실시간 윈도우 환경에서 DMS 모델을 이용한 자동 음성 제어 시스템에 관한 연구

A Study on the Automatic Speech Control System Using DMS model on Real-Time Windows Environment

이 정 기*, 남 동 선*, 양 진 우**, 김 순 협*
(Jeong Gi Lee*, Dong Sun Nam*, Jin Woo Yang**, Soon Hyop Kim*)

※ 이 연구는 1998년도 광운대학교 학술 연구비 지원에 의하여 이루어진 것임.

요 약

본 논문은 음성인식을 이용한 실시간 윈도우 자동 제어 시스템에 관한 연구이다. 사용된 음성 모델은 수행 속도를 높이기 위해 제안된 가변 DMS 모델을 이용하였으며, 인식 알고리즘으로 이를 이용한 One-Stage DP 알고리즘을 사용한다. 인식 대상단어는 윈도우에서 자주 사용되는 66개의 윈도우 제어 명령어들로 구성한다. 본 연구에서 온라인으로 음성을 처리하기 위해 음성 검출 알고리즘을 구현하였으며, 기존 DMS(Dynamic Multi Section)모델 생성시 고정적으로 적용하던 섹션의 수를 입력 신호의 지속 시간을 고려하여 가변적으로 적용한 가변 DMS 모델을 제안하였다. 또한 윈도우에서 사용자 작업에 의해 현재 상태에 인식 대상으로 불필요한 인식 대상단어가 발생하게 되는데 이를 효율적으로 처리하기 위해 사용 모델을 재구성하여 사용하도록 제안하였으며, 인간의 청각적 특성을 고려하여 음성신호에서 개인의 특성은 제외하고 음성 자체의 특징만을 추출하여 특징 벡터를 생성하는 인지 선형 예측(Perceptual Linear Predictive)분석 방법을 이용하였다. 시스템 성능 평가 결과 가변 동적 다중 섹션 모델(Variable DMS model)과 기존의 DMS 모델은 인식률 면에서는 거의 동일하지만 인식 수행 속도는 제안된 모델의 계산량이 기존 모델보다 작기 때문에 향상되었고, 다중 화자 독립 인식률은 99.08%, 다중 화자 종속 인식률은 99.39%의 인식률을 나타내었으며, 실제 노이즈가 있는 환경에서 화자독립실험의 경우 96.25%의 인식률을 보여 주었다.

핵심용어: DMS모델, 인지 선형 예측, 가변 동적 다중 섹션 모델, 음성인식

ABSTRACT

Is this paper, we studied on the automatic speech control system in real-time windows environment using voice recognition. The applied reference pattern is the variable DMS model which is proposed to fasten execution speed and the one-stage DP algorithm using this model is used for recognition algorithm. The recognition vocabulary set is composed of control command words which are frequently used in windows environment. In this paper, an automatic speech period detection algorithm which is for on-line voice processing in windows environment is implemented. The variable DMS model which applies variable number of section in consideration of duration of the input signal is proposed. Sometimes, unnecessary recognition target word are generated. therefore model is reconstructed in on-line to handle this efficiently. The Perceptual Linear Predictive analysis method which generate feature vector from extracted feature of voice is applied. According to the experiment result, but recognition speech is fastened in the proposed model because of small loud of calculation. The multi-speaker-independent recognition rate and the multi-speaker-dependent recogniton rate is 99.08% and 99.39% respectively. In the noisy environment the recognition rate is 96.25%

Key words: Dynamic multi section, Perceptual linear predictive, Variable DMS model, Speech recognition
투고분야: 음성처리(2.5)

I. 서 론

21세기 정보의 물결 속에서 사람들은 다양한 정보 처리 서비스를 제공받을 수 있게 되었다. 이러한 서비스는 대부분

* 광운대학교 컴퓨터공학과
** 춘천기능대학 전자과
접수일자 : 1999년 11월 19일

컴퓨터를 통하여 이루어지고 있다. 반면 일반 사람들은 컴퓨터를 사용하는데 대부분 미숙련자들이다. 이러한 사용자를 위해 새로운 인터페이스, 편리하고 쉬운 인터페이스의 연구가 많이 진행 중이다.[6] 특히, 인간과 기계간의 가장 쉬운 의사 소통 도구로 연구되는 음성을 이용한 인터페이스에 관한 연구가 실용화를 위하여 전화망 서비스, 차량 제어 시스템, 윈도우 제어, 공장 자동화, 의료 기기 등의 분야에서 연구 중이다.

본 논문에서는 이러한 인간-기계 사이에 가장 편리하고 쉬운 인터페이스로 고려되어지는 음성을 이용하여 윈도우 시스템을 제어함으로 사용자에게 편의성, 작업의 효율성, 작업의 용이성 등의 장점을 제공하기 위해 윈도우 시스템에서 작동하는 윈도우 음성 제어 시스템을 구현 및 설계하였다.

본 연구에서 구현된 시스템은 고립단어인식 시스템의 단점을 장점으로 수용할 수 있도록 하였으며, 사용모델은 Dynamic Multi-Section을 사용하였으며, 인식 알고리듬은 연결어 인식을 단어 인식 시간 정도의 시간에 처리해 주는 OneStage DP 방법을 이용하여 단어 인식에 사용하였다. 또한 특징 벡터로는 LPC, Mel-Cep, PLP 각각 13차의 특징 벡터를 비교 실험하여 PLP[5]를 인식 시스템의 특징 벡터로 사용하였으며, 실시간 음성의 끝구간 검출을 위해 300msec 마다 임계값을 재설정 하였고 절대 에너지와 영교차율 값을 이용하였으며, 실제 윈도우에서 음성을 입력받을 때 발생 할 수 있는 주변 잡음의 처리를 위해 입력 신호에 대한 최대 Peak Amplitude를 사용하였다. 또한 사용자의 작업 공간을 최대한 침범하지 않고 사용자에게 인식 시스템의 현재 상태 등의 정보 제공을 위해 최소 공간을 활용하는 인터페이스를 적용하여 시스템을 구현하였다.

II. 본 론

2.1. 자동 음성 구간 검출[2][6]

본 논문에서 구현된 자동 음성 구간 검출 알고리듬은 크게 기본 신호 처리 구간(Basic 1,2), 음성 시작 버퍼 검출 구간, 음성 끝 버퍼 검출 구간의 3 부분으로 나뉘어 진다. 기본 신호 처리 구간에서는 DC Offset을 제거하기 위해 입력 버퍼의 최초 5 Frame 값의 Mean Value를 계산하여 이후 데이터로부터 차감 해준다. 또한 임계값 설정을 위해 입력 신호로부터 절대 에너지와 ZCR을 계산하고 이전 입력 버퍼에서 설정된 에너지와 비교하여 음성의 구간검출에 사용된다. 음성 시작 구간 검출에 사용되는 변수는 PreEnergy(이전 버퍼의 평균 절대 에너지), Energy (현재 버퍼의 평균 절대 에너지), EnergyBig(현재 버퍼에서 설정된 임계값), ZCRSum(현재 버퍼에서 계산된 ZCR의 총 합), ZCRThreh(이전 버퍼의 ZCR로부터 계산된 임계값)이고 다음 조건은 사용된 음성 시작 구간 검출 조건을 나타낸다.

Condition 1.

(PreEnergy > Energy) ∩ (EnergyBig > 750) (1)

Condition 2.

(PreEnergy > Energy) ∩ (ZCRSum < ZCRThreh) (2)

위의 조건을 만족하여 음성의 시작 구간을 검출 후 끝점 검출은 간단하게 이전 구간의 절대 에너지로 설정된 임계값을 이용하여 예비 검출, 확인 검출의 두 부분으로 분리되어 처리된다. 또한 실제 윈도우를 사용자가 사용할 경우 실험 환경에서 발생되는 주변 잡음이 있기 마련인데, 이러한 잡음을 처리하기 위해 현재 입력되는 입력 신호의 평균 Peak Amplitude와 최대 Peak Amplitude를 이용하였다.

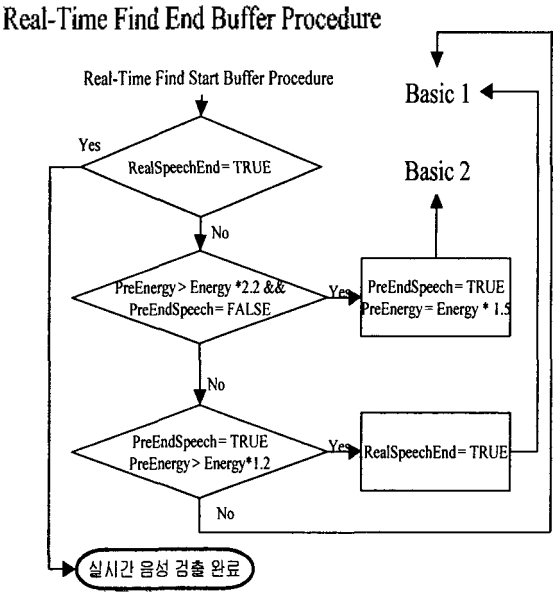

그림 1. 음성 종료 부분 검출 과정
Fig. 1. The end point detection process.

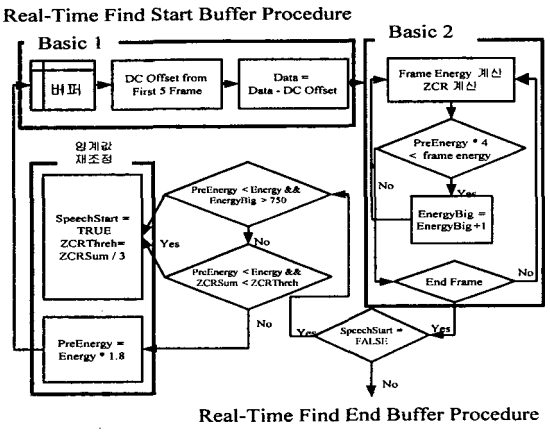

그림 2. 음성 시작 부분 검출 과정
Fig. 2. The beginning point detection process.

그림 1은 입력 되는 버퍼를 이용하여 음성의 시작 부분이 포함된 버퍼를 검출하는 처리 과정이며 그림 2는 입력이 시작된 후 계속적으로 입력되는 버퍼에서 음성 신호의 종료 부분을 찾아내는 알고리듬에 관한 처리 과정을 나타낸다. 구현된 처리 과정 수행 결과 음성 검출 시간은 발성 종료후 600msec 이내에 종료된다.

2.2. 가변 수 섹션 DMS 모델

인간은 Syllable 길이의 음성으로부터 정보를 얻어 이를 이해하고 의사 소통 정보로 이용한다.[5]

일반적으로 DMS 모델[3][4] 생성 시 섹션의 수를 모든 인식 대상 단어에 대하여 고정적으로 사용한다. 그러나 본 논문에서는 이러한 부분에서 발생되는 불필요한 계산 시간과 메모리를 줄이고, 궁극적으로는 인식 시간을 줄이기 위하여 인식 단어의 지속 시간에 따라 섹션의 수를 가변적으로 설정하는 가변수 섹션 DMS 모델을 제안한다.

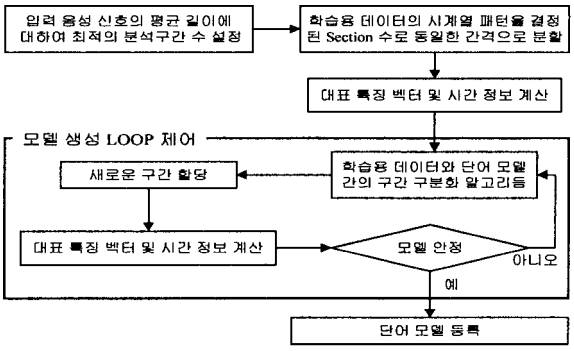

그림 3. 가변 섹션 모델 생성 과정
Fig. 3. The process of variable section model.

제안된 모델은 66개의 윈도우 제어 명령어의 DMS 모델로 구성되어 있으며 특징 벡터로 PLP 13차를 사용한다. 66개의 명령어는 400msec~1200msec의 지속 시간을 가진 단어들로 구성된다. 모델은 다중 화자가 발성한 단어를 이용하여 생성하고 각 단어에 대한 섹션 수 결정은 각 단어들의 지속 시간의 평균을 실험 결과로 나타난 한 섹션의 크기 45msec로 나누어 결정한다.

$$\text{j 번째 단어의 섹션 수} : \frac{\sum_{n=1}^{N} DT(n)_j}{T} \qquad (3)$$

DT : 입력 신호의 지속 시간
j : 발성 단어
n : 발성 화자
T : 할당된 섹션 당 지속시간 길이

다음 표 1은 구현된 음성 제어 시스템에서 인식 가능한 대상 단어와 다중 화자 발성 데이터에 대한 지속 시간과 그에 따라 위의 식 (3)에 의해 결정된 결과를 나타낸다.

표 1. 인식 대상 단어 당 섹션 수
Table 1. Number of section for each recognition words.

음절 수	지속 시간	결정된 섹션 수
1, 2 음절	400~700msec	9 Section
3, 4 음절	750~1150msec	15 Section
5, 6 음절	1000~1200msec	20 Section

III. 구현된 음성 제어 시스템

3.1. 사용자 인터페이스

구현된 시스템의 인터페이스는 사용자에게 편하도록 최소한의 사용자 작업 윈도우의 공간을 차지하도록 구성하였으며 사용자에게 현재 인식 가능한 인식 대상 단어 및 모든 상황에서 인식 가능한 대상 단어등으로 분류하여 사용자에게 보여준다.

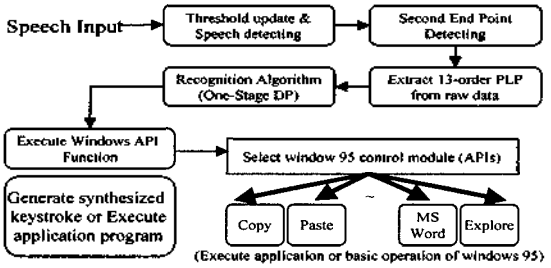

그림 5. 시스템 전체 구성도
Fig. 5. Whole system diagram.

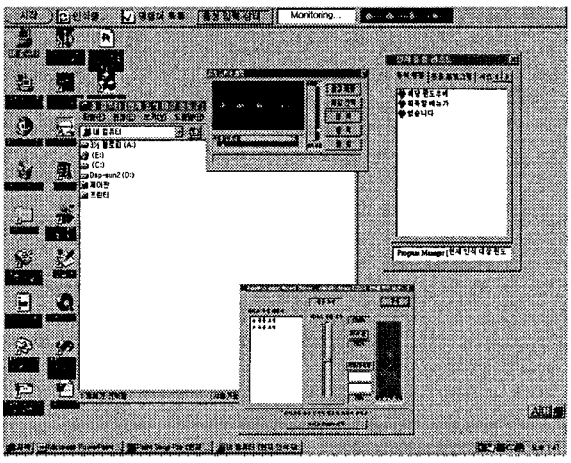

그림 6. 구현된 시스템 인터페이스
Fig. 6. Interface of the implemented system.

3.2. 구현된 윈도우 제어 시스템

구현된 시스템은 다음과 같은 간단한 화면 구성을 가진다. 메인 화면과 명령 리스트 출력 화면, 인식 시작을 설정하는 상태 창 그리고 동적으로 갱신되는 메뉴를 출력하는 창이 존재한다.

그림 7. 시스템 메인 화면
Fig. 7. Main system screen.

위 그림 7은 인식 시스템을 처음 수행시킨 화면이다.

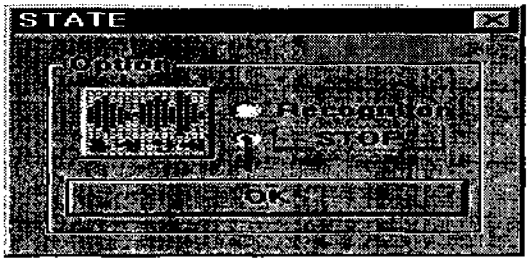

그림 8. 인식 상태 설정을 위한 상태창
Fig. 8. Status window to config recognition status

그림 8은 인식 상태를 설정하기 위한 상태창을 나타낸다. Recognition버튼을 설정하면 인식 가능상태가 된다.

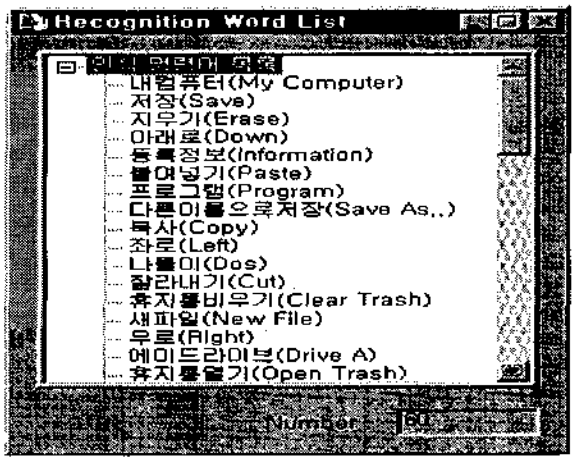

그림 9. 사용자를 위한 명령 리스트 출력 창
Fig. 9. Instruction list window.

IV. 실험 및 고찰

4.1. DB 구축

실험에 사용한 음성 DB는 남성 화자 10인이 각각 3회 발성한 데이터를 이용하여 5인의 2회 발성 데이터는 모델 생성에 1회는 종속 화자 인식 실험을 위하여 사용하였으며, 나머지 5인의 데이터를 이용하화자 독립 인식 실험을 수행하였다. 다음 표 2 는 음성 입력 설정 상황을 나타낸다.

표 2. 입력 데이터 설정
Table 2. Setting for input data.

설정 내용	값
Sample 주파수	11.025(KHz)
Channel 수	Mono(Channel 1)
양자화 bit 수	16bits

4.2. 명령어 분류

구현된 시스템에서 인식 대상으로 하는 인식어는 다음과 같이 크게 2가지 방법을 이용하여 수행한다. 첫째는 keystroke를 synthesize하고 이를 API Function을 이용하여 WM_KEKDOWN, WM_KEYUP message를 발생시킨다. 둘째는 윈도우 상에 존재하는 수행 가능한 파일 또는 자체로는 수행가능하지 않지만 윈도우 상에서 연결된 프로그램이 있는 파일을 수행시키는 방법을 사용하였다. 프로그램의 메뉴에 있는 경우 동적으로 메뉴를 해당 인식 단어에 link 시켜 구동하도록 하였다.

표 3. 입력 음성 설정 사항
Table 3. Input speech setting.

<table>
<tr><th>구분</th><th colspan="5">windows 95 명 령 어</th></tr>
<tr><td rowspan="14">키보드 메세지</td><td>저장</td><td>예</td><td rowspan="5">링크파일과 특별한모듈</td><td>휴지통 비우기</td><td>휴지통 열기</td></tr>
<tr><td>지우기</td><td>인쇄</td><td>네트워크 환경</td><td>끄기</td></tr>
<tr><td>아래로</td><td>아니오</td><td>단축메뉴</td><td>이름순으로 정렬</td></tr>
<tr><td>붙여넣기</td><td>엔터</td><td>종류순으로 정렬</td><td>크기순으로 정렬</td></tr>
<tr><td>등록정보</td><td>취소</td><td>날짜순으로 정렬</td><td>자동순으로 정렬</td></tr>
<tr><td>다른 이름으로 저장</td><td>창닫기</td><td rowspan="9">Netscape 제어</td><td>인터넷</td><td>다음사이트</td></tr>
<tr><td>복사</td><td>탭</td><td>이전사이트</td><td>홈</td></tr>
<tr><td>좌로</td><td>확대</td><td>중지</td><td>북마크</td></tr>
<tr><td>새파일</td><td>축소</td><td>답장보내기</td><td>메일보내기</td></tr>
<tr><td>우로</td><td>열기</td><td>메일확인</td><td>주소록 보기</td></tr>
<tr><td>다음창</td><td>위로</td><td>넷 검색</td><td></td></tr>
<tr><td>이전창</td><td>시작</td><td colspan="2"></td></tr>
<tr><td>닫기</td><td colspan="3"></td></tr>
<tr><td rowspan="4">실행 파일</td><td>내 컴퓨터</td><td colspan="2">에이(A) 드라이브</td><td>이전페이지</td><td>파일관리자</td></tr>
<tr><td>프로그램</td><td colspan="2">비(B) 드라이브</td><td>전화걸기</td><td>담색기</td></tr>
<tr><td>나들이</td><td colspan="2">시이(C) 드라이브</td><td>도와줘</td><td>다음 페이지</td></tr>
<tr><td>잘라내기</td><td colspan="3"></td></tr>
</table>

4.3. 사용 API Function

인식 된 결과는 다음에서 제시하는 API함수에 의하여 실제 windows 상에서의 수행으로 이어진다.

가. Keyboard Message Synthesize

```
Function prototype : void keybd_event(
        BYTE bVk, // virtural-key code
        BYTE bScan,// hardware scan code
        DWORD dwFlags, //option
        DWORD dwExtraInfo //additional data  );
Example : "P"를 누른 메시지를 발생시킴
keybd_event( P ,0x19, KEYEVENTF_KEYUP,0);
```

나. File 또는 응용 프로그램 수행

```
Function prototype : HINSTANCE ShellExecute(
        HWND hwnd,  // handle to parent window
        LPCTSTR lpOperation,// specifies operation
        PCTSTR lpFile,// filename or folder name
        LPCTSTR lpParameters, //specifies executable-file
        LPCTSTR lpDirectory,// specifies default directory
        INT nShowCmd  // whether file is shown
        );
Example : "recycled.lnk"이란 파일을 수행 시킴
::ShellExecute( NULL,  "open",  "recycled.lnk",  NULL,
            NULL, SW_SHOW);
```

다. 메뉴의 동적 Link

동적메뉴 링크에 대한 구현은 다음의 API함수들을 이용하여 구현 하였다.

GetMenu(), GetMenuItemCount(), GetMenuString()
GetMenuState(), GetMenuItemID()

4.4. 실험 결과

인식 실험은 첫째, 파라메타 비교 실험, 둘째, 섹션에 따른 실험, 셋째, 구현된 시스템에서의 실험으로 나누어 수행된다.

표 4. 화자 종속 파라메타 비교 실험 결과
Table 4. Result from speaker dependent experiments.

	5 섹션	6 섹션	7 섹션	8 섹션	9 섹션	10 섹션	11 섹션	12 섹션	13 섹션	14 섹션
A	27.2	57.5	66.6	75.7	92.4	89.3	95.4	96.9	100	100
B	37.8	50.0	62.1	75.7	86.3	89.3	93.9	96.9	100	100
C	34.8	48.4	68.1	80.3	89.3	93.9	96.9	100.	100	100
D	36.3	42.4	60.6	74.2	83.3	86.3	89.3	96.9	95.4	95.4
E	34.8	40.9	65.1	72.7	87.8	90.9	93.9	93.9	95.4	96.9
평 균	34.2	47.8	64.5	75.7	87.8	89.9	93.9	97.2	98.7	99.0

표 5. 섹션 수에 따른 인식율 비교 실험 (종속)
Table 5. Recognition rates depends on number of sections.

	화자A	화자B	화자C	화자D	화자E	평 균
LPC15	92.42	93.93	96.96	96.96	93.93	94.84
LPC20	98.48	95.45	100.0	95.45	96.96	97.27
Mel15	96.96	93.93	98.48	95.45	95.45	96.06
Mel20	98.48	96.96	96.96	98.48	100.0	98.00
PLP15	100.0	100.0	100.0	96.96	98.48	99.09
PLP20	100.0	100.0	100.0	96.96	100.0	99.39
평 균	97.72	96.71	98.73	96.71	97.47	97.47

표 4.의 결과 LPC, Mel-Cep, PLP 특징 벡터 각각 13차를 이용하여 DMS 모델의 섹션 수를 15, 20 섹션으로 변화시키면서 비교 실험을 수행하였다. 실험 결과 PLP - 13차 DMS 20 섹션의 경우 화자 종속으로 99.39의 인식율로 가장 우수한 결과를 나타내었다. 또한 표 5.의 결과를 이용하여 각 단어당 섹션의 수를 결정하는데 사용하였다.

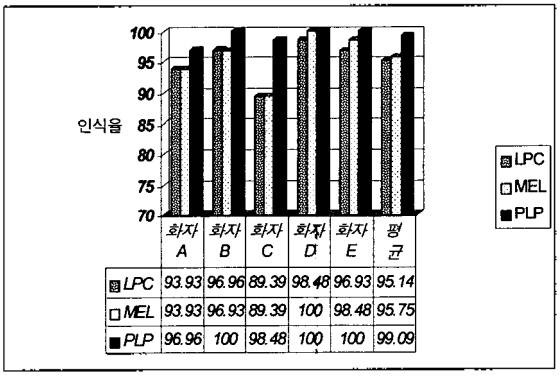

그림 10. 제안된 모델과 PLP 13차 인식 (독립)
Fig. 10. The proposed model and PLP 13th recognition.

V. 결 과

본 연구에서는 기존 DMS 모델 생성에 고정적으로 사용하던 섹션의 수를 인식 대상 단어의 지속 시간에 따라 변경되는 가변 섹션을 적용하는 모델을 제안하였다. 본 시스템에서 사용한 인식 알고리듬과 파라메타를 이용하여 인식율은 거의 변화가 없었지만 인식 시간적 측면에서는 약 20% 개선되었다. 인식율은 66개의 제어 명령에 대하여 화자 독립 99.08%의 인식율을 얻었으며 윈도우에서 음성을 추가적인 인터페이스 수단으로 사용 가능하도록 음성 제어 시스템을 구현 하였다.

참 고 문 헌

1. L.R. Rabiner, B.H. Juang, "Fundamentals of Speech Recognition," Prentice Hall, 1993.

2. L.R. Rabiner, M.R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterance," The Bell system Technical Journal, Vol. 54, No. 2, PP297-315, Feb. 1975.
3. H. Sakoe, S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Transactions on communications, pp159-165, 1978.
4. Hermann Ney, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition," IEEE Transaction on Acoustic, Speech,and Signal Processing, Vol. ASSP-32, NO. 2, pp263-271, April, 1984.
5. H. Hermansky, "Should Recognizers Have Ears?," Proc. ESCA Tutorial and Research Workshop on Robust Speech, 1994.
6. 남동선, 이정숙, 이성권, 김순협, 이황섭, "음성 인식을 이용한 Windows 95 제어 시스템의 구현," 한국 음향 학회 학술 발표회 논문집 제 17 권 1호 pp43-46, 1998. 6.

▲이 정 기(Lee Geong Gi) 1971년 10월 1일생

1995년 3월～1997년 2월 : 한려산업대학교(공학사)
1997년 3월～2000년 2월 : 광운대학교 대학원 컴퓨터공학과 (공학석사)
2000년 1월～현재 : 현대증권 사이버 추진팀 음성인식 담당
※주관심 분야 : 음성인식, 신호처리

▲남 동 선(Nam Dong Sun) 1974년 8월 23일생

1993년 3월～1997년 2월 : 용인대학교 (이학사)
1997년 3월～1999년 2월 : 광운대학교 대학원 컴퓨터 공학과(공학석사)
1999년 3월～현재 : 인포텍 시스템 연구원
※ 주관심 분야 : 음성인식, 신호처리

▲양 진 우(Jin Woo Yang)
한국음향학회지 제 15권 4호 참조

▲김 순 협(Soon Hyop Kim)
한국음향학회지 제10권 1호 참조