음성인식을 이용한 Windows 95 제어 시스템의 구현

The Implementation of Windows 95 Control System with Speech Recognition

남동선*, 이정숙*, 이성권*, 김순협*, 이항섭**

(Dong-Sun Nam, Jung-Suk Lee, Sung-Kwon Lee, Soon-Hyob Kim, Hang-Seop Lee)

*Kwangwoon University, **Electronics Telecommunications Research Institute

요 약

본 논문은 컴퓨터 사용에 미숙한 초보자나 키보드나 마우스를 사용할 수 없는 신체적인 조건을 가진 장애인 또는 PC사용에 미숙한 사용자들을 위해 기존의 인터페이스에 추가적으로 음성을 사용하여 더 효율적인 작업 환경을 만들기 위한 음성을 이용한 Window95 환경에서의 음성 인식 시스템 구현에 관한 것이다.

인터페이스 구현을 위해 사용되는 인식 알고리즘으로는 연결어 인식에 사용되는 OSDP[1] 알고리듬을 단독어 인식에 적용하여 사용하였다. 특징 벡터는 화자 독립적인 특성을 지닌 Perceptual Linear Predictive(PLP)[2] 13차 계수를 사용하였다. 인식 대상 어휘는 윈도우 사용자에게 자주 사용되는 60개의 명령어로 설정하였다. 인식된 후 그 결과는 구현된 시스템의 명령 실행 모듈로 전달되어 윈도우 상에서 실제 수행된다. 구현된 시스템에서는 노트북 내장 마이크를 사용하여 음성을 검출하였고 이를 위한 음성 구간 검출 알고리즘을 사용하였다. 기준 패턴은 20대 남성화자 9인이 2회 발성한 데이터를 이용하였고, 화자 독립으로 온라인 인식률은 91.71%이고, 오프라인 인식률은 96.4%의 인식률을 얻었다.

Ⅰ. 서론

현대 사회는 많은 사람들이 함께 모여 생활하고 여러 가지의 수단을 이용하여 서로의 의사를 전달한다. 그 중 인간이 가장 많이 사용하고 가장 먼저 배우고 쉽게 사용할 수 있는 의사 전달 도구는 음성이다. 때문에 기계와 인간과의 의사 전달을 위해 가장 효율적이고 쉬운 의사 소통 도구로 적합한 것은 바로 "음성"일 것이다. 또한, PC의 사용자 계층은 점점 그 범위와 수가 넓어지고 있고 앞으로도 계속 증가할 것이다. 때문에 많은 사람들은 더 편리하고 익히기 쉬운 PC의 사용법을 원하고 생활 속에서 더 많이 PC를 활용하기를 원한다.

따라서 인간에게 가장 친숙한 음성을 이용함으로써 이런 사용자들의 필요를 충족시키는데 도움을 줄 수 있다. 본 논문에서는 windows 95를 사용할 때 기존의 인터페이스 도구(키보드, 마우스)만을 이용하는 것보다

음성을 추가적으로 이용하여 또는 음성 단독으로 이용하는 것이 편리한 60개의 명령어를 선정하여 Windows 95를 제어함으로써 사용자에게 편리한 인터페이스를 제공에 중점을 둔다. 본 논문에서는 노트북상에서 20 Section DMS model (Dynamic Multi-Section)을 사용하였고 음성 특징 파라메타로는 인지 선형 예측(Perceptual Linear Prediction, PLP) 기술을 이용하였는데 이 기술은 인간의 청각 스펙트럼을 모사하고, 음성 정보를 압축하는 효과를 보임으로 인하여 화자독립 음성 인식에 적합한 음성 특징으로 알려지고 있다. 본 연구에서는 PLP 13차 계수를 음성의 특징으로 사용하였다. 이러한 음성의 특징을 이용하여 수행되는 인식 알고리즘은 OSDP (One-Stage Dynamic Programming) 방법을 단독어 인식에 적용하여 사용하였다. 인식 시스템의 모델 훈련을 위해서는 20대 남성 화자 9명이 2회 발성한 데이터를 이용하였고, 화자 독립으로 온라

인 및 오프라인으로 각각 10인 2회, 5인 3회의 실험을 수행하여 온라인 인식률은 91.71%, 오프라인은 96.4%의 인식률을 얻었다.

이 논문의 구성은 다음과 같이 이루어졌다. 2장에서는 시스템에서 사용되는 음성 검출법과 DMS 모델, 및 OSDP 인식 알고리즘, 특징 벡터 PLP에 대하여 간략히 설명하고 3장에서는 인식 후 명령어의 실행을 위해 사용된 APIs Functions에 대해 설명하고 마지막으로 4장과 5장에서는 실험 결과 및 결론을 맺는다.

Ⅱ. Window 95환경 하에서의 인식 시스템

2.1 전체 시스템 개요

전체 시스템은 다음의 그림1과 같은 구성으로 동작하게 된다. 음성은 음성 구간 검출 및 음성 끝점 검출 과정을 거쳐 PLP 13차 특징 벡터가 추출되고 20 Section DMS model[3]을 reference로 하는 OSDP 인식 알고리즘을 통하여 인식된다. 인식된 결과는 API(Application Programming Interface)[4][5] 함수들로 이루어진 명령 실행 루틴 (Command Execution Routine)으로 전달되고 전달된 결과에 따라 실제 윈도우 operation이 이루어진다.

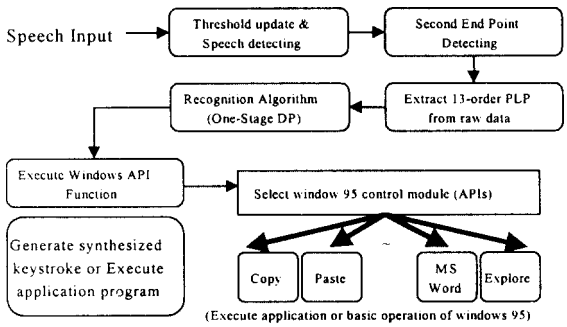

그림 1. 전체 시스템 Block Diagram

윈도우 95상에서 실제 명령을 구동시키기 위해 첫째, 윈도우 message 이용, 둘째, Application program을 수행시킬 때 해당 프로그램의 실행파일을 구동시키는 방법, 셋째, 해당 명령을 수행시키기 위한 Specific 한 routine을 생성하여 사용하는 방법을 이용하였다.

2.2 음성 데이터 베이스 구축

본 인식 시스템에서 사용한 음성 DB는 조용한 실험실 환경에서 Notebook 내장 마이크를 이용하여 20대 남성 화자 9명이 2회 발성한 데이터를 사용하였고 Off-line, On-line 실험에는 모델 생성에 참여하지 않은 20대 화자에 의해 수행되었다.

표 1. 입력 음성 설정 사항

설정 내용	값
Sample per Sec	11.025 (KHz)
Channel 수	Mono (Channel 1)
Bit per Sample	16 bits

2.3 Perceptual Linear Predictive (PLP)

음성 신호의 인지선형예측 (PLP) 분석은 이산 푸리에 변환(DFT)과 선형예측(Linear Prediction) 기술이 조합된 음성분석 방법이다. 이러한 인지 선형 예측 기술을 이용한 음성의 인지 선형 예측 계수를 구하는 방법은 아래 그림 2와 같다.

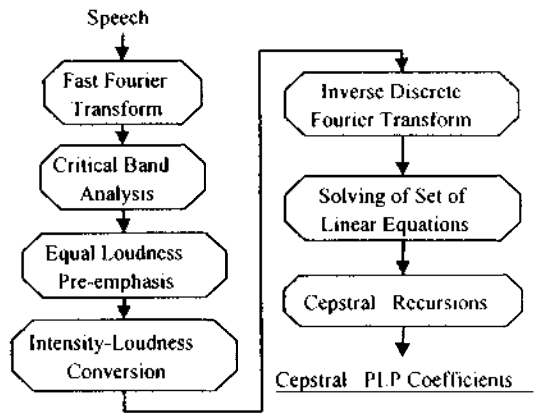

그림2. 음성의 인지선형예측 분석

이 인지선형예측 기술은 인간의 청각 스펙트럼을 모사(Imitation)하고, 음성 정보를 압축하는 효과를 보임으로 인하여 화자 독립 음성 인식에 적합한 음성 Feature이다. 본 논문에서는 13차의 Cepstral PLP coefficient를 음성의 feature로 사용하고 있다.

2.4 DMS(Dynamic Multi Section) modeling Procedure

이 모델은 유사한 특징을 가지는 벡터들을 한 구간으로 만들기 위해 구간을 동적으로 분할하여 대표 특징 벡터를 구함으로써 짧게 발음되는 특성까지도 대표 특징 벡터로 선택될 수 있도록 한 모델이다. 본 논문에서는 모델을 20개의 Section으로 나누어 실험을 수행하였다. 모델 생성에 사용되는 구간 구분화를 위한 DP알고리즘은 누적거리 D를 계산하여 사용한다. 즉

$$D(i,j) = d_s(t_i, m_j) + \min \begin{cases} D(i-1,j) \\ D(i-1,j-1) \end{cases} \quad (1 \le I, 1 \le j \le J) \qquad (1)$$

식에서 i 는 학습 데이터의 프레임 번호이고 j는 단어

모델의 구간번호, d_i는 국부적 거리, $D(i,j)$는 누적 거리를 나타낸다 위 식을 이용하여 학습용 데이터와 단어 모델사이의 거리를 구하고 그 값이 가장 작으면 모델에 등록한다.

그림 3은 "내컴퓨터"란 단어에 대한 OSDP 알고리듬의 예이다. 본 논문에서는 One-Stage DP방법을 단어 인식에 적용하여 사용하였다.

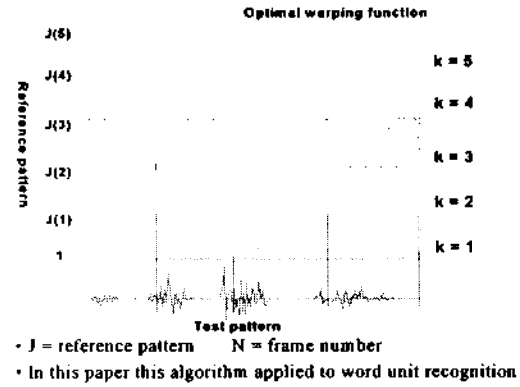

그림 3. OSDP 알고리듬의 예

2.5 Real-Time 음성 구간 검출

본 논문에서는 실시간 처리를 위하여 그림 4와 같은 알고리듬으로 구현되었다. 버퍼의 크기는 50ms이며 음성 입력을 위해 총 4개의 버퍼를 순환하여 사용하도록 하였다. 또한 실시간 음성 검출을 위한 작업 수행 중 데이터의 손실을 최소화하기 위해서 Threshold 설정을 위한 Energy 계산 시 다음 식 (2)와 같이 한 프레임 128 Sample 계산을 위해 w=5, w=3, w=1의 실험으로 데모 notebook 시스템에서는 w=5(실험값)의 설정을 사용한다.

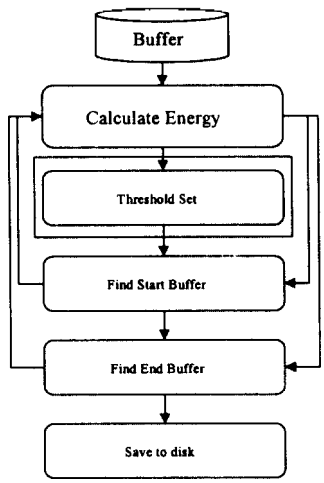

그림 4. 음성 구간 검출

$$E = \sum_{n=1}^{\frac{128}{w}} x(w \times n)^2 \qquad w = 5 \ (1\ frame) \qquad (2)$$

Ⅲ. Window 95 명령 실행 루틴

3.1 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 2. 입력 음성 설정 사항

구분	windows 95 명 령 어				
키보드 메세지	저장	예	컴퓨터 파일과 특별한 모듈	휴지통 바우기	휴지통 열기
	자우기	인쇄		네트워크 환경	끄
	아래로	아니오		단축메뉴	이름순으로 정렬
	불어넣기	엔터		종류순으로 정렬	크기순으로 정렬
	등록정보	취소		날짜순으로 정렬	자동순으로 정렬
	다른 이름으로 저장	창닫기	Netscape 제어	인터넷	다음사이트
	복사	탭		이전사이트	홈
	확로	확대		중지	북마크
	새파일	축소		담장보내기	메일보내기
	우로	열기		메일확인	주소록 보기
	다음창	위로		넷 검색	
	이전창	시작			
	닫기				
실행 파일	내 컴퓨터	에이(A) 드라이브		이전페이지	파일관리자
	프로그램	비(B) 드라이브		전화걸기	달력기
	나들이	시아(C) 드라이브		도와줘	다음 페이지
	잘라내기				

3.2 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

동적 메뉴 링크에 대한 구현은 다음의 API함수들을 이용하여 구현 하였다.

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

Ⅳ. 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 생성되는 메뉴를 출력하는 창이 존재한다.

그림 5. 시스템 메인 화면

위 그림 5는 인식 시스템을 처음 수행시킨 화면이다.

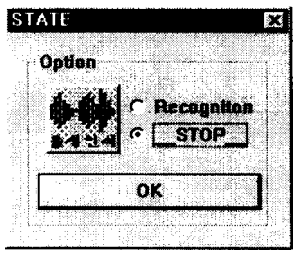

그림 6. 인식 상태 설정을 위한 상태창

그림 6은 인식 상태를 설정하기 위한 상태창을 나타낸다. Recognition버튼을 설정하면 인식 가능상태가 된다.

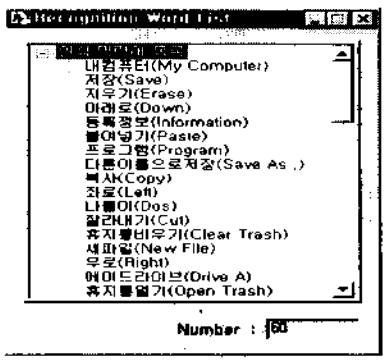

그림8. 사용자를 위한 명령 리스트 출력 창

Ⅴ. 실험 및 결론

시스템의 인식 실험에서 특징 벡터는 PLP 13차 계수를 사용하였고 화자 독립 온라인 실험과 화자 독립 오프라인 실험으로 나누어 실시하였다. 오프라인 실험인 경우 20대 화자 5인이 notebook 내장 마이크를 이용하여 3회 발성한 데이터를 이용하여 실험하였고 온라인 실험의 경우 또한 내장 마이크를 사용하여 20대 화자 10인이 2회 발성하여 60개의 윈도우 제어 명령을 실제 구동시키며 인식 실험을 하였다.

표3. 화자 독립 Off-line 인식 실험 결과

Speaker /Utterance	1st	2nd	3rd	AVE
A	100.0%	98.3%	100.0%	99.4%
B	98.3%	98.3%	98.3%	98.3%
C	95.0%	96.3%	96.6%	96.0%
D	98.3%	96.6%	98.3%	97.7%
E	100.0%	83.3%	88.3%	90.5%
AVE	98.3%	94.9%	96.3%	96.4%

표4. 화자 독립 On-line 인식 실험 결과

Speaker /Utterance	1st	2nd	AVE
A	98.3%	95.0%	96.65%
B	93.3%	91.6%	92.45%
C	96.6%	93.3%	94.95%
D	91.6%	88.3%	89.95%
E	90.0%	93.3%	91.65%
F	88.3%	91.6%	89.95%
G	95.0%	91.6%	93.30%
H	93.3%	85.0%	89.15%
I	95.0%	90.0%	92.50%
J	85.0%	88.3%	86.65%
AVE	92.64%	90.8%	91.72%

인식 결과는 위의 표3과 표4에서와 같다. 오프라인 화자 독립 실험의 경우 평균 96.4%의 인식률을 얻었고, 온라인 화자 독립 실험의 경우 평균 91.71%의 인식률을 얻었다. 이와 같은 온라인과 오프라인과 차이의 원인은 실시간 수행 중 주변 잡음의 변화와 명령 수행으로 인한 notebook시스템 내부의 잡음 증가 등이 주요 원인 이었다.

참고 문헌

[1] Hermann Ney, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition", IEEE Transaction on Acoustics, Speech, and Signal Processing, Vol. ASSP-32, No.2, pp263 ~ 271 April, 1984

[2] H. Hermansky, "Perceptual Linear Predictive(PLP) Analysis of Speech" J. Acoust. Soc. Am. 87(4), pp1738 ~ 1752, April 1990

[3] 변용규, "DMS 모델을 이용한 단독어 인식에 관한 연구" 박사학위 논문, 광운대학교 대학원, 1991.2.

[4] Richard simon,"Windows 95 WIN32 Programming API Bible", Waite Group Press

[5] "Microsoft Visual C++ MFC Library Reference, Part1, Part 2", Microsoft Press, 1997