산업기술연구(강원대학교 산업기술연구소 논문집), 제21권 B호, 2001.
Journal of Industrial Technology, Kangwon Natl. Univ., Korea, No. 21 B, 2001.

DSP Processor(TMS320C32)를 이용한 화자인증 보안시스템의 구현

Implementation of Speaker Verification Security System Using DSP Processor(TMS320C32)

함 영 준* 권 혁 재** 최 수 영*** 정 익 주****
Haam, Young-Jun Kwon, Hyuk-Jae Choi, Soo-Young Jeong, Ik-Joo

Abstract

The speech includes various kinds of information : language information, speaker's information, affectivity, hygienic condition, utterance environment etc. when a person communicates with others. All technologies to utilize in real life processing this speech are called the speech technology. The speech technology contains speaker's information that among them and it includes a speech which is known as a speaker recognition.

DTW(Dynamic Time Warping) is the speaker recognition technology that seeks the pattern of standard speech signal and the similarity degree in an inputted speech signal using dynamic programming.

In this study, using TMS320C32 DSP processor, we are to embody this DTW and to construct a security system.

키워드 : *디지털신호처리, 음성인식, 화자인증, 보안시스템*
Keywords : *DSP, Speech Recognition, Speaker Identification, Security system*

1. 서론

현대 사회는 디지털 사회로 불릴 만큼 모든 것이 디지털화 되어가고 있으며 따라서 디지털 신호의 처리(Digital Signal Processing - 이하 DSP)에도 큰 관심이 모여지고 있다. DSP를 응용 할 수 있는 분야는 매우 광범위한데 그 가운데 음성 인식이라고 하는 분야는 최근 새롭게 대두되어 매우 급속한 발전을 이루고 있는 부분이다. 매우 복잡한 연산 처리와 소형화의 큰 과제를 모두 해결하기 위해 DSP Processor의 사용이 필수적이며 본 연구에서는 DSP Processor로 Texas Instrument사의 TMS320C32를 이용하였다.

음성인식이라는 분야에는 화자인증, 핵심어인증, TTS, STT 등등의 많은 세부 분류가 있는데 그 가운데 본 연구에서는 화자인증 기술을 이용하여 보안 기능의 시스템(자동문)을 구현하였다. 사람의 목소리가 각기 다르다는 점을 이용한 방식으로써 사용자의 ID를 음성이라는 편리한 정보전달 수단을 기계에 이식(移植;transplantation)하는 방법과 이를 인식(認識;recognition)하는 방법을 이용한 연구이다.

Speaker Verification Technique과 쉽게 연관지어 생각해 볼 수 있는 예로 전화의 수화기에서 들려오는 친구의 목소리를 인지하는 과정을 보면 먼

* 강원대학교 전자공학과 학사과정
** 강원대학교 전자공학과 학사과정
*** 강원대학교 전자공학과 학사과정
****강원대학교 전자공학과 교수, 공학박사

저 사람의 청각 및 인지를 담당하는 기관에서는 우선 이 목소리를 자신이 알고 있는 모든 목소리와 비교해서 자신의 기억 속에 있는 누군가와 매우 비슷하다는 사실을 일단 파악하게 되는데 이를 화자 식별(Speaker Identification)이라고 한다. 그리고 다음 단계로는 들리는 음성과 기억 속의 음성을 계속해서 비교하여 어느 부분에서 그 유사성이 매우 커지게 될 시점에는 결정적인 판단을 내리게 되는데 이를 화자 검증(Speaker Verification)이라고 한다. 따라서 Speaker Recognition 기술은 다시 Speaker Identification 과 Speaker Verification으로 나눌 수 있다. Speaker Identification은 등록된 화자들 중에서 가장 유사한 화자를 골라내는 것을 말하며 Speaker Verification 기술은 승인(Acceptance) 및 거절(Rejection)을 행하는 과정이며 기준 패턴(저장되어진 화자별 codebook-reference data)과 입력 패턴을 서로 비교하여 승인 또는 거절하게 된다.

본 연구에서는 화자식별 기술과 화자 검증 기술을 조합, 응용하여 보안용도로 사용이 가능한 화자 인증 보안 자동문을 구현하였는데, 2장에서는 화자 인증에 관한 알고리즘들의 간단한 이론적인 내용을 다루었으며, 3장에서는 알고리즘들의 세부적인 내용들과 그것을 구현하기 위한 Hardware system의 구성과 구현방법을 다루었으며, 4장에서는 구현되어진 시스템을 통한 본 연구의 결과들을, 그리고 마지막 5장에서는 연구의 결과를 내렸다.

2. 이론적 배경

Speaker Recognition System은 다양한 방법으로 구현될 수 있다. Speaker Recognition의 실제적인 알고리즘을 어떤 형태로 구현할 것인가의 관점에서 보면 문맥 종속(Text Dependent)과 문맥 독립(Text Independent)으로 나눌 수 있는데 Text Dependent란 정해진 말 즉, 미리 정해놓은 단어나 문장 등을 말하는 것을 뜻하며, Text Dependent System의 경우에는 그 특성 때문에 DTW (Dynamic Time Warping) 알고리즘을 주로 사용하게 된다. 반면 Text Independent란 미리 정한 말이 없이 무작위의 선택된 말을 하는 것이다. Text Independent System의 경우에는 HMM (Hidden Markov Model) 알고리즘을 많이 사용하고 있다.

이상의 대표적인 두 알고리즘의 비교는 다음의 표 1에서 보여진다.

표 1. DTW와 HMM의 간단한 비교

	Text Dependent	Text Independent
Algorithm	DTW	HMM
Data의 양	짧은발성	대규모음성
Advantage	성능이 좋다. 구현이 쉽다.	아무 말이나 가능하다
Disadvantage	흉내내기 가능	Training 과정이 복잡하다.

따라서 본 연구에서 구현한 System의 알고리즘은 Text Dependent Speaker Recognition이므로 DTW 알고리즘을 사용하였다. 일반적으로 핵심어 인식[1] 기능의 System은 화자에 관계없이 누구의 음성이라도 핵심어를 이끌어내지만 본 System에서는 화자에도 크게 종속되어져 있어야 하므로 임계치를 매우 낮게 정해서 적용을 해야만 한다. 구현되어진 알고리즘에 관한 내용을 정리해서 블록도로 살펴보면 다음의 그림 1과 같다.

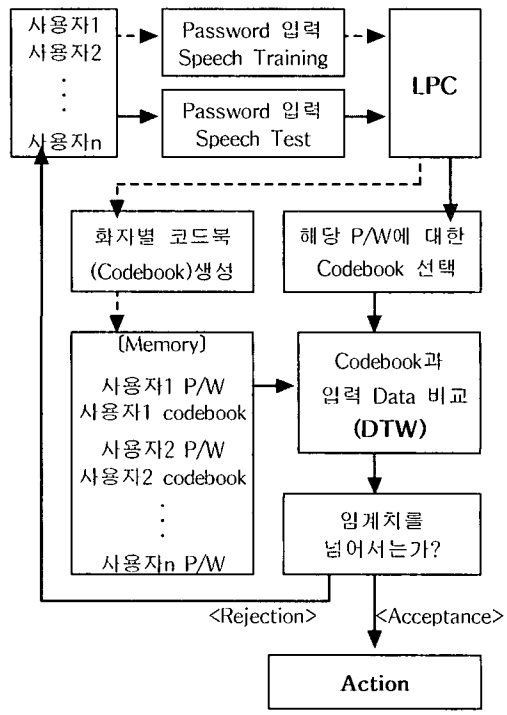

그림 1. 구현되어진 Speaker Recognition System의 기본적인 블록도
(---▶ : Training ——▶ : Test)

1) 여러 음절의 문장 속에서 핵심이 되는 하나 또는 몇 개의 단어를 검출해 내는 기술

3. 연구 내용

3.1 System Algorithms 세부사항

3.1.1 Endpoint Detection

입력된 음성신호의 실제 음성구간 Detection은 화자인증 System의 Performance에 큰 영향을 미치므로 여러 과정 중에서 매우 중요한 부분이다. 입력신호가 유성음(음성신호)일 경우에는 그 에너지의 크기가 크고 주파수가 낮으며, 무성음(잡음)일 경우에는 그 크기가 작고 주파수가 매우 높다. 본 연구에서는 신호 에너지의 임계치를 적당히 정해 놓고 들어오는 입력 신호의 에너지를 계속해서 관찰을 하여 임계치보다 낮으면 신호로 취급하지 않고 Threshold value보다 에너지가 크면 신호로 받아들이고 그 신호의 에너지가 다시 임계치 이하로 내려갈 때까지를 음성 신호로 판단을 하게 되고, 그 구간을 검출하여 의미를 가진 Signal로 사용하게 된다.

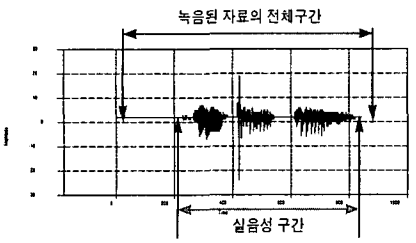

그림 2. 실제 음성구간 검출

하지만 단지 신호의 에너지만을 측정 방법으로 사용한 것이 아니라 level zero crossing을 아래와 같이 적용하여 함께 이용한다.

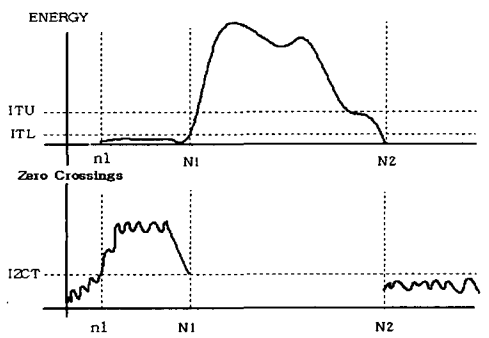

그림 3. Endpoint Detection

interval(실제 음성신호의 지속간격)은 항상 ITU (threshold)를 초과한다. 따라서, 이 값 이상의 data를 갖는 것을 interval로 여기고 beginning and ending point는 이 간격 바깥에 위치한다고 가정하고, zero-crossing rate를 이용하여 beginning point 는 N1앞에서 구하고 ending point는 N2 뒤에서 구한다. zero-crossing rate이 정해진 임계치인 IZCT(rate이 3이상)보다 크기 시작하는 곳이 beginning point가 되고, 같은 방법으로 N2이후에서 IZCT보다 값이 작아지기 시작하는 곳이 ending point가 된다.

3.1.2 LPC(Linear Predictive Coding)

LPC는 음성의 발성기관(Vocal Track)을 하나의 필터로 가정하고, 그 필터의 계수를 음성의 특징 벡터로 사용하는 것이다.

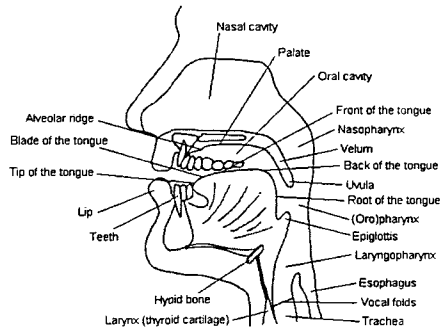

그림 4. Vocal Track (성도)

Vocal Track은 A(z)이며 그 필터를 구동하는 입력 소스는 유성음의 경우 임펄스 열로 무성음의 경우 백색 잡음으로 modeling한다.

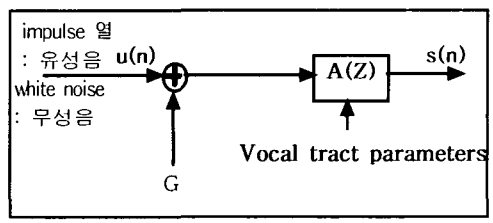

그림 5. 음성인식을 위한 LPC Model

다음에 보여지는 그림 6은 LPC의 블록도이다.

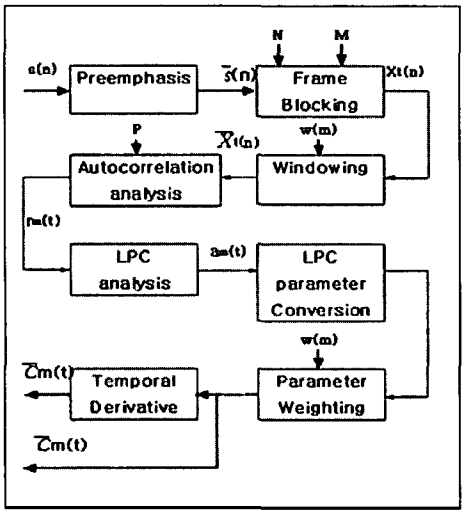

그림 6. LPC process의 블록도

앞에서 보여진 블록도의 각 블록에 대해 간단히 살펴보면 다음과 같다.

① Preemphasis

계산의 안정성과 선형 예측을 이용한 분석시 Vocal Track의 영향만을 고려하기 위해 사용하며 고주파 성분을 강조하는 효과로서 fixed first-order system을 널리 사용한다.

② Frame Blocking

Preemphasis된 음성신호인 s̃(n)를 N개의 샘플로 블럭을 나눈다. 인접한 블록은 M샘플의 차이가 난다. 즉 (N-M)개의 샘플이 overlap 된다.

③ Windowing

각각의 프레임의 처음과 끝에서의 불연속을 최소화하기 위해 해밍윈도우[2]를 사용한다. 0≤n<N의 윈도우 w(n)을 사용하면 윈도우의 결과로 신호는 다음과 같이 나타나고

$$\ddot{x}_l(n) = x_l(n)w(n), \quad 0 \le n < N \qquad (1\text{-}1)$$

해밍 윈도우는 다음과 같다.

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n < N-1 \qquad (1\text{-}2)$$

각 프레임마다 같은 윈도우를 적용하므로 미리 계산해 테이블로 저장해 둔다.

④ Autocorrelation Analysis

Windowing된 각각의 프레임을 autocorrelation한다.

$$r(m) = \sum_{n=0}^{N-1-m} \ddot{x}_l(n)\,\ddot{x}_l(n+m), \quad m=0,1,\ldots,p, \qquad (1\text{-}3)$$

이 식에서 p는 LPC Analysis의 차수이다. 일반적으로 p는 8에서 16까지의 값이고 p=8을 가장 널리 사용한다. 본 연구의 실제 구현에서는 LPC의 차수를 14차로 하였다.

⑤ LPC Analysis

각각의 autocorrelation된 프레임을 LPC parameter set으로 변환하는 단계에서는 LPC 필터 계수를 구한다. 이런 계수들은 Vocal Track의 모양에 대한 정보를 주파수 영역을 근간으로 표현한 것이라고 생각하면 된다. 발음이 서로 다른 것은 발음할 때 Vocal Track의 모양이 다르기 때문이다. 따라서 Vocal Track의 모양에 대한 정보가 발음에 대한 정보라고 볼 수 있다. 또한 몇 차로 분석하느냐에 따라 분석의 성능이 달라지게 된다.

⑥ LPC Parameter Conversion to Cepstral Coefficients

LPC 필터 계수가 구해지면 이를 Cepstral 계수로 변환한다. Cepstral Coefficients란 로그 크기 스펙트럼(log magnitude spectrum)의 FT(fourier Transform)의 계수를 말한다. 음성인식에서 LPC 계수나 log area ratio 보다 더 신뢰할 수 있는 것으로 보인다.

⑦ Parameter Weighting

Cepstral 계수의 민감도를 완화시키기 위해 tepered window 기법을 사용한다. 일반적인 방법으로 log magnitude spectrum 과 the differented(in frequency) log magnitude spectrum을 사용한다.

$$\log |S(e^{jw})| = \sum_{m=-\infty}^{\infty} c_m e^{-jwm} \qquad (1\text{-}4)$$

: log magnitude spectrum

$$\frac{\partial}{\partial w}[\log |S(e^{jw})|] = \sum_{m=-\infty}^{\infty} (-jw)\, c_m e^{-jwm} \qquad (1\text{-}5)$$

: different log magnitude

식 (1-5)는 식 (1-4)에서 고정된 스펙트럼의 기울기를 가지고 있다. 따라서 미분을 하면 기울기는

2) FIR Filter 설계기법 중 windowing기법의 하나

상수가 된다. 또한 식(1-4)와 같을 때 나타나는 peak값 (e.g. formants)은 미분을 해도 잘 보전되어 있다. 즉, 미분을 함으로써 (-jw)이 곱하게 되어 weighting으로써 나타낼 수 있다.

$$\frac{\partial}{\partial w}[\log|S(e^{jw})|] = \sum_{m=-\infty}^{\infty} \hat{c}_m e^{-jwm} \quad (1\text{-}6)$$

where $\hat{c}_m = c_m(-jm)$

$\hat{c}_m = w_m c_m$, $1 \le m \le Q$

⑧ Temporal Cepstral Derivative

더 향상되고 확장된 스펙트럼의 표현을 위해 Temporal Cepstral Derivative을 사용한다. log magnitude spectrum을 시간에 대해 미분하면 Fourier Series form형태로 나타난다. 즉, 시간 t일 때 m번째의 cepstral 계수 Cm(t)는 discrete time으로 표시된다.

$$\frac{\partial}{\partial w}[\log|S(e^{jw},t)|] = \sum_{m=-\infty}^{\infty} \frac{\partial c_m(t)}{\partial t} e^{-jwm}$$

$$\frac{\partial c_m(t)}{\partial t} = \Delta c_m(t) \approx \mu \sum_{k=-K}^{K} k\, c_m(t+k)$$

μ는 적당한 normalization 상수이며, 이상의 계산에 의하여 각각의 프레임에서 weighted Cepstral 계수는 Q의 벡터이다. 따라서 Q의 벡터를 첨부해야 한다.

$$o_t' = (\hat{c}_1(t), \ldots, \hat{c}_Q(t), \Delta c_1(t), \ldots, \Delta c_Q(t))$$

위의 식에서 o_t는 2Q의 요소를 가진 벡터이다. 그리고 '는 transpose를 의미한다. 결과적으로 한 프레임으로부터 Q개의 weighted Cepstral 계수($\hat{c}_m$, $1 \le m \le Q$)를 추출해내게 된다. 즉 한 프레임의 음성 신호로부터 Q개의 성분을 갖는 특징 벡터를 추출해 이것으로 그 프레임의 음성 신호를 대표하는 것이다.

3.1.3 DTW(Pattern-Comparison)

어떤 두 프레임의 특징 벡터가 있다면 이 두 특징 벡터의 유사성 또는 상이함을 어떻게 알 수 있을까 하는 것을 생각해야 한다. 입력된 음성들은 여러 프레임으로 나누어져 분석되고 결국 연속된 특징 벡터로 표현되므로 이들 음성간의 유사성을 비교하기 위해서는 별도의 척도가 필요하게 된다. 이를 거리(distance)라고 한다. 좋은 거리 측정 방식은 다음과 같은 성질은 가져야 한다.

- 음향학적으로 유사한 두 프레임의 거리는 가깝게, 음향학적으로 상이한 두 프레임의 거리는 멀게 판정해야 한다.
- 수학적으로 다루기에 무리가 없어야 하며 계산량이 적당해야 한다.

음성은 지속 시간이 달라지는 특성을 가지고 있는데 이것은 유사성을 측정하려는 두 음성 신호간에 나타나는 시간축의 차이이다. 일반적으로 유사성 측정이 동일 차원의 두 벡터 A, B 간에 이루어지는 것과는 다르게 음성은 유사성을 측정하려는 두 벡터간의 차원이 다르게 된다. 즉, 1차원의 벡터 A, B사이의 Distance(유사성과 반비례 관계)는 A-B로, 2차원 벡터 A, B의 Distance는 $\sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$로 벡터의 요소끼리 연산한다. 그러나 음성의 경우에는 시간 축에서 비교하는 데이터 개수(프레임 수)가 달라짐으로써 A=(a_0,a_1,a_2,…,a_n), B=(b_0,b_1,b_2,…,b_m), n≠m인 경우가 일반적이다. 따라서 두 음성의 유사성을 측정하기 위해서는 시간 축에서 발생하는 차이를 극복하기 위한 알고리즘이 필요한데 대표적인 알고리즘은 DTW와 HMM이다. 본 연구에서 DTW를 사용하였으므로 그에 관해 알아보도록 하겠다.

DTW는 기준이 되는 음성신호의 패턴과 입력된 음성 신호간의 유사도를 동적 프로그래밍(dynamic programming)을 이용해 시간 축에서 차이를 보상하기 위한 방법이다. 예를 들어 길이가 M인 입력 음성 패턴을 T=T(1),T(2). ..., T(M)길이가 N인 기준 패턴을 R=R(1),R(2),...,R(N)라고 하면 두 패턴간의 유사도 D는 다음과 같이 누적거리로 표현된다.

$$D = \sum_{n=1}^{N} d(R(n), T(w(n))) \quad (1\text{-}7)$$

이때 d(R(n),T(w(n)))는 R의 n번째와 T의 w(n)번째의 국부적 유사도(Local Distance)이며, DTW는 두 패턴간의 누적 거리 최적화하는 (m,n)평면의 최적 경로 m=w(n)를 찾는 방법이다.

DTW의 경우 기준 모델 집합의 작성은 간단하다. 인식하고자 하는 명령어들을 발음하고 분석한 후 연속된 프레임들의 특징 벡터들을 저장하고 있으면 된다. 인식 시에는 입력된 음성을 분석해 특징 벡터를 추출한 후 이들 기준 모델 집합의 구성원과 개별적으로 DTW하여 가장 적은 누적 거리를 주는 구성원을 찾으면 된다.

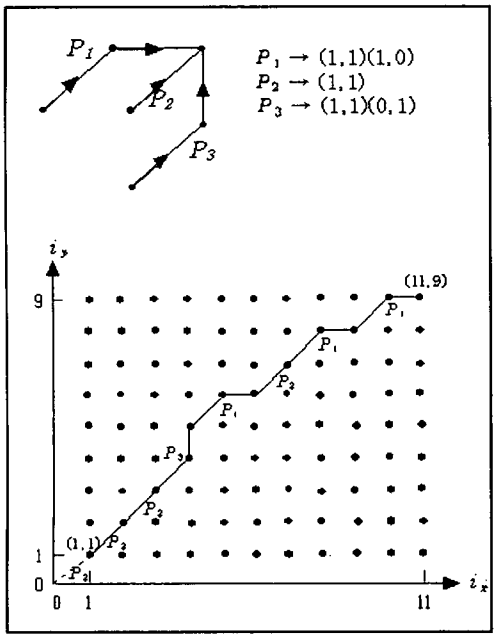

그림 7. DTW 알고리즘의 예

3.1.4 Final Decision(Decision Logic)

음성 인식 시스템은 인식하고자 하는 단어에 대한 모델을 미리 갖고 있다가 입력된 음성을 저장된 모든 모델과 비교해 그 유사도가 최대인 모델을 인식의 결과로 간주한다. 따라서 등록되지 않은 단어가 발음된 경우에도 일단은 인식 결과가 출력된다. 그러므로 최종 인식 결정 과정에서는 인식결과가 타당한지 검증해서 타당하면 인식 결과를 출력하고, 그렇지 않으면 거부해 인식기의 오 동작을 막는다. 이 타당성의 검증은 주로 입력 음성에 대해 가장 유사한 결과를 갖는 모델과의 유사도 값과 모델을 생성할 때 계산한 유사도 값 사이의 차이의 정도를 이용한다.

3.2 시스템 구성

Main Board는 기본적으로 DSP Processor를 비롯하여 CODEC, Oscillators, PPI(Programmable Peripheral Interface), SRAM, Parallel Port, PAL(Programmable Array Logic)등으로 구성이 되었다. 이 회로의 동작은 프로그램을 호스트 PC로부터 다운로드해서 동작하게 되는데 PC와의 호스트 인터페이스는 Board의 Parallel Port를 통해 이루어진다. 여기서는 DSP Processor와 CODEC[3], PPI에 대해서만 간단히 살펴보기로 하겠다.

3.2.1 DSP Processor

현재 사용되고 있는 프로세서의 종류는 크게 범용 프로세서, 마이크로 컨트롤러, 그리고 DSP Processor로 나눌 수 있다. 범용 프로세서는 보통의 컴퓨터가 수행하는 여러 가지 일들을 소화할 수 있도록 필요한 기능을 갖추고 있다. 예를 들면 Intel Pentium CPU가 여기에 속한다. 마이크로 컨트롤러는 주로 주변의 장치들을 관리하기에 적합하도록 만들어진 프로세서이다. 주변 장치에 여러 가지 명령을 내리면 되는 프로세서이므로 연산을 빨리 해야 하거나, 혹은 기억 용량이 커야 할 필요는 없으므로 그에 맞추어 프로세서의 기능들을 줄이고 대신 가격을 싸게 하는 등의 장점을 갖도록 설계되어 있다. Intel 80196등이 이에 속한다. DSP Processor는 디지털 신호처리(영상처리, 음성처리, Filtering등)에 알맞게 설계된 프로세서이다.

본 연구에서는 TI(Texas Instrument)사의 DSP Processor인 TMS320C32를 사용하여 구현하였다. 이 Processor는 실수 연산에 적합하도록 설계된 산술연산장치(ALU : Arithmetic Logic Unit)뿐만 아니라, 32bit Data Bus와 24bit Address Bus, 2개의 Timer, 2-Channel DMA, 그리고 serial port 등이 갖추어져 있다.

3.2.2 CODEC

CS4213(본 연구에 사용된 CODEC)은 스테레오 오디오 CODEC으로서 단일 칩으로 구성되어 있고 A/D & D/A converting, Filtering, Level Setting 등을 수행하며 동작 방법에 따라 Serial Mode(SM) 3,4,5로 구분되는데 본 연구에서는 SM-4(Master Sub-Mode)를 사용하였다. 그리고 SM-4에서의 Sampling rate은 Processor Memory map의 814000H번지에 값을 적절히 넣어줌으로써 8KHz-48KHz 사이에서 8가지의 Sampling rate 중 하나를 선택할 수 있는데 본 연구에서는 8KHz의 Sampling rate를 사용하였다.

3.2.3 PPI (Programmable Peripheral Interface) 8255

8255는 INTEL사의 범용 Parallel I/O Interface 이다. PPI는 프로그래밍을 통해 자신의 기능을 정하고 CPU와 주변장치 사이에서 그 규칙대로 신호들을 해석하여 전달해 주는 일을 하는 장치이며 8255는 3개의 Port(Port A, B, C)의 8bit 입출력을 갖고있는 범용 입출력 장치이다. 8255가 프로그래밍이 가능하다는 것은 필요에 따라 동작 규칙을

3) COder DECoder의 준말로 신호의 Coding과 Decoding을 담당하는 Unit.

바꿀 수 있다는 것이다. 8255의 세 가지 모드는 Processor control register에 특정 값을 써 넣음으로써 각 Port가 입력인지 출력인지 그 규칙을 정하게 되는 것이다.

본 연구에서는 8255의 각 Port가 정해진 기능만을 수행하지 않고 상황에 따라 계속해서 변하며 사용자와 Processor 사이의 중개자 역할을 수행한다. 그리고 사용자의 요구에 따른 외부장치들의 Control 신호도 8255를 통해 내 보내게 된다. 이에 관한 내용은 뒤에 나오는 그림 9에서 설명된다.

3.3 시스템 구현

앞에서 설명되었던 DSP Processor를 이용한 Speech Recognition을 구현하기 위해 C-언어와 Assemble 언어를 이용하여 Programing 하였는데 그 실행 과정을 블록도로 보면 그림 8과 같다.

그림 8의 블록도를 보면, 먼저 마이크를 통해 입력된 Signal을 CODEC에서 받아 Coding(A/D Converting)을 해서 Processor에게 전달을 하고 Processor는 CODEC으로부터 전송되어오는 Data

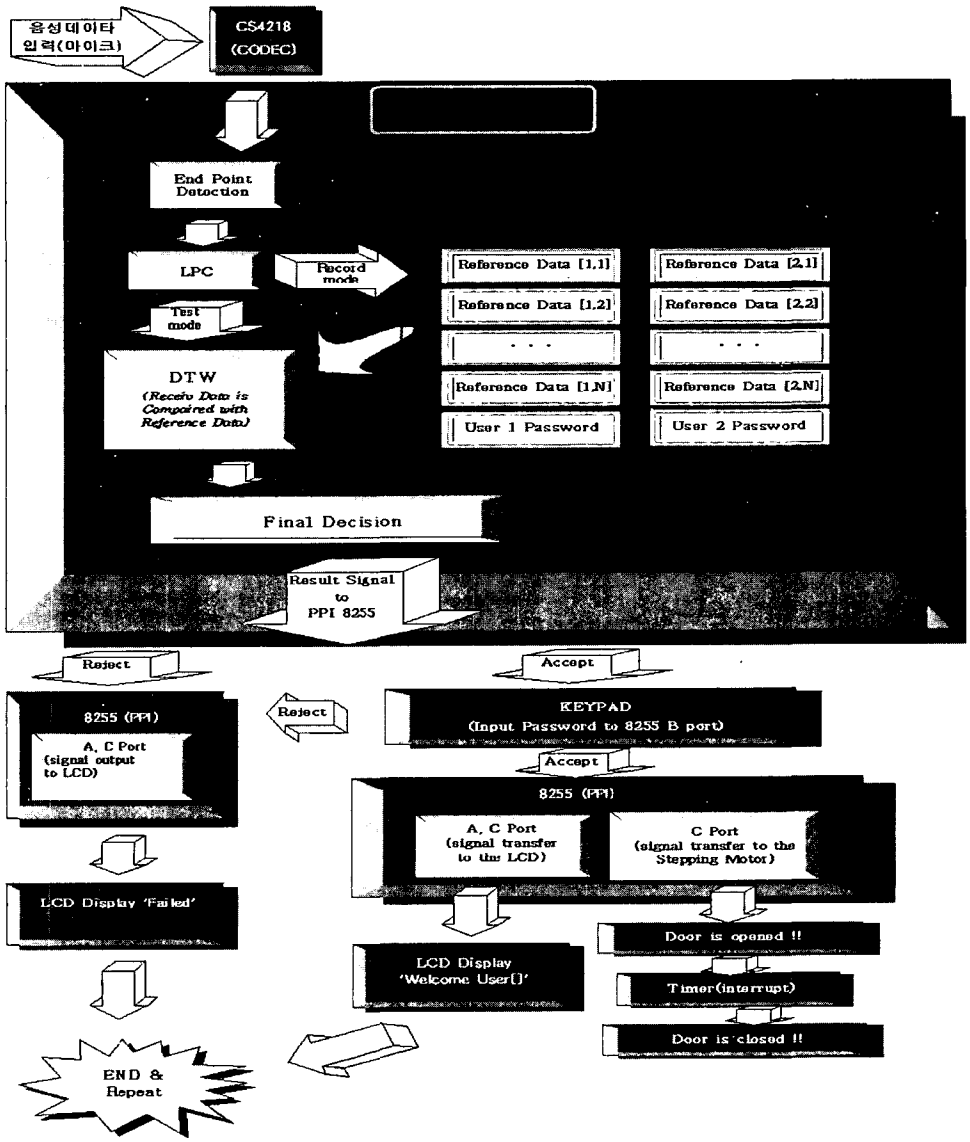

그림 8. 구현되어진 System의 블록도

들을 Endpoint detection해서 어디서부터 어디까지 실제 입력되는 speech data인지를 판단하게되고, 이렇게 해서 들어온 하나의 Data 묶음을 LPC(Linear Predictive Coding)과정을 통해 Reference Data(Filter 계수들)로 저장을 하게 된다. 이러한 과정을 N번 수행하여 Reference Data들(Codebook)을 모두 저장한 후 실제 Test Data가 입력이 되면 다시 end point detection과 LPC 과정을 수행하고 여기서 나오는 Data와 Reference data를 DTW(Dynamic Time Wrapping)과정을 통해 각각을 비교하여 N개의 Distance 값들을 Final Decision으로 넘겨주게 된다. Final Decision 과정에서는 DTW를 수행한 결과를 가지고 Test로 들어온 Speech data가 등록된 Speaker에 의한 Data가 맞는지는 판단하게 된다. 그리고 그 결과를 각각의 상황에 맞게 PPI 8255 등을 통해 외부의 장치들로 내 보내게 된다.

User와의 Interface와 문을 열고 닫는 동작을 위한 제어를 위해 앞에서 설명했듯이 PPI 8255를 사용하였다. 8255의 A Port 8bit와 C Port의 상위 4bit은 LCD Display Control용으로, B Port는 Keypad에서 들어오는 Data의 검출에, C Port의 하위 4bit은 Stepping Motor Control에 각각 사용하였다.

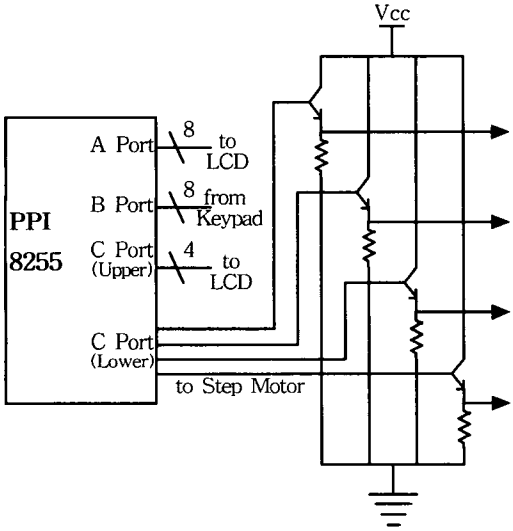

그림 9. 8255 Port 구성 및 모터구동회로

만일 Speaker가 등록된 Speaker가 아니라면 8255를 통해 LCD에 Error 메시지를 출력하게 되고 만일 Speaker가 등록된 Speaker라면 8255를 통해 LCD에 인증메시지를 출력하고 동시에 Stepping Motor[4]에 신호를 내어보내서 문을 여는 동작을 하게 하였다. 그리고 문이 다 열리면 Processor에서 Timer Interrupt가 동작을 해서 일정한 시간 이 지나면 다시 8288에게 문을 닫는 신호를 내어보내서 문이 닫히게된다.

이곳에 사용된 Motor 구동 회로 부분의 회로도는 그림 9에서 보인바와 같고 전체적인 Main Board는 그림 10과 같다. 그리고 마지막으로 그림 11에서는 아크릴로 제작되어 Step Motor를 이용해 자동으로 동작하는 자동문을 보였다.

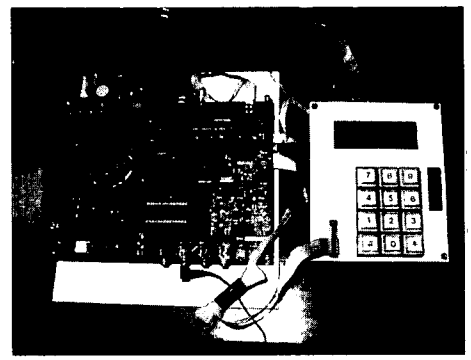

그림 10. 전체 회로와 Keypad, LCD의 모습

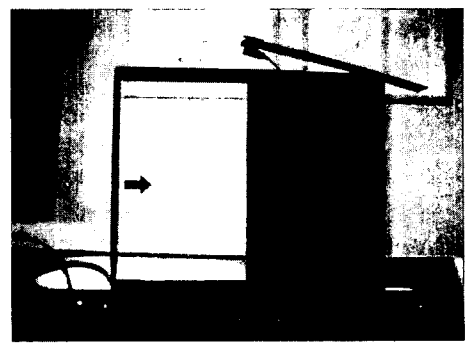

그림 11. Step Motor로 동작하는 자동문

4. 연구 결과

미국에서 상용화되는 Speaker Verification(화자인증) System은 1000명의 다른 사람(Impostor)이 시험하여 1명 이하의 사람을 잘못 승인(False Acceptance)하고, 본인이 100회 발성하여서 1번 이하의 잘못된 거절(False Rejection)을 화자 인식 시스템의 최소 규격으로 삼고 있다. 이러한 규격을 조금 축소하여 Test를 실시하였다. 조건은 음절수

4) EM-196 Motor로 5개의 입력 라인이 있다.

(1-6)에 따른 인식률을 고려했고, 다음으로는 잘못된 화자(20명이 각3회Test)를 고려했다. 등록된 화자 인식률은 각 음절 당 20번 반복 Test를 하였다. 그리고 본 연구의 Final Decision 과정에서 단순한 Distance 값으로 비교하는 방법과 frame 길이(∝ 음절수)를 Normalize하여 Distance와 비교하는 방법으로 나누어 Test를 하였는데 표 2는 frame normalize[5]를 적용하지 않았을 경우의 Test이고 표 3은 frame normalize를 적용한 경우의 Test Data이다.

표 2. Frame normalize되지 않은 결과

	음절수에 따른 인식률 (Impostor 20명)						Total
	1 음절	2 음절	3 음절	4 음절	5 음절	6 음절	
Impostor 거절 확률	48 /60	53 /60	57 /60	60 /60	60 /60	60 /60	338 /360
User 인식 확률	40 /40	39 /40	40 /40	37 /40	32 /40	22 /40	210 /240
Total	88 /100	92 /100	97 /100	97 /100	92 /100	82 /100	548 /600

표 3. Frame normalize된 결과

	음절수에 따른 인식률 (Impostor 20명)						Total
	1 음절	2 음절	3 음절	4 음절	5 음절	6 음절	
Impostor 거절 확률	58 /60	59 /60	60 /60	60 /60	60 /60	60 /60	357 /360
User 인식 확률	40 /40	39 /40	40 /40	40 /40	40 /40	32 /40	231 /240
Total	98 /100	98 /100	100 /100	100 /100	100 /100	92 /100	588 /600

이상의 performance test 결과를 보면 Frame Normalize 방법이 간단하지만 필수적임을 알 수가 있었고, 위의 Data에서 알 수 있듯이 Frame Normalized Data 역시도 6음절 이상이 되면 인식률이 떨어지는 것을 알 수가 있는데, 이것은 Memory의 용량[6]을 감안하여 Frame 수를 40으로 제한해 놓았기 때문에 40 Frame(일반적인 대화속도로 약 6음절 정도)을 넘어가는 단어가 입력이 된다면 인식률이 떨어지게 되기 때문이다. 그리고 단음절인 경우 비슷한 음성의 특성을 가진(귀를 통해 서도 분별하기 힘든) 사람의 경우에 어쩌다 인식이 되는 경우가 있는데 이것은 Reference Data 입력 시에 제한을 두거나(3음절이상...등의) 프로그램 상에서 너무 짧은 frame을 거절하는 등의 방법을 통해 해결할 수가 있으므로 크게 문제가 되지 않는다면 3-5음절 정도의 보편적인 System인 경우(O표시된 부분) 설계 시에 원했던 performance를 얻을 수가 있다.

현재 구현되어진 System의 성능은 User를 2명으로 한정한다. 이유는 Memory의 제한 때문이다. 그리고 reference data를 1인당 3회식 저장하여 기준이 되는 data로 삼고있으며 구현하는 과정에서 본 연구의 중심이 음성인식을 이용한 화자 인증이지만 음성만으로 열리는 문인 경우엔 보안에 치명적인 문제(정밀하게 녹음이 되어 재생이 되는 경우 문이 열림)가 있었기 때문에 이를 위한 해결책으로 2중 보안을 선택하였다. 즉 Speech data 외에 User를 확인할 수 있는 또 다른 방법을 추가하였는데 바로 Password(4-digit number)이다. Speech data와 Password 이 두 가지가 모두 등록된 화자의 것과 일치하였을 경우에만 문이 열리도록 설계를 하였다. 그리고 초기 구현에서는 Display 장치를 7-segments를 사용하였다. 하지만 7-segments의 영문 표기가 제한적이고 불편하였기 때문에 7-segments 를 LCD로 교체를 하였다. 이 두 가지 즉, Password 입력과 LCD Display를 위해 그림 10에서 보인바와 같이 Keypad와 LCD를 함께 하나의 User I/O module로 구현하였다.

5. 결론

DTW 알고리즘을 이용한 화자인증에 관한 이번 연구를 통해 Text Dependent한 System에서 DTW 알고리즘이 어떠한 성능을 발휘하는가를 확증할 수 있었다. 이것은 큰 장점을 가지는데 그것은 구현이 상대적으로 쉬우면서도 그 성능이 떨어지지 않고 원하는 performance를 만족한다는 것이다. 하지만 본 연구를 통해 문제점 또한 발견할 수가 있었는데 때문에 다음과 같은 사항들이 향후 계속해서 연구되어져야할 과제로 여겨진다.

먼저 본 연구에서 구현되어진 System은 실험실

5) Distance*10/frame < 39(환경에 따라 조금씩 변경)으로 Normalization하였다.

6) External memory(SRAM-In board) size : 64k word(=512KB)

에서의 성능과 많은 사람들이 오가는 Hall에서의 성능에 다소의 차이가 있음을 확인하였는데 이것은 주변에서 사람들의 웅성거리는 소리가 음성 data처럼 system으로 들어가서 User의 음성 인증을 방해하기 때문이었다. 이를 해결하는 매우 효율적인 방법이 계속해서 연구되어져야할 것이다.

또한 System 자체의 Hardware적인 문제로서 Board의 빈약한 Memory 때문에 더 많은 사용자의 등록과 더욱 긴 문장의 사용에 큰 제한이 있었다. 그리고 그 문제는 Sampling rate에도 제한을 주는데 더 정교하게 Sampling을 하게되면(예들 들어 24Kbps)그에 따라 연산속도도 현재의 System보다 3배 이상 빨라져야하며 구현을 위한 수많은 계수들(buffer 포함)중 상당수는 그 크기가 커져야 하는데 이것은 Memory의 크기와 직접적인 관계에 있으므로 제한요인이 된다. 때문에 향후에 이러한 문제를 효율적으로 해결하는 Hardware적이거나 Software적인 방법들이 충분히 모색되어져야 할 것으로 사료된다.

참 고 문 헌

[1] Lawrence Rabiner, Biing-Hwang Juang, "Fundamentals of speech Recognition", Prentice-Hall International Inc., 1993

[2] Vinay K. Ingle, John G. Proakis, "Digital signal Processing using MATLAB", Brooks/Cole Publishing Company, 2000

[3] Texas Instrument "TMS320C3x User's Guide", Texas Instrument, 1997

[4] 이지홍, 서일 DSP(주) 기술연구소, "DSP Chip의 활용", 서일DSP(주), 2000

[5] Adel S. Sedra, kenneth C. Smith, "Microelectronic circuits (3'rd Edition)", Saunders College publishing, 1991

[6] Ronald J. Tocci, Neal S. Widmer, "Digital Systems-Prindiples and Applications(7th Edition)", Prentice-Hall International Inc., 1998

[7] Samir S. Soliman, Mandyam D. Srinath, "Continuous And Discrete Signals And Systems(Second Edition)", Prentice-Hall, 1998

[8] Tomoko Matsui, Sadaoki Furui, "Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMM's", IEEE Transaction on Speech and Audio Processing, VOL.2.NO.3, July 1994