출 시점  
Fig. 3. The threshold application and the detection point

할 척도가 필요하다. 이러한 경우 기준이 되는 문턱값을 설정하고 거리값이 그 값보다 작은 경우 핵심어가 검출되었다고 인정하고 그 값보다 클 경우 무시하여 문제를 해결 할 수 있다.

$$\begin{aligned} \text{detection} & : D_k^c(m) \leq TH_k^c \\ \text{don't care} & : \text{otherwise} \end{aligned} \quad (6)$$

k번째 핵심어의 c번째 집단을 대표하는 기준 패턴의 DTW결과가 m번째 프레임에서 지정된 문턱값  $TH_k^c$  보다 같거나 작으면 핵심어가 검출되었다는 것을 의미한다.

### 3. 문턱값의 적용

식(6)을 적용하여 연속 음성 내에서 핵심어를 검출하게 된다면 핵심어 부근에서 매우 많은 수의 검출이 이루어진다. 우리는 핵심어의 어떤 한 지점에서 검출이 이루어지는 것을 바라므로 문턱값을 실제 적용할 경우 변형된 형태로 이용할 필요가 있다. 즉, 검출이 이루어지는 문턱값보다 작은 거리가 발생한 이후에 계속 문턱값보다 작은 값을 유지한다면 검출을 보류하며 기다리다가 문턱값보다 높아지는 순간 검출을 수행하면 이러한 문제를 해결할 수 있다. 그러나 이러한 방법의 경우에도 거리값이 문턱값 부근에서 진동하게 될 때 많은 검출이 발생할 수 있다. 여러번의 검출이 발생할 경우 그 중의 하나만이 가치 있는 경우이고 다른 경우에는 모두 무시하는 것이 옳바르다. 일반적인 단어의 경우 일정한 길이를 가지고 있으므로 한 번의 검출 이후 단어 길이의 최소값보다 가까운 구간에서 검출이 발생한다면 올바른 검출이 아닐 가능성이 크므로 무시해 줄 수 있다. 또한 올바른 기준 패턴에서 작은 값이 나오더라도 일

정 구간 뒤에 올바른 기준 패턴이 더 작은 값을 나타낼 수도 있다. 이러한 여유 프레임수를 Margin이라 두고 그림과 같이 검출을 수행하지 않는 구간을 설정할 수 있다. 대략적으로 Margin=35(프레임)으로 두고 이용하였다.

## IV. 핵심어 검출 실험

### 1. 데이터베이스 및 실험 구성

본 논문의 실험에서는 핵심어 검출 전용 데이터베이스인 RoadRally를 이용하였다. RoadRally corpora는 Stonehenge와 Waterloo라는 두개의 영역으로 구분되어 있다. Stonehenge는 낭독음성 및 대화음성을 고성능 마이크를 이용하여 녹음한 뒤 전화채널을 모방한 필터를 적용한 것이고, Waterloo는 실제 전화상에서 낭독음성을 녹음한 뒤 같은 필터를 적용한 것이다. 실험에서는 Stonehenge의 40명 남성의 낭독 음성(paragraph)을 이용하였다. 본문은 미리 지정된 핵심어가 최소 1회 이상 나타나는 1분 가량의 것이고, 실험을 수행할 전체 데이터가 많지 않기 때문에 40문장을 4개의 세트로 나누어 3개의 세트로 학습하고 나머지 1개의 세트로 테스트하는 과정을 4회 반복하여 많은 데이터를 이용하는 효과를 내었다. 핵심어는 11개를 이용하였으며 모두 434회 발음된다. 제공된 음성은 10kHz로 샘플링(sampling)되어 녹음되어 있으며 실험을 위하여 16kHz로 샘플링 주파수를 변환하여 이용하였다.

### 2. 전처리 과정 및 기준패턴 만들기

DTW를 적용하기 위해서는 테스트 패턴과 정합될 기준 패턴이 미리 만들어져 있어야 한다. 핵심어 하나에 대하여 다수의 기준 패턴이 만들어 질 수 있으며, 그 과정에서 여러 가지 기법이 이용될 수 있다. 그러한 일련의 방법들에 대하여 설명한다.

본 논문에서는 특징 벡터로 ZCPA(Zero Crossing with Peak Amplitudes)를 이용하였다<sup>[10]</sup>. 구해진 ZCPA는 프레임 간격이 10ms이며 주파수에 따른 16차 벡터로 나타난다. 중심 주파수는 대략 200Hz에서 4kHz사이 분포하고 있으며, 하나의 프레임은 가장 낮은 저주파 필터 बैं크의 경우 대략 50ms의 음성 구간, 가장 높은 고주파 필터 बैं크의 경우 약 3ms의 음성 구간에서 특징이 추출된다. 이렇게 추출된 특징 벡터는 다시 cepstrum(cepstrum)형태로 바뀌었다. cepstrum은 특징 벡터의 각 요소간 상관관계를 제거해 주므로 인식률의 향상 및 여

러가지 장점을 가지고 있다. ZCPA 캡스트림은 ZCPA 출력에 DCT(Discrete Cosine Transform)의 역변환을 이용하여 구할 수 있다. ZCPA 캡스트림  $\tilde{c}(m, l)$ 는 다음과 같이 구해진다.

$$\tilde{c}(m, l) = \sum_{i=1}^N y(m, i) \cos \left[ l \left( i - \frac{1}{2} \right) \frac{\pi}{N} \right], \quad 0 \leq l \leq L \quad (7)$$

$m$ 은 현재의 프레임의 의미하고,  $l$ 은 하나의 프레임 내에서 벡터의 한 요소를 나타낸다. 마찬가지로의 형태가 ZCPA 특징벡터에도 적용되므로  $y(m, i)$ 는 ZCPA 특징벡터의  $m$ 번째 프레임의  $i$ 번째 요소를 의미한다. 이용된 ZCPA에서는 16차 계수가 추출되고 캡스트림은 13차 계수를 이용하였으므로  $N=16$ ,  $L=12$ 가 된다.

이렇게 구해진 ZCPA 캡스트림은 각 차원 요소별로 다른 분산값을 갖는다. 특히 에너지를 의미하는 캡스트림의 0차 계수는 분산이 크며 다른 요소들은 이에 비해 상대적으로 분산이 작다. 각 차원마다 분산이 다를 경우 DTW가 고려하는 유클리디안 거리(Euclidean distance) 상에서는 분산에 따른 가중치가 주어지는 것과 같은 효과가 생긴다. 이러한 현상을 해결하기 위하여 추출된 특징 벡터에 분산을 고려한 요소를 첨가하였다.

$$s(m, l) = \left[ \tilde{c}(m, l) - M(l) \right] \times \frac{\beta}{\text{std}(l)} + M(l) \quad (8)$$

$s$ 는 분산값을 적용하여 특징 벡터를 변형한 것을 의미하고  $M(l)$ 은 캡스트림의  $l$ 차 계수의 평균값을 나타낸다.  $\text{std}(l)$ 은  $l$ 차 계수의 표준편차이고  $\beta$ 는 크기를 조정하기 위하여 쓰였다.

이렇게 정규화 된 음성 특징 벡터라도 사람이나 시간에 따라 같은 발음에 대해서도 그 크기가 다르게 된다. 보다 나은 인식 결과를 위해서 미리 기준 패턴과 테스트 패턴의 에너지를 정규화하여 이용하는 것이 유리하다. 그러나 미리 끝점이 추출되지 않은 상태의 연속음성 내에서 핵심어 검출을 수행할 경우 테스트 패턴의 전체 에너지를 미리 알 수 없으므로 이러한 과정이 불가능하다. 따라서 패턴 단위의 에너지 정규화 방법보다는 프레임 단위의 에너지 정규화가 더 효율적이다. 각 프레임의 에너지를 모두 같은 값으로 맞추어 준다면 프레임을 구성하는 요소들은 프레임 내에서의 상호간 에너지 비율에 따라 그 값이 조절된다. 이러한 정규화 과정은 원래의 패턴과는 전체 모양이 다르게 되는 것이 일반적이지만

특징을 결정짓는 요소인 각 성분의 상대적 비율은 그대로 유지되므로 주요한 주파수 성분과 그 구성 상황을 알 수 있는 합리적인 방법이다.

$$n(m, l) = \sqrt{\frac{E_{\text{norm}}}{\sum_{i=0}^L s^2(m, i)}} \times s(m, l) \quad (9)$$

$n(m, l)$ 은  $m$ 번째 프레임의  $l$ 차 캡스트림 계수에 대한 정규화 결과이다.  $E_{\text{norm}}$ 은 하나의 프레임이 가지게 될 에너지를 의미한다.

이러한 전처리 과정을 거친 테스트 패턴은 단일 끝점 DTW에 적용이 가능하지만, 그러기 위하여 각 핵심어에 대한 기준 패턴이 미리 생성되어 있어야 한다. 입력되는 테스트 패턴이 어떠한 기준 패턴과 가장 비슷하며 얼마나 오차가 있는지를 검사하여 결과를 얻기 때문이다. 모든 학습 패턴을 기준 패턴으로 이용할 수도 있지만 실제 같은 발음의 패턴들은 서로 매우 비슷한 형태를 띠고 있으므로 모든 데이터를 저장하고 있는 것은 메모리의 낭비 등 비효율적인 면이 있다. 그러므로 학습 데이터를 이용하여 몇 개의 기준 패턴을 만들어 이용하는 것이 일반적이다. 기준 패턴을 만드는 방법 또한 음성 인식 과정의 중요한 문제 중 하나이며 하나의 핵심어 당 기준 패턴의 수, 각각의 발음들을 몇 개의 부류로 나누는 방법, 크기를 정규화 하는 법, 여러 개의 발음을 모아 하나의 대표값으로 만드는 법 등은 대표적인 기준 패턴 생성의 쟁점 사항이다.

본 논문에서는 기존의 DTW를 이용하여 기준 패턴을 생성한다. 기존의 DTW는 시작점과 끝점이 미리 추출되어 있어야 하고 연산량이 많다는 단점이 있지만 기준 패턴을 만드는 과정은 시간에 크게 구애 받지 않는 과정이며, 패턴들이 모두 미리 잘라져 있으므로 최적의 경로를 찾아주는 기존 DTW 방식을 이용하는 것이 더 효과적이다.

$N(k)$ 개의 패턴을 가지는 핵심어  $k$ 에 대하여  $P$ 개의 기준 패턴을 만드는 경우,

(1) 임의의  $P$ 개 패턴을 중심 패턴이라 가정하고  $N(k)$ 개의 패턴을 중심패턴 각각과 DTW한다.

(2) DTW 거리값이 작은 중심패턴으로 모으면  $P$ 개 그룹을 만들 수 있다.

(3) 각 패턴과 해당되는 중심패턴의 거리를  $d$ 라고 두고  $D = \sum_{i=1}^{N(k)} d_i^2$ 인  $D$ 를 구한다.

표 1. 하나의 학습세트를 이용한 고립 단어 인식률  
Table 1. The recognition rates of an isolated word with one training set

| 학습세트 | DTW 인식률(%) |       |
|------|------------|-------|
|      | 기존         | 단일 끝점 |
| 1    | 96.9       | 95.1  |
| 2    | 96.6       | 92.2  |
| 3    | 96.6       | 94.5  |
| 4    | 96.6       | 93.9  |
| 평균   | 96.7       | 93.9  |

(4) 이 과정을 반복하여 D값이 최소가 되는 P개의 중심 패턴을 찾아낸다.

(5) 찾아진 그룹내에서 모든 패턴들을 중심패턴에 대해 DTW하고 중심패턴을 기준으로 정렬된 프레임들을 평균내어 기준패턴을 만든다.

(6) 얻어진 기준패턴에 대해 에너지 정규화를 하여 완성한다.

### 3. 단일 끝점 DTW 성능검증

단일 끝점 DTW의 성능을 검증하기 위하여 고립단어 인식 실험을 선행하였으며 비교 분석을 위하여 기존의 DTW 성능을 함께 수록하였다. 기존의 DTW는 국부 연속성 제약의 Type V를 이용하여 시작점과 끝점을 맞추어 주었고, 제안된 DTW는 개발한 형태 그대로를 이용하여 한 끝을 맞추고 시작점을 찾아가도록 실험하였다. 단, 찾아진 시작점이 테스트 패턴의 시작점을 지나서 존재할 경우 그 구간에서의 테스트 패턴은 '0'의 값을 가지도록 하였다.

먼저 하나의 세트에 의해서 만들어진 11개 핵심어, 핵심어 당 두개의 기준 패턴을 이용하여 나머지 세개의 세트를 테스트 패턴으로 둔 고립 단어 인식 실험을 하여 보았다. 기존의 DTW를 이용하는 경우에는 입력되는 테스트 패턴의 길이와 비교하고자 하는 기준 패턴의 길이가 3배 이상 차이가 날 수 있으므로 이러한 경우는 두 패턴의 길이를 무한대로 처리하였다. 그 결과를 표 1에 정리하였다.

표 1에서 알 수 있듯이 기존의 DTW가 제안된 DTW보다 인식률이 같거나 좋았다. 이것은 기존의 DTW가 최적경로 탐색 알고리즘을 이용하는데 비하여 제안된 DTW는 준최적 탐색 알고리즘을 이용하고 있기 때문이다. 즉, 기존의 DTW는 확실하게 정해진 두 개의 끝점을

표 2. 세 개의 학습세트를 이용한 고립 단어 인식률  
Table 2. The recognition rates of an isolated word with three training set

| 테스트세트 | DTW 인식률(%) |       |
|-------|------------|-------|
|       | 기존         | 단일 끝점 |
| 1     | 96.3       | 98.2  |
| 2     | 100.0      | 99.1  |
| 3     | 96.3       | 95.4  |
| 4     | 97.2       | 97.2  |
| 평균    | 96.7       | 97.5  |

가지고 최적경로를 탐색하는데 비하여 제안된 DTW는 하나의 끝점을 기준으로 국부적 최적경로의 연속점을 이용하므로, 제안된 DTW방식의 준최적 경로가 실제의 최적경로와 다르게 되거나, 나머지 하나의 끝점을 올바르게 찾지 못하여 정상적인 비교를 하지 못하게 될 가능성이 있다. 그렇지만 연산량면에서는 제안된 DTW가 기존의 DTW에 비해 월등히 우수하기 때문에 약간의 인식률 저하를 감수하더라도 제안된 DTW는 이용의 가치가 있다.

다음으로 핵심어 검출 실험에 이용할 모양대로 세개의 세트를 학습 세트로 이용하고 나머지 하나의 세트를 테스트 세트로 이용한 실험을 수행해 보았다. 이것은 학습패턴의 수가 늘어난 효과를 가져온다. 결과를 표 2에 정리하였다.

기준 패턴의 수가 많아 지면 제안된 DTW의 성능은 기존의 DTW에 거의 가까워지거나 어떤 경우 더 높은 인식률을 보이는 것을 알 수 있다. 이상의 예에서 제안된 DTW는 인식률에서 기존의 DTW에 비해 손해를 거의 보지 않으며 연산량을 월등히 줄였음을 알 수 있다.

### 4. 핵심어 검출 실험

11개 핵심어의 위치를 검출해보는 실험을 수행하였다. 제안된 DTW에서도 한쪽 끝점은 지정되어야 하므로 모든 프레임을 한쪽 끝점이라고 가정하고 매 프레임마다 계산을 반복하였다. 연속 음성 내에서 모든 프레임은 실제로 핵심어의 끝점이 될 가능성이 있기 때문이다. 세트당 핵심어 당 기준패턴의 수는 2개로 두었으며, 실제 실험에서는 3세트가 학습세트가 되므로 핵심어 당 6개의 기준패턴을 가지고 있다.

단일 끝점 DTW를 적용하여 계산된 거리값은 기준 패턴마다 설정되어 있는 문턱값과 비교되어 검출이 결

표 3. 문턱값 설정 실험 : 핵심어 검출시 오인식이 없을 경우의 인식률

Table 3. The threshold setting experiment : the recognition rates with no false alarm in keyword spotting

| 핵심어        | 인식수 | 전체수 | 인식률(%) |
|------------|-----|-----|--------|
| mountain   | 21  | 40  | 52.5   |
| secondary  | 38  | 40  | 95.0   |
| Middleton  | 27  | 37  | 73.0   |
| Boonsboro  | 32  | 39  | 82.1   |
| Conway     | 33  | 40  | 82.5   |
| thicket    | 30  | 39  | 76.9   |
| primary    | 34  | 40  | 85.0   |
| minus      | 27  | 39  | 69.2   |
| interstate | 39  | 40  | 92.5   |
| Waterloo   | 34  | 40  | 85.0   |
| retrace    | 36  | 40  | 90.0   |
| 전체         | 349 | 434 | 80.4   |

정된다. 핵심어 검출의 성능을 나타내는 지표로 ROC(Receiver Operation Characteristic) 곡선을 이용하였고, 그러기 위하여는 오인식 수가 변하며 인식률을 측정하여야 한다. 문턱값이 하나로 고정되어 있을 때 오인식수는 고정되므로 모든 문턱값을 같은 비율로 조정하며 실험을 하였다. 기준이 되는 문턱값은 학습 세트에 적용하였을 때 오인식이 0인 경우의 문턱값을 기준으로 설정하였다. 각각의 학습 세트에 대하여 오인식이 하나도 없도록 문턱값을 조절하고 그때의 인식률을 각 단어에 대하여 측정하였다. 그 결과를 표 3에 나타내었다.

핵심어에 따라서 인식률이 약 50%에서 95%까지 변하는 것을 알 수 있다. 발음이 특이하거나 긴 핵심어의 경우 인식률이 높으며, 발음이 짧거나 약하게 발음되는 핵심어의 경우 인식률이 나쁘다. 그 예로 'mountain' 단어의 경우 실제 발음이 'moun-' 정도로 끝나게 되어 인식률이 좋지 않았고 'minus'와 같이 짧은 단어가 인식률이 나쁘게 된다.

이러한 기준 문턱값을 바탕으로 11개 핵심어 전체에 대한 검출을 시도 하였다. 4개의 세트를 전부 적용하여 결과를 합하였고 핵심어의 문턱값은 같은 비율로 함께 조정하였다. 결과를 ROC 곡선의 형태로 그림 4에 나타내었다.

대략 5FA/H/W에서 50%, 10FA/H/W에서 60% 정도의 인식률을 보이고 있음을 알 수 있다. 핵심어의 평균

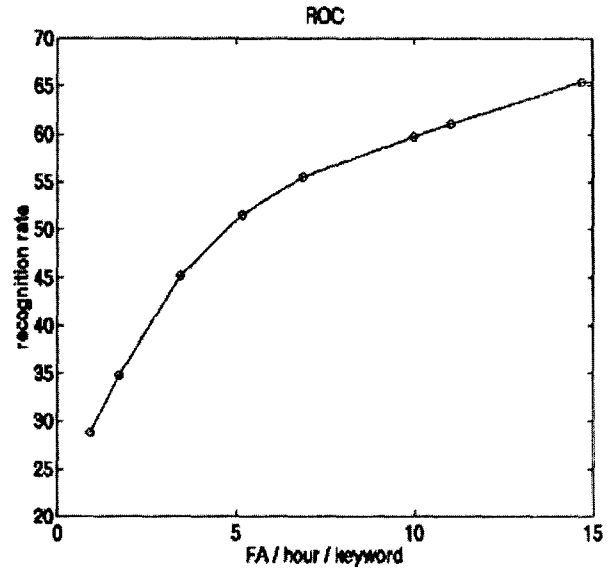


그림 4. 11개 단어에 대한 핵심어 검출 곡선  
Fig. 4. The ROC curve for 11 keywords

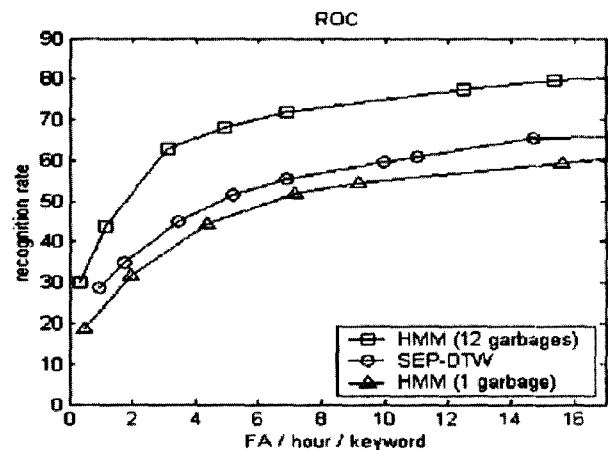


그림 5. 비슷한 조건의 HMM과의 성능 비교  
Fig. 5. Comparison to HMM in the similar condition.

길이는 대략 420ms정도이므로 한시간의 길이에는 단어가 8,000개 이상 존재한다고 생각할 수 있다. 이 중 10개 정도의 오인식은 매우 작은 값을 알 수 있다.

실제로 RoadRally를 이용한 핵심어 검출은 많은 사람들에게 의해서 이루어져 왔다. 그 중 하나의 예로 HMM을 이용한 Wilpon 등의 실험에서는 20개의 핵심어에 대한 Stonehenge의 자연스러운 대화 음성 테스트 결과로 핵심어당 1시간에 8.8개의 오인식률(8.8 FA/H/W)상황에서 74%의 인식률과 0.6 FA/H/W 상황에서 35%의 인식률을 얻었다<sup>[1]</sup>. 그러나 다른 결과들과 본 논문의 결과를 직접적으로 비교하는 것은 다른 실험들이 본 논문의 실험과 데이터 구성 등이 다르므로 정확한 비교라 할 수 없다. 예를 들면, 본 논문에서는 35분의 음성을 학습한데 비하여 Gish의 논문<sup>[4]</sup>에서는 80분의 음성을 이용하였으므로 학습량이 많다. 또한 많은 양의 메모리를 이용하여



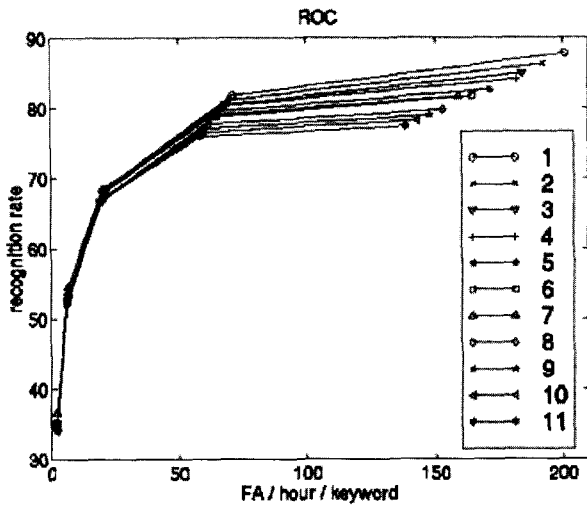


그림 6. 핵심어의 수에 따른 성능 곡선  
Fig. 6. The ROC curves of various number of keywords

핵심어가 아닌 단어에 대한 정보를 학습하며, 인식률이 나쁜 단어에 대하여 2차 테스트를 거치는 등 구조가 복잡하다.

따라서 보다 공정한 비교를 위하여 같은 데이터 세트와 실험조건을 가지는 HMM을 이용하여 실험을 반복하였다. HMM에서는 기본적인 학습단위로 음소(phoneme)를 이용하는 것이 보다 일반적이나 여기에서는 핵심어에 대한 정보를 우선하였으므로 단어(word)를 학습단위로 이용하였다. 이용된 조건으로는 state수의 경우 핵심어에 대하여 14개, silence와 garbage에 대하여 각 30개씩으로 두었고, Gaussian Mixtures의 수는 핵심어의 경우 2개, silence의 경우 1개, garbage의 경우 12개로 두었다. 이 때, garbage 모델을 구성하는 방법 또한 하나의 파라미터가 될 수 있으므로 garbage 모델의 수를 1개와 12개로 두고 실험을 반복하였다. garbage 모델을 1개만 쓴 경우는 제안된 DTW방식과 가장 비슷한 조건을 가진 경우이고, 12개를 쓴 경우는 이 실험에 최적화 되도록 HMM을 학습시킨 경우이다. 모든 모델의 구조는 Left-to-Right 구조이다. 이러한 HMM 모델을 이용하여 구해진 ROC 곡선을 원래의 ROC 곡선과 함께 그림 5에 나타내었다. 같은 오인식수를 보일 때, garbage 모델을 1개만 쓴 경우 제안된 방식에 비하여 약 5%정도 인식률이 나뻐으나, garbage 모델을 12개 쓴 경우 제안된 방식에 비하여 약 15%정도 인식률이 좋았다. 실제로 HMM의 경우 state의 수, Gaussian Mixture의 수, garbage 모델의 수 및 구성 방법에 따라 상당한 인식률의 차이가 있었으며 연산시간이 매우 오래 걸린다는 단점이 있었다. 특히 핵심어 검출을 위하여 단일끝점 DTW를 이용할 경우 오직 핵심어만을 학습시키고 적용할 수 있는 반

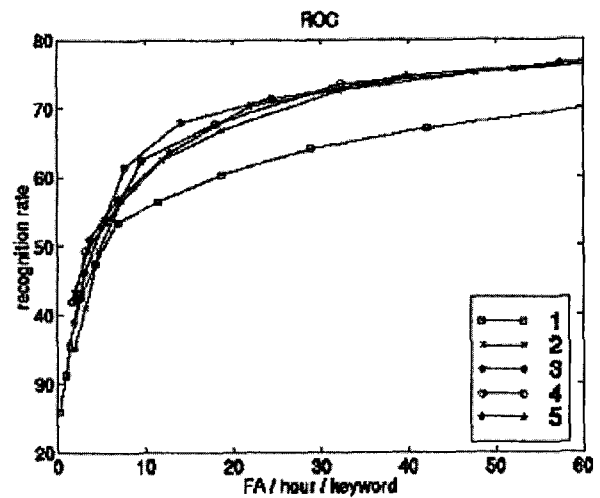


그림 7. 세트 당 기준패턴의 수에 따른 성능 곡선  
Fig. 7. The ROC curves of various number of reference patterns per set

면에 HMM의 경우 많은 파라미터의 조절이 필요하며 garbage 모델을 생성하고 학습시켜야 한다. 또한 HMM을 이용하여 실험을 반복하였을 때, 인식률은 핵심어의 올바른 학습에 의한 것보다는 얼마나 좋은 garbage 모델을 만들었느냐에 크게 좌우되었다. 즉, 단일끝점 DTW를 이용한 핵심어 검출 방법은 HMM을 이용하는 것에 비하여 학습이 쉽고, 구조가 간단하며, 연산시간이 매우 빠른 장점을 가지고 있음을 알 수 있다.

### 5. 조건의 변화

몇 가지 조건을 바꾸어 실험을 반복하여 보았다. 먼저 핵심어의 수를 변화시켰을 경우의 ROC 곡선을 구하여 보았다. 핵심어의 수는 1개에서 11개까지 모든 경우에 대하여 실험을 수행하였으며, n개의 핵심어를 이용하는 경우도 어떠한 핵심어를 이용하느냐에 따라 많은 경우가 존재하므로 여러 경우에 대한 평균값을 이용하였다. 그 결과를 그림 6에 나타내었다.

오인식이 적은 경우에는 핵심어의 수와 인식률이 거의 관계가 없음을 알 수 있다. 오인식 수는 핵심어의 수로 나뉘어 정규화되어 있으므로 오인식 값은 하나의 핵심어에 대하여 오인식이 발생할 확률을 나타내는 것과 같다. 즉, 오인식이 거의 없을 경우에는 각각의 기준패턴에 대하여 틀릴 확률이 비슷하므로 인식률의 차이가 보이지 않는다. 다만 오인식이 많아지면서 여러 기준 패턴들이 서로 영향을 끼치게 되므로 서로의 인식률은 차이를 보이게 된다. 핵심어의 수가 적을 수록 더 높은 인식률을 보이는 것을 알 수 있다.

또한 기준 패턴의 수를 변화시키며 핵심어 검출을 수

행하여 보았다. 하나의 학습 세트당 하나의 키워드에 대하여 두개의 기준 패턴을 생성하였던 것을 하나에서 다섯개까지 변화시켜 보면서 11개 핵심어에 대하여 실험을 반복하였다. 그 결과를 그림 7에 나타내었다.

하나의 학습 세트에 대하여 기준 패턴을 하나만 만들 경우는 인식률이 좋지 않으며 그 외의 경우는 서로 인식률이 거의 비슷함을 알 수 있다. 기준 패턴의 수가 적다면 검출할 수 있는 경우의 수가 적어지므로 인식률이 떨어지는 것은 당연하다. 하지만 기준 패턴의 수가 증가하더라도 어느 정도 이상이 되면 서로 매우 비슷한 기준 패턴들이 생성되므로 여러개의 기준 패턴의 효과를 보지 못하게 된다. 이러한 상황을 그림 7은 잘 나타내어 주고 있고, 효율적인 세트당 하나의 핵심어의 기준 패턴 수는 두개 임을 알 수 있다.

## V. 결 론

본 논문에서는 단일 끝점 DTW를 제안하였다. 이것은 끝점을 추출하지 않고 DTW를 적용할 수 있는 장점이 있으며, 검색경로가 줄어들어 기존의 방법들에 비하여 연산량을 현저하게 줄였다. 또한, 구조적으로 새로운 경사 가중치를 이용하고 있고, 패턴간의 프레임 차이를 이용한 거리 측정법을 도입하여 성능의 향상을 가져왔다. 단일 끝점 DTW는 그 구조도 간단하여 실시간 처리를 위한 하드웨어에 적합한 알고리즘이다.

실제 실험을 위하여 전처리 과정과 기준패턴 생성 과정을 거치게 되는데, 표준편차 및 프레임 에너지에 대한 정규화 방법이 이용되었으며 각 패턴들을 평균 내어 기준패턴을 만드는 방법이 이용되었다.

새로이 제안한 DTW의 성능을 검증하기 위하여 실시한 모의 실험은 RoadRally Corpora를 이용하여 수행되었으며 간단한 고립 단어 인식 실험과 직접적인 핵심어 검출 실험을 통하여 그 우수성을 확인하였다. 핵심어 검출 실험을 통하여 이 방법이 성공적으로 동작하고 있음을 보였고 결과와 계산량을 비교해 보았을 때 이 방법이 효율적임을 알 수 있었다. 다만 특정 핵심어에서 그 성능이 저하되는 것으로 보아 핵심어를 미리 제시된 기준에 따라 선정하여 이용한다면 보다 좋은 성능을 얻을 수 있을 것으로 생각된다.

## 참 고 문 헌

- [1] J. G. Wilpon, L. G. Miller, and P. Modi, "Improvements and Applications for Key Word Recognition Using Hidden Markov Modeling Techniques," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, pp. 309-312, May 1991.
- [2] Herbert Gish and Kenney Ng, "A Segmental Speech Model with Applications to Word Spotting," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, Vol. 2, pp. 447-450, Apr. 1993.
- [3] Torsten Zeppenfeld, Rick Houghton, and Alex Waibel, "Improving the MS-TDNN for Word Spotting," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, Vol. 2, pp. 475-478, Apr. 1993.
- [4] Dominique Vicard, "Transient Part Recognition for Continuous Speech Using Transition Spotting," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, pp. 461-464, Apr. 1988.
- [5] J. C. Junqua and J. P. Haton, Robustness in Automatic Speech Recognition, Kluwer Academic Publishers, p. 325-345, 1996.
- [6] Alan. L. Higgins and Robert E. Wohlford, "Keyword Recognition Using Template Concatenation," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, pp. 1233-1236, Mar. 1985.
- [7] Sei-ichi Nakagawa, Alexander G. Hauptmann, and Masaru Tomita, "On Quick Word Spotting Techniques," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, pp. 2311-2314, Apr. 1986.
- [8] C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Speech Recognition," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, pp. 173-177, Apr. 1980.
- [9] L. Rabiner and B. Juang, Fundamentals of Speech Recognition, Prentice Hall, p. 229-232, 1993.
- [10] D. S. Kim and S. Y. Lee and R. M. Kil, "Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments," IEEE Transaction on Speech and Audio Processing, vol. 7, no. 1, pp. 55-69, Jan. 1999.

— 저 자 소 개 —



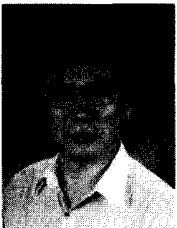
**최 용 선(정회원)**  
 2000년 2월 한국과학기술원 전자  
 전산학과 졸업(공학사)  
 2002년 2월 한국과학기술원 전자  
 전산학과 졸업(공학석사)  
 2002년 3월~현재 한국과학기술원  
 바이오시스템학과 박사과  
 정 재학중

2002년 7월~2003년 6월 미국 University of  
 Connecticut Health Center 방문  
 연구원  
 <주관심분야: 음성인식, 신호처리, 청각모델 등>



**오 상 훈(정회원)**  
 1986년 2월 부산대학교  
 전자공학과 졸업(공학사)  
 1988년 2월 부산대학교  
 전자공학과 졸업(공학석사)  
 1999년 8월 한국과학기술원 전기  
 및 전자공학과 졸업(공학  
 박사)

1988년 1월~1989년 12월 금성반도체(주) 사원  
 1990년 1월~1998년 6월 한국전자통신연구원  
 선임연구원  
 1999년 9월~2000년 3월 한국과학기술원 뇌과학  
 연구센터 위촉연구원  
 2000년 4월~2000년 10월 RIKEN, Brain Science  
 Institute, Research Scientist  
 2000년 11월~2001년 10월 (주)엑스텔테크놀로지  
 연구소장  
 2001년 11월~2002년 2월 한국과학기술원 초빙교수  
 2002년 3월~현재 목원대학교 정보통신전파공학부  
 조교수  
 <주관심분야: ICA, BSS, 지능적 신호처리, 패턴인  
 식 등>



**이 수 영(정회원)**  
 1975년 서울대학교 전자공학과  
 졸업(공학사)  
 1977년 한국과학원 전기전자공학과  
 졸업(공학석사)  
 1984년 Polytechnic Institute of  
 New York 졸업(공학박사)

1977년~1980년 대한엔지니어링(주) (대리)  
 1982년~1985년 미국 General Physics Corp.  
 (Staff / Senior Scientist)  
 1986년~2002년 한국과학기술원 전자전산학과  
 전기및전자공학전공 교수  
 2002년~현재 한국과학기술원 바이오시스템학과 교수  
 1998년~현재 뇌과학연구센터 소장  
 <주관심분야: 음성인식, 신경회로망, 인공시스템,  
 청각모델 등>

