시간 정보와 VQ를 이용한 DDD 지역명 인식에 관한 연구

A Study on the Speech Recognition for DDD Area-Name Using Vector Quantization with Time Information.

이 성 권*, 이 강 성*, 안 태 옥*, 조 형 제**, 변 용 규***, 김 순 협*
(S. K. LEE, K. S. LEE, T. O. ANN, H. J. CHO, Y. G. BYON, S. H. KIM)

요 약

본 논문은 불특정 화자의 DDD 지역명 인식 실험에 관한 것으로 VQ(Vector Quantization) 방식을 이용하여 실험 하였고 인식대상 어휘로는 다이얼링 시스템의 응용을 목적으로 전국 146개의 DDD 지역명을 선정하였다.

특징 파라메타로는 12차 LPC Cepstrum 계수를 사용하여 코우드북을 작성 하였으며, 중심점을 찾는 방법으로는 MINSUM 방법과 MINIMAX 방법을 사용하였고 코우드북 작성에는 Splitting rule 3가지를 사용하였다.

코우드북도 Single Section 코우드북과 시간정보를 포함하는 Multi Section 코우드북으로 나누어 작성 하였고 Section을 Overlapping 하여 가면서 코우드북을 작성하여 실험 하였다.

실험 결과 minsum 방법이 minimax 보다 인식률이 좋은 것으로 나타났으며 화자 독립의 경우 약 90%의 인식율을 얻을 수 있었다.

ABSTRACT

In this paper, we proposed the study on speaker-independent isolated word recognition for DDD area-name using vector quantization and chose total 146 DDD area-name to recognize words for application of dialing system.

We made the codebook using 12th LPC cepstrum coefficients and used the minsum and the minimax method to find the centroid and we applied 3 splitting rule to a codebook generation.

The single section and the multi section with time information were used to generate the codebooks and the over-lapped section codebook was used, too.

From the experiment result, we proved that the minsum method was better than the minimax method and the evaluation of the system yielded an accuracy of about 90 percents in case of speaker-independent.

* 광운대학교 전자계산기 공학과
** 동국대학교 전자계산학과
*** 서울산업대학교 전자계산학과

(본 연구는 1989년 한국과학재단 기초연구지원으로 수행된 연구의 일부분임)

I. 서 론

음성에 관한 자연 과학적 연구에 박차가 가해진 것은 전기통신이 되면서 부터이며 그 이후의 착실한 연구에 의해 Computer를 중심으로 한 새로운 수단이 가하여지고, 정보 통신 시스템의 발전에 따라 기계와 인간 간에 정보 교환의 필요성이 증가하면서 인간에게 더 자연스럽고 친숙한 정보 전달 수단이 요구되는등 음성 연구는 최근 30년간에 현저한 진보를 해왔다.

음성 인식의 역사는 1950년대 모음, 숫자 인식 시스템으로 발전되어 1971년에 Advanced Research Projects Agency (ARPA)에서 음성 이해 시스템의 연구를 지원하면서 본격화 되었다.[1] 현재 미국,일본등 선진외국에서는 지난 십수년간 격리 단어 인식, 연결 단어 인식, 연속 음성 분야에 상당한 발전을 이룩했으며, 인공 지능과의 결합에 의한 음성 이해 시스템 개발을 위해 많은 기초 연구를 국가 혹은 국제적 규모의 과제로 수행하고 있다.

음성 인식에 있어서 최종의 목표는 발성자가 의도하는 것이 무엇인가를 정확하게 추정하는 것이다. 음성 인식을 규정하는 조건으로는 음성의 형태, 화자의 수, 인식 단어의 수, 인식 환경, 인식 방법등이 있는데 이들을 비교하여 정리한 것이 표1에 있다.

표 1. 음성 인식의 분류
Table. 1 Branch of Speech Recognition.

분 류	종 류	비 고
음성의 형태	격리 단어	단어의 앞뒤에묵음이있다고 가정한발음
	연결단어	격리 단어가 연결되어 발음된 음성
	연속 음성	자연스럽게 발음된 음성
화자의 수	화자 종속	training한 화자의 음성으로 test하는 실험
	화자 독립	training하지않은 음성으로 test하는 실험
인식 방법	패턴 매칭	음성의 특징을 비교하여 인식하는 실험
	확률적 방법	음성의 발생 확률을 이용하는 방법

한편 음성 처리에 새로운 접근이 시도되었는데 그중 하나가 벡터 양자화(VQ)이다.[2] 본 논문에서는 VQ에 의해 불특정 화자의 음성 인식을 하는데 있어서 대상어로 146개의 DDD번호에 의한 지역명을 선정하였다. 이유로는 다이얼 시스템에의 응용을 목적으로 하였으며 지역번호를 touch하는 대신, 음성으로 직접 그 지역을 말하여 연결시킬 수 있도록 하기위한 방법의 일환이다. 또한 특징 parameter로는 LPC cepstrum vector를 사용하였으며, VQ의 codebook을 작성하는데 있어서 불특정 화자의 인식을 위해 여러 사람이 발음한 전체 data를 splitting 기법을 사용하여 작성하였다.

II. VQ 이론

1. VQ

VQ란 연속이거나 이산인 벡터들의 sequence를 통신이나 디지탈 채널에 저장 하기에 적당한 지탈 sequence와 mapping하기 위한 코딩방법이다.[1][2] VQ의 가장 큰목적은 데이타 압축으로 데이타의 신뢰성을 잃지 않으며, 최대 한도로 전송 bit rate 를 줄이는데 있다. 데이타 압축에 기여한 Shannon 의 rate distortion이론에 의하면 스칼라 대신에 벡터를 코딩함으로써 더 좋은 성능을 얻을 수 있다는 것이다. 따라서 음성인식에 있어서 데이타 압축이라는 측면에서 표준 패턴을 생성하는데 VQ를 이용한다. 즉 VQ를 음성인식에 이용하면 codebook이 reference word가 되므로 기억 용량을 작게 할 수 있고 또한 codebook의 크기가 작음에 따라 인식하는데 걸리는 시간도 적게 걸린다. 그러므로 VQ는 입력 음성의 특징 벡터를 이미 저장되어 있는 특징 벡터들 중의 하나로 mapping시켜 주는 것을 의미하며, 기본적인 구성도는 그림1과 같다.

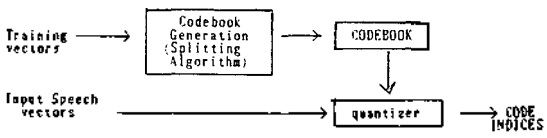

그림 1. 벡터 양자화의 블럭도.
Fig. 1 Block diagram of Vector Quantizer.

시험 벡터들에 의해 codebook이 만들어지며, 입력 벡터는 codebook와 벡터들 중에서 최소의 왜곡률을 갖는 벡터로 양자화 된다. 그림 2에 VQ를 이용한 음성 인식 시스템이 그려져 있다.

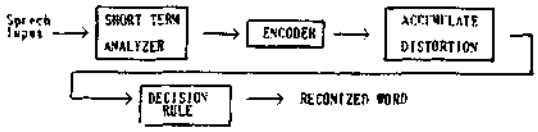

그림 2. VQ를 이용한 음성 인식 시스템.
Fig. 2 Speech recognition System using VQ

2. TRAINING DATA

VQ를 이용한 음성 인식 시스템에서 training data를 구성하는 방법은 두가지가 있다. 첫번째는 single section codebook을 만들기 위한 구성법이고,두번째는 multisection codebook을 만들기 위한 구성법이다.

Single section codebook은 한 단어당 reference word template로서 하나의 codebook을 취하는 방법이다. 그러므로 이 방법을 채택하면 training data는 고전적인 training data와 같이 단어를 M번 발음한 음성 데이타로 구성된다.

Multisection codebook은 한 단어당 reference word template로서 2개 이상의 codebook을 취하는 방법이다. 그러므로 매 단어의 reference pattern을 만들기 위해서는 두개 이상의 training data가 필요하다. Training data를 만들기 위해서 처음에 한 단어를 한번 발음한 것을 원하는 section N 만큼 시간축으로 분할하여 N개의 초기 training data를 만든다. 다음에 이 초기 training data를 매 발음 할때마다 만들어 N*M 개의 초기 training data에서부터 각 section 별로 M개씩 모아 N개의 최종 training data를 만든다.

3. CODEBOOK 작성

VQ를 이용한 음성 인식 시스템에서 codebook은 곧 reference template가 되므로 training data의 특성이 잘 나타나도록 codebook을 만들어야 한다. 본 논문에서는 splitting을 이용한 codebook 작성 방법을 사용하였다.

3-1. Splitting rule

I개의 LPC cepstrum 벡터로 구성된 training 집합 T가 주어졌다고 가정하면 가장 가까운 codebook entry 로부터 집합 T안에 벡터와 평균 거리값이 가장 최소화되는 M^* 개의 벡터로 구성된 codebook을 찾는 것이다. 이 reference 벡터의 집합을 R이라 하면 평균 왜곡값 $DI(M^*)$는 다음과 같이 구해진다.

$$DI(M^*)=\min_{R|\ 1\le m\le M} 1/I \sum_{i=1}^{I}\min_{m} [d(Ti, Rm)] \quad (1)$$

여기서 d(Ti, Rm)은 training 벡터 Ti와 codebook entry Rm 과의 거리값이다. 그리고 splitting rule을 설명하기에 앞서 다음 사항들을 정리하면,

$T_M(m)$: 크기가 M인 VQ에서 m번째 codebook entry에 의해 나타내지는 training vector들의 집합.

$C_M(m)$: $T_M(m)$에 있는 training vector 들의 갯수.

$d_M(m)$: m번째 codebook entry로부터의 $C_M(m)$ vector 들의 average distance(distortion).

$D_M(m)$: $C_M(m)$ vector들의 total distance(distortion).

위에 정의한 사항들의 관계를 살펴보면,

$$I=\sum_{m=1}^{M} C_M(m) \quad (2)$$

$$d_M(m)=1/C_M(m)\sum_{q=1}^{C_M(m)} d(T_M(m)q, R_M) \quad (3)$$

$$D_M(m)=C_M(m)*d_M(m) \quad (4)$$

$$D_M(m)=\min_{R|}\frac{\sum_{m=1}^{M} D_M(m)}{\sum_{m=1}^{M} C_M(m)} \quad \cdots\cdots(5\text{-}1)$$

$$=\min_{R|}\frac{\sum_{m=1}^{M} d_M(m)\ C_M(m)}{\sum_{m=1}^{M} C_M(m)} \quad (5\text{-}2)$$

위의 정의에 입각하여 다음 세가지 rule을 본 논문에서 적용 시켰다.

Rule1: Split the clust, m, with the largest number of vectors, $D_M(m)$.

Rule2: Split the clust, m, with the largest distortion, $C_M(m)$.

Rule3: Split the clust, m, with the average distortion, $d_M(m)$.

3-2. Splitting algorithm

3.1에 제시된 3가지 splitting rule에 따라 codebook을 작성하는 알고리즘은 다음과 같다.

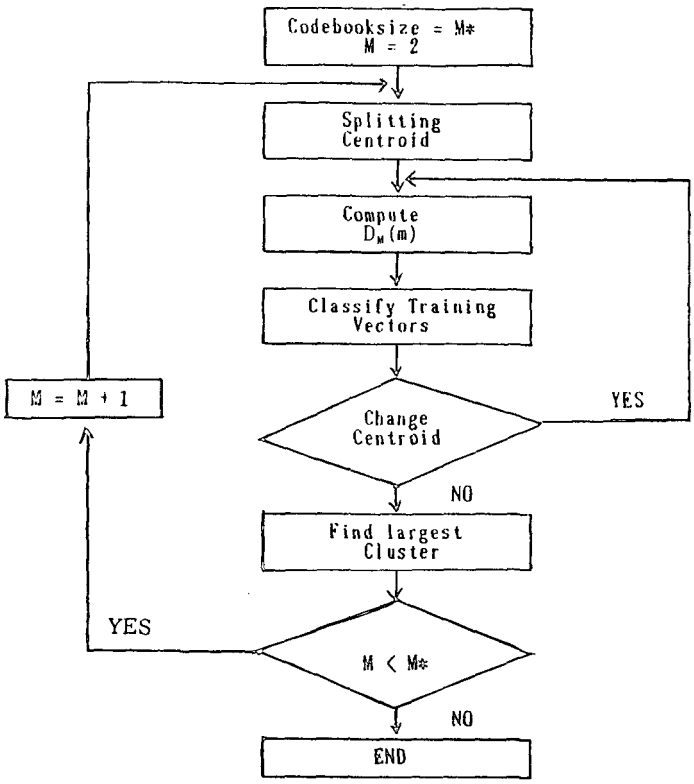

그림 3. Splitting 알고리즘
Fig. 3 Splitting algorithm

(1) Codebook의 vector 수 N을 정하고 M=1로 둔다. 모든 입력 벡터들의 중심을 x(A)라 할때 최초의 codebook은 $A_0(1)=x(A)$가 된다.

(2) M개의 벡터로 이루어진 재생 codebook은 $A_0(M)=\{yi: i=1, M\}$이 되고 각 벡터 yi는 yi+e과 yi−e의 두개의 근접 벡터로 분리되어 2M개의 벡터를 가진 codebook $A(M)=\{yi+e, yi-e, i=1 \cdots, M\}$이 된다. 이때 M은 2M이 된다.

(3) 입력 시험벡터와 codebook의 벡터 간에 왜곡률을 구해 codebook의 각 벡터에 가장 가까운 후보 입력 벡터를 찾는다.

(4) (3)에서 구한 후보 입력 벡터를 찾는 것을 반복하여 시행한다. 그리고 codebook의 벡터수가 N이면 수행을 마치고 아니면 다음으로 넘어간다.

(5) Codebook의 각 벡터는 후보 입력 벡터들의 중심값으로 대치 되어 단계(3)으로 가서 계속 수행한다.

3-3. Splitting 에서 중심점 계산 방법

1) Minimax method

n개의 패턴을 가진 집단 A를 $A=\{x1, x2, x3, \cdots, xn\}$로 표시되면

$$X=X1^* \ni \max_{1\le i\le N} d(X1^*, X1) < \min_{1\le M\le N} \max_{1\le i\le N} d(X_m, X1) \quad (6)$$

여기서 d(a,b)는 a와 b간의 거리를 말한다. 즉, minimax 중심점은 A에 있는 다른 모든 패턴에 대한 최대 거리가 그 집단에 있는 모든 패턴에 대해 최소가 되는 집단에의 패턴이다.

2) Minsum method

집단에 모든 다른 패턴들에 대한 거리의 합이 최소가 되는 패턴을 집단의 중심점으로 정한다. A={x1, x2,… xn} 일때 minsum 집단의 중심점은

$$X=X1 \min_{1\le1\le N} \sum_{i=1}^{N} d(XI^{*}\ XI) \qquad (7)$$

이 방법과 Minimax 방법과의 차이는, Minimax에서의 비교하는 과정이 Minsum 방법에서는 덧셈으로 대치된다. 데이타가 분산된 경우에는 Minimax보다 훨씬 좋은 결과를 나타낸다.

3-4. Distortion measure

VQ 음성 인식 시스템에 있어서 distortion measure란 test data와 reference word template와의 차이를 구하는 일이다. Reference template가 codebook으로 구성되어 졌으므로 test data x와 reference templeate C={xi;i=1,…,N}와의 distortion은 다음과 같이 정의된다.

(1) Test data x와 reference word template의 code word x와의 distortion d(x,xi) {i=1,…,N} (N은 codebook size 임)을 구한다.

(2) Test data x와 reference word template와의 distortion D는 다음의 D=min d(x,xi) {i=1,…,N}으로 정의한다.

3-5. VQ 방법에 의한 거리값 계산

LPC cepstrum vector의 training set을 Ci, i=1,…I, 라 하자. 이 벡터들은 어휘에서의 단어들이 다양한 화자에 의해서 발음될때 일어나는 LPC cepstrum이다. VQ에 숨은 주요 개념은 주어진 M에 대하여 가장 가까이에 있는 codebook entry C에 의해 training set vector C의 각각에 대해 평균 distortion 이 최소가 되도록 LPC cepstrum vector로 최적의 codebook의 집합을 결정하는 것이다. 공식적으로는 두 LPC cepstrum vector Cm과 Ci간의 거리로서 d(Cm, Ci) 라고 정의한다면 그때 VQ의 목표는 집합 C를 찾는 것이다. 이것에 대한 식은,

$$DM=\min \{1/I \sum_{i=1}^{I} \min_{1\le m\le N} [d(Cm, Ci)]\} \qquad (8)$$

위와 같다. DM은 vector quantization의 평균 distortion 이다. 위 식에서 M값 (codebook entry)에 대해 최적의 해를 발견하고 M이 바라는 만큼 반복한다. System에서 사용하는 국부 적인 distance는 다음과 같다.

$$d(Cr, Ct)= \sum_{j=1}^{I} (Ci-Cj)^2 \qquad (9)$$

3-6. Decision rule

Test 음성 data와 모든 reference word template와의 distortion을 구한후 test 음성 data는 가장 작은 distortion을 갖는 reference word template의 단어를 test data로 인식한다.

Ⅲ. MULTISECTION VQ

1. Multi Section 벡터 양자화

단어 음성 인식에서는 발성 속도에 따른 시간변동의 제거를 위해 DP matching (Dynamic Programing)이 많이 이용되고 있다. 이 방법은 시간축의 선형변환에 따른 계산량이 증가한다. 그러므로 시간 정규화가 필요 없는 단어 별로 작성된 VQ codebook에 의해 단어들의 움향적인 특성만을 비교하는 방법을 쓴다.

그러나 codebook에는 시간적 정보가 포함되어 있지 않아서 움향적 특성이 유사한 단어들 사이에 부정확한 인식이 일어난다. 따라서 한 단어를 발성순서에 따라 몇개의 section으로 나누고 section 별로 독립된 codebook 을 작성하므로써 시간적 정보를 포함시키는 Multi Section 벡터 양자화가 Burton등에 의해 제안되었다. Burton의 MSVQ에 따르면 MS codebook 작성에 따르는 모든 음성은 발성시간에 관계없이 일정수의 정해진 길이를 갖는 frame으로 정규화 되어야 한다. 그러므로 발성 시간이 짧은 단어는 정규화 길이에 일치 시키기 위해서 인접 frame을 중첩시켜야 한다. 이것은 분석과 거리 계산에 있어 불필요한 요인이

될 수 있다. 본 논문에서는 frame 정규화없이 전체 frame을 구한 후, 일정한 section으로 나누어 codebook을 작성하였고 또 일정 section으로 나눈 후 각 section을 overlapping 하면서 codebook을 작성하는 방법으로도 실험 하였다.

2. Multi Section codebook 작성

VQ codebook의 sequence로써 시간 정보를 포함하는 방법을 MS codebook이라고 한다. 어떤 단어의 MS codebook은 그 단어를 동일 길이의 section으로 나누고각 section 마다 splitting 방법을 써서 작성한다. 본 논문에서는 음성 data의 전체 frame을 구한후 일정한 section으로 나누어 codebook을 작성하였다.

2.1 4 Multi Section codebook 작성

그림 4에 4MSVQ codebook을 작성하는 과정이 나타나 있다. 한 단어 W를 1회 발성한 음성을 training sequence로 사용한다. 한 frame을 LPC 분석하여 얻은 cepstrum 베터를 v라 하면 1회 발성한 음성은

$$W=\{v1, v2, v3,\cdots, vk\} \quad (10)$$

와 같이 나타낼 수 있다. 인식 대상 어휘가 모두 L개의 단어로 되어 있을때 각 단어 마다 I회 발성 된 음성으로 MS codebook을 구성하기 위해 이들을 J개의 section 으로 나눈다.

$$Wl(i)=\{V1(i)\ V2(i)\cdots VJ(i)\}^{t}\ (l=1,2,\cdots,L) \quad (11)$$

만약 한 section이 N frame으로 구성되어 있다면

$$V1(i)=\{v1(i)\ V2(i)\ V3(i)\cdots vN(i)\}$$
$$VJ(i)=\{v(M-1)N+1(i)\ v(M-1)N+2(i)\cdots\ vMN(i)\} \quad (12)$$

와 같이 각 section을 베터 sequence로 표시 할 수 있다. 그림 4에서 보는 바와 같이 각 section의 frame 수는 다르나, 4 section으로 되어 있으므로 4개의 독립된 VQ codebook의 조합에 의해 MS codebook이 구성된다. Section j에 해당하는 training sequence의 집합을 Sj라 하면

$$Sj=\{vj(1)\ Vj(2)\cdots\ Vj(I)\}\ (\ j=1,2,\cdots,4)\cdots(13)$$

이 된다. 각 section에 대한 codebook Cj는 Sj를 training sequence로 하여서 splitting 방법에 의해 작성된다. 이 과정을 통해 작성된 codebook의 sequence의 집합을 Sj라 하면

$$C=\{C1, C2, C3, C4\} \quad \cdots(14)$$

는 MS codebook을 의미한다. 각 section codebook Cj는 2 개의 code word로 이루어는데 이때 R을 codebook rate라 한다. 본 논문에서는 codebook rate으로 R=2를 사용하였다.

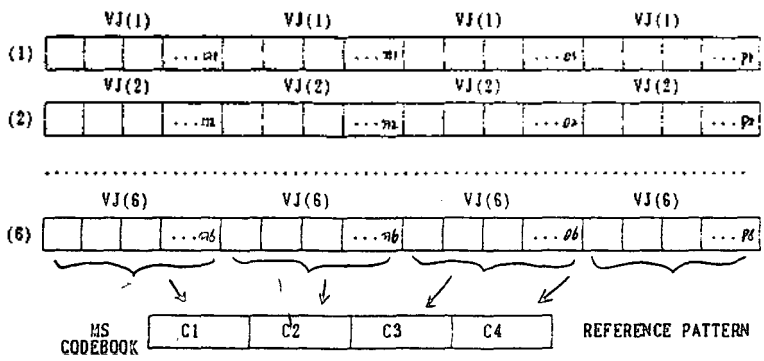

그림 4. 4MS codebook 작성
Fig. 4 4Multi Section codebook generation.

2.2 8 Multi Section codebook 작성

8 MSVQ codebook을 만드는 과정은 4 MSVQ codebook을 만드는 과정과 동일하나, 단지 section의 수를 4에서 8로 늘리는 것만이 다르다. 그래서 8개의 각 section codebook이 모여서 한 단어의 전체 codebook 을 이루게 된다.

2.3. 9 Multi Overlapping Section codebook 작성

4 MSVQ codebook을 만드는 것처럼 9 Multi Overapping Section codebook을 작성하는 과정이 그림 5에 나타나 있다. Codebook을 작성하는 방법은 4 MSVQ와 동일하나 2번째 section부터 overlapping시켜 나가면서 codebook을 만드는 방법만이다르다. Overlapping 시켜 나가는 이유는 section으로 분할할때 분할된 부분의 정보 손실을 우려하여 9개의 section으로 나눈후 overlapping 하였다.

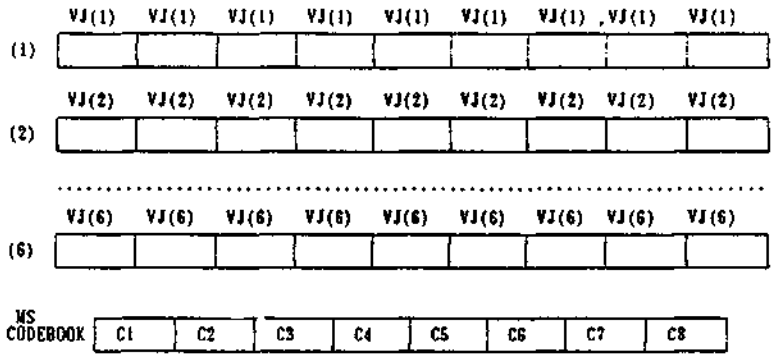

그림 5. 9 Overlapped Multi Section codebook 작성.
Fig. 5 9 Overlapped Multi Section codebook generation.

각 단어 마다 1회 발성 된 음성으로 codebook을 작성하기 위하여 J개의 section으로 나눈다.

$$W(I)=\{Vj(i)\}\ (j=1,\cdots,9)\ (I=1,\cdots 6)\cdots(15)$$

codebook rate=1로 하였으며 8개의 section codebook이 만들어진다.

C={C1, C2, C3, C4, C5, C6, C7, C8}

각 codebook은

$$Cj=\{Vj(I),\ Vj+1(I)\}\ (j=1,\cdots,8)\ (I=1,\cdots 6)\cdots(16)$$

2.4. MSVQ에 의한 단어 인식

일반적인 VQ방법에 의한 거리값 계산과 동일하나 MSVQ에서는 section별 거리값을 합하여 total distortion 값으로 최적의 codebook을 찾는 것이 다르다. 인식하고자 하는 시험 입력 음성 Wx는 먼저 전체 frame을 구한 후 J개의 section으로 나뉜다. 이것을 vector sequence로 나타내면

$$Wx=\{X1,\ X2,\ X3,\cdots Xj\}$$

가 되고 Xj는 J번째 section을 구성하는 frame들로부터 LPC분석을 통해 구한 특징 벡터의 sequence이다. 그래서 Xj는

$$Xj=\{x(j-1)N^{*}+1,\ x(j-1)N^{*}+2,\ \cdots,\ xjN^{*}\}\cdots \quad (17)$$

· N^{*} 는 각 section에 frame 수

로 표시된다. 이들 각 section에 대한 특징 vector들을 표준 패턴의 상대 section codebook의 code word들과

거리 비교를 통해 전체 평균거리인 $D^l av$를 구한다. 어떤 단어 l에 대한 표준 패턴과 전체 평균 거리 $D^l av$는

$$D^l av = 1/N^* \sum_{j=1}^{N^*} dj(Xj, Cj)^2 \cdots (18)$$

$$dj = \sum_{i=(j-1)N^*+1}^{jN^*} \min_{t=0,1\cdots 2^R} d(xi, C^l jt) \quad (t=0,1\cdots 2^R) \cdots (19)$$

식(19)에서 $C^l jt$는 단어 l의 j번째 section codebook의 한 code word를 나타낸다. 이상의 과정을 모든 단어의 표준 패턴에 대하여 반복하여 최종적으로 전체 평균 거리가 최소인 단어

$$1^* = \arg\min_{1} D^l av$$ 이다.

단어 W_1^*을 Wx와 동일한 단어로 인식한다.

Ⅳ. 실험 및 결과

1. 실험조건 및 대상어

VQ에 의한 불특정 화자의 음성 인식을 하는데 있어서 대상 어휘로는 146개의 DDD 번호에 의한 지역명을 선정 하였고 3명의 남성 화자에 의해 각각 3번씩 발성된 것 중에서 각각 2번씩 발음된 것으로 codebook을 작성하였다. 그리고 각각 1번씩 발음된 것으로 인식 실험을 하였다.

본 논문에서 제안한 격리 단어 시스템은 그림 6.과 같다. 마이크를 통하여 입력된 신호는 Sampling 주파수를 8KHz로 하였으며 3.5 KHz 저역 여파기를 통과한 후 12bit A/D 변환을 거쳐 음성신호를 구한 다음, 시작과 끝 구간을 검출한후 LPC 계수를 구한다. 그 다음 이것을 LPC cepstrum 계수로 변환하여 codebook 을 작성한다.

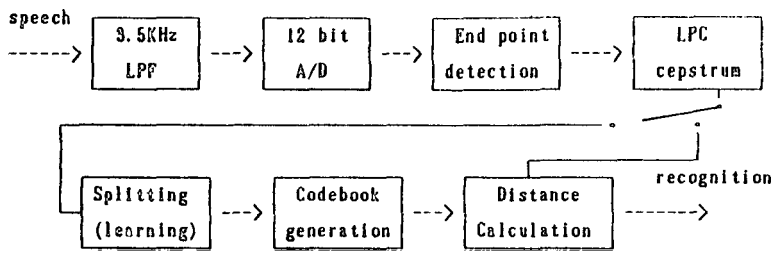

그림 6. 인식 시스템.
Fig. 6 Recognition System.

2.1. Single section codebook (codebook size=8)

본 실험에서는 전체 146개의 DDD 지역명을 '시'와 '도'로 나누어서 각기 '시'와'도'는 인식되었다는 가정하에 특별시와 직할시는 한데 묶어서 9개의 '도'와 합쳐 10개의 군으로 나누어서 인식하였다.

시는 6개 지역, 경기도 21개 지역, 강원도 17개 지역, 충청북도 10개 지역 충청남도 15개 지역, 경상북도 23개 지역, 경상남도 20개 지역, 전라북도 13개 지역, 전라남도 21개 지역, 제주도 등 10개 군으로 나뉘어 인식하였다.

Codebook size 8로 하였을때 중심점을 잡는 방법을 minsum과 minimax 방법으로 하여 splitting rule(1)을 적용한 결과 minsum 방법이 minimax 방법보다 더 나은 결과가 나타났다. 그 결과 표가 아래에 나타나 있다.

표 2. 인식률 (Codebook size=8, rule (1))
Table. 2 Recognition rate (Codebook size=8, rule (1))

Rule(1) : Largest number of vectors		
화 자	minsum	minimax
A	92 %	78 %
B	90 %	75 %
C	92 %	75 %

그러므로 rule 3가지 방법을 minsum 방법에 적용하여 비교 인식하였다.

표 3. 인식률 (Codebook size=8, rule (1)(2)(3))
Table. 3 Recognition rate (Codebook size=8, rule (1)(2)(3))

	MINSUM METHOD		
화 자	Rule (1)	Rule (2)	Rule (3)
	92 %	72 %	87 %
	90 %	64 %	85 %
	92 %	64 %	88 %

2.2. Single section codebook (codebook size=16)

본 실험에서는 VQ codebook size를 16으로 하고 '시'와 '도'는 각각 인식 되었다는 가정하에 인식 실험을 하였다. codebook size가 8일때처럼 splitting rule (1)을 사용하여 minsum과 minimax 방법을 적용, 인식한 결과, minsum이 minimax보다 인식율이 좋으며 codebook size가 8일때 보다 인식율이 증가하였다.

표 4. 인식률 (Codebook size=16, rule (1))
Table. 4 Recognition rate (Codebook size=16, rule (1))

Rule (1) : Largest number of vectors		
화 자	minsum	minimax
A	96 %	94 %
B	92 %	89 %
C	96 %	88 %

2.3. Multisection codebook (section=4: codebook size=16)

본 실험부터는 Multi section codebook으로 작성하여 인식 대상도 시와도의 구분없이 전국 지역으로 확대하여 인식하였다. 이 실험에서는 minsum과 minimax를 적용하여 splitting rule (1)을 써서 실험하였고 그 결과가 아래에 나타나 있다.

표 5. 인식률 (Codebook size=16, rule (1))
Table. 5 Recognition rate (Codebook size=16, rule (1))

Rule (1) : Largest number of vectors		
화 자	minsum	minimax
A	92 %	84 %
B	86 %	81 %
C	75 %	80 %

2.4. Multisection codebook (section=8: codebook size=16)

본 실험에서는 Multisection codebook을 작성하는데 있어 section의 수를 배로 늘려 실험하였다. Codebook size의 수는 16으로 고정시키고 minsum과 minimax 방법으로 splitting rule (1)로 하여 실험한 결과 오히려 인식율이 떨어지는 결과를 낳았다. 그결과가 아래에 있다.

표 6. 인식률 (Codebook size=16, rule (1))
Table. 6 Recognition rate (Codebook size=16, rule (1))

Rule (1) : Largest number of vectors		
화 자	minsum	minimax
A	82 %	79 %
B	76 %	75 %
C	72 %	70 %

2.5. Multisection overlapping codebook (section=8: codebook size=16)

본 실험에서는 Multisection codebook을 작성할때 section으로 분할되는 부분의 정보 손실을 우려하여 codebook 작성할때 overlapping 하였다. 인식 대상도 전국 지역명 146개를 군으로 나눔없이 하였으며 minsum 방법과 minimax 방법을 적용하여 splitting rule(1)을 적용하여 인식 실험하였다. 그리하여 4 section codebook으로 실험하였을 때 보다, 8 section codebook으로 실험하였을때 보다 좋은 인식율을 얻을 수 있었다. 그러나 minimax 방법은 4MSVQ에 비하여 상대적으로 좋지 않은 결과를 보이고 있다. 그 결과가 아래에 나타나 있다.

표 7. 인식률 (Codebook size=16, rule (1))
Table. 7 Recognition rate (Codebook size=16, rule (1))

Rule (1) : Largest number of vectors		
화 자	minsum	minimax
A	95 %	79 %
B	87 %	76 %
C	86 %	72 %

V. 결 론

본 논문에서는 VQ 방식을 이용하여 single section codebook과 multi section codebook을 만들어 실험하였다. 실험한 결과는 표에 나타난 바와 같이 codeword 즉, codebook size가 증가함에 따라 인식율이 나아졌으며, single section codebook 보다는 multisection codebook 일 경우 codebook만드는데 시간도 적게 걸리며 좋은 인식율을 보이고 있다. Single section codebook 일 경우 음향학적 특성이 유사한 단어들 사이에 오인식되는 경우가 빈번이 발생하여 인식율을 낮추는 결과가 나왔으나,codebook size를 크게 함에 따라서 오인식이 줄어들었다.

한편 single section codebook에 시간정보가 포함되지 않아 도시명인 "양평"이 "평택"등으로 오인식이 빈번이 발생한다고 생각되어져 multisection VQ로 실험하였다. 한 단어의 전체 프레임의 길이가 크지 않다고 생각되어 4 section으로 나누어 실험한 결과 전국 지역을 대상으로 인식 실험을 할 수 있었다.

또한 section을 배로 나누어 codebook을 만들어 실험한 결과 오히려 인식율이 떨어졌다. 이것은 너무많은 section으로 나누어 생긴 결과로 추정되며 나누어진 section에 대한 정보의 손실로 인해 인식율이 떨어지는 것으로 사료된다.

그래서 section으로 나뉜 부분에 대한 정보의 손실을 막고자 section의 수를 늘리고, section을 overlapping하여 codebook을 작성하여 실험한 결과 overlapping하지 않은 MSVQ보다 5%의 나은 결과를 얻을 수 있었다. 그리고 중심점은 minsum 방법이, splitting rule은 베터의수가 좋은 결과를 보여 주었다. 화자독립의 경우 약90%의 인식율을 얻어 제안된 알고리즘의 유효성을 입증하였다. 앞으로 vector간의 거리값 계산과 overlapping에 대한 점을 개선한다면 더 좋은 결과를 얻을수 있을것으로 사료된다.

참 고 문 헌

1. D.R, Reddy "Speech Recognition by machine: A Review," Proc. IEEE, Vol. 64, No.4, pp.501~503, APR.1976.
2. L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs, N.J.: Pretice-Hall 1987
3. A.E. Rosenberg, "Automatic Speaker Verification: A Review, "Proc. IEEE, Vol. 64, pp475~487 1976
4. Manfred R. Schroeder "Linear Predictive Coding of Speech: Review and Currenct Directions," IEEE Comm. Magazine, Vol.23, No.8, August 1985.
5. P. Wallich "Putting speech recognition to work," IEEE Spectrum, Vol. 24, pp. 55~57, Apr. 1987.
6. M. Robert Ito and R.W. Donaldson, "Zero-Crossing Measurements for Recognition of speech sound," IEEE Trans. on Audio and Electro-acou-stics, Vol. AV-19, No. 3, pp.235~242 Sep. 1971.
7. L.R. Rabiner and S.E Levinson, "Isolated and connected Word Recog nition Theory and Selected Applications" IEEE Trans. on Communication, vol. COM-29, No.5, pp621~639, May 1981.
8. J.D. Markel and A.H. Gray, Linear Prediction of Speech, Spring-Verlag Berline Heidelberg 1976.
9. Seiich NAKAGAWA, Mitsunori SAAKMOTO, " Evaluation of FFT Cepstrum and LPC Cepstrum for speech and Speaker Recognition," 일본 전자 통신학회 논문집, Vol. J66~A No.12, pp1199~1206 1983.
10. Shikano, K, Kohda, M. "On the LPC distance Measures for Vowel Recognition in coutinuous utterances, " IEEE Jpn. D. J63~D157, May. 1980
11. R.M. Gray, "Vector quantization", IEEE ASSP Magaz ine, Vol. 1, pp4~29 Apr. 1984.
12. Y. Linde, A. Buzo, and R.M. Gray "An algorithm of Vector Quantizer Design", IEEE Trans. Comman, Vol. COM-28 pp. 84~95, Jan 1980.

▲이 성 권

1982년 2월 배재고등학교졸
1988년 2월 광운대 전자계산기공학과 졸업
1990년 2월 광운대 대학원 전자계산기공학과졸업 예정

▲이 강 성 : 8권3호 참조

▲안 태 욱 : 8권4호 참조

▲변 용 규 : 8권3호 참조

▲조 형 제

1949년 10월 24일생
1973년 부산대학교 전자공학과 졸업
1975년 한국과학원 전기및전자공학과 졸업(석사) 금성통신(주) 연구소 근무
1986년 한국과학기술원 전기및 전자공학과 박사 학위 취득
• 현재 : 동국대학교 전자계산학과 교수
※관심분야 : 음성인식, 패턴인식, 컴퓨터통신, 컴퓨터그래픽

▲김 순 협 : 8권3호 참조