PC를 이용한 실시간 음성검출 알고리즘에 관한 연구

정훈, 정권, 정익주
강원대학교 전자공학과

A Study on the Development of Real Time Speech Detection in PC

Hoon Chung , Kwon Chung and Ikjoo Chung
Dept. of Electronics, Kangwon National University

요약

본 논문에서는 윈도우즈용 음성인식 Software "Voice Access" 를 개발하며 연구한 실시간 음성검출 알고리즘에 관해 소개한다.

이 음성검출 알고리즘은 200 sample 단위의 프레임 에너지, 프레임 영교차율, 음성의 길이를 음성검출의 파라메타로 사용한다. 각 파라메타의 문턱값은 신호의 평균값, 잡음의 표준편차, 미디안 표준편차와 한국어의 음성적 특성을 고려하여 설정하였으며 주변의 환경에 적응해 가며 문턱값을 조정하므로 주변 잡음환경의 변화에 대해서도 강인한 음성검출 결과를 보여준다. 또한 실시간으로 음성을 검출하므로 실용성이 높다. 음성의 검출은 일반사운드 카드를 통해 16-bit의 8KHz로 샘플링된 신호를 사용한다. 음성검출을 위한 분석은 200 sample씩 하고 100 sample씩 overlap하여서 수행한다. 음성 검출을 위한 모든분석은 특별한 DSP의 도움없이 486DX이상에서 실시간으로 구현했다.

1. 서론

잡음환경하에서 고립단어음성인식 시스템의 경우 오인식은 주로 부정확한 음성의 검출에서 기인한다. 최근 고립단어 인식시스템의 평가에서 인식에러의 50%이상이 부정확한 음성검출에 의해 발생한다고 밝히고있다. 특히 비교패턴 간의 끝점 정보를 이용하여 시간보상을 하는 DTW 기반의 음성인식 시스템의 경우에는 HMM이나 ANN기반의 인식시스템보다 음성검출 부분이 전체인식률및 인식속도에 미치는 영향이 크다. 따라서 DTW기반의 음성인식시스템의 경우에는 매우 정확한 음성검출이 요구된다. 일반적으로 음성의 검출은 잡음이있는 환경에서 수행되므로 정확한 음성검출을 위해서는 주변 잡음환경을 잘 모델링하여야 하며, 효과적인 음성검출 파라메타와 Decision Rule를 가지고 있어야한다. 그리고, 시간에 따라 변화하는 주변 잡음환경에 적응할 수 있도록 설계되어야한다. 또한, 실용적인 음성인식 시스템을 위해서는 실시간의 음성검출이 불가피하다. 이미 잘알려진 Rabiner & Sambur 의 끝점검출 알고리즘을 포함한 대다수의 끝점검출 알고리즘은 Off-line처리로, 어느 구간내에 음성이 포함되었다는 가정하에서 음성의 끝점을 검출해낸다. 어떤경우에는 두번 이상의 구간탐색을 하여 음성을 검출함으로서 실시간 음성검출에는 적합하지 않다.

본 논문에서는 위에서 제기된 잡음적응과 실시간 처리 문제에 대해 신뢰성있는 음성검출 결과를 보여주는 알고리즘을 소개한다. 본 음성검출 알고리즘에서는 프레임 에너지, 프레임 영교차율, 음성의 길이를 음성과 잡음구분의 파라메타로 설정했다. 각 파라메타의 문턱값은 주변잡음환경을 채취하여 설정하였으며, 한국어의 음성적 특성을 고려하였다. 잡음으로 판명된 프레임으로 부터 음성판별 파라메타의 문턱값을 조정하여 주변잡음에 적응하게 한다. 또한 Double-buffering 기법을 이용하여 입력 sample의 손실없이 실시간으로 음성검출을 구현하였다. 본 논문에서 구현한 실시간 음성검출 알고리즘은 일반 사운드 카드가 설치된 윈도우즈 환경하에서 전 과정을 486DX만을 이용하여 구현하였다.

2. 입력 부분

음성의 입력은 윈도우즈 환경하에서 일반적인 사운드 카드를 이용하여 이루어 진다. 윈도우즈는 MCI(Media Control Interface)를 이용한 High-Level의 오디오 서비스와 Low-Level의 오디오 서비스를 제공한다. MCI를 이용한 오디오 서비스는 사운드 카드의 미세한 부분까지 제어하기에 어려움이 있으므로 Low-Level의 오디오 서비스를 이용하여 음성을 입력받는다. 응용프로그램과 사운드 카드의 데이타 교환은 그림 1과 같은 구조를 가진다.

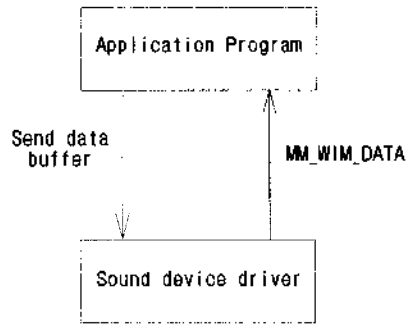

그림 1 응용 프로그램과 사운드간의 데이타 교환 방식

응용프로그램은 초기에 A/D변환된 결과를 저장할 버퍼를 사운드 드라이버에 넘겨준다. A/D변환이 시작되고 지정된 버퍼가 다 차거나 사운드카드를 초기화하면 응용프로그램에서 사운드 드라이버에 넘겨준 버퍼가 응용프로그램에 MM_WIM_DATA 메시지의 파라메타로 넘어온다. 응용프로그램은 전달된 버퍼의 내용을 분석하고 다시 사운드 드라이버 다음 A/D을 위해서 넘겨준다. 음성검출의 입력 데이타는 16bit resolution을 가지며 8KHz로 Sampling되었다.

3. 실시간 음성구간 검출

3.1 사전조정(Caliabration)

일반 사운드카드의 경우 연구장비와 달리 200Hz ~ 4KHz의 Bandpass filter를 사용할 수 없고 단지 Lowpass filter만을 이용할 수 있으므로(Sigma-Delta Modulation 방식 사용하는 경우 Bandpass Filter 사용가능) 사전조정에서 DC Bias를 제거하고 주변의 잡음정도를 알아내어 음성검출의 기본 문턱값들을 결정한다. 약 5초정도의 주변잡음을 채취하여 200 sample을 한 프레임으로 하고 100 sample씩 overlap하여 각 프레임의 평균 m_i, 표준편차 s_i를 구한다. 여기서 i 는 프레임 index이다. m_i와 s_i 로 부터 다음 파라메타들을 구한다.

Averaged frame mean : $M = \frac{1}{N}\sum_{i=0}^{N-1} m_i$ (1)

Averaged frame standard deviation : $ASD = \frac{1}{N}\sum_{i=0}^{N-1} s_i$ (2)

Median frame standard deviation :

$$MSD = median\{s_0, s_1, \ldots, s_{N-1}\} \quad (3)$$

where N is the total number of frames.

여기서 M은 사운드카드의 DC Bias를 제거하는데 사용되며 ASD와 MSD는 영교차율의 문턱값과 관련되어 사용되어진다. ASD를 이용하여 다음과 같은 프레임 영교차율 zcr_i을 구한다.

$$zcr_i = \sum_{k=0}^{199} count(x_{i,k}, x_{i,k+1}) \quad (4\text{-A})$$

여기서 $count(x_{i,k}, x_{i,k+1})$ 는

$$\begin{aligned} &x_{i,k} - M < 2ASD < x_{i,k+1} - M \text{ 또는} \\ &x_{i,k+1} - M < 2ASD < x_{i,k} - M \text{ 또는} \\ &x_{i,k} - M < -2ASD < x_{i,k+1} - M \text{ 또는} \\ &x_{i,k+1} - M < -2ASD < x_{i,k} - M \text{ 이면} \end{aligned} \quad (4\text{-B})$$

1 이고 그렇지 않으면 0 이다.

프레임 영교차율 zcr로 부터 다음과 같은 파라메타를 구한다.

Average frame zcr : $AZCR = \frac{1}{N}\sum_{i=0}^{N-1} zcr_i$ (5)

Median frame zcr :

$$MZCR = median\{zcr_0, zcr_1, \ldots, zcr_{N-1}\} \quad (6)$$

최종적으로 사전조정을 통하여 구한 잡음의 표준편차와 영교차율은 다음과 같이 정한다.

$$SD = 0.3 * ASD + 0.7 * MSD \quad (7\text{-A})$$
$$ZCR = 0.3 * AZCR + 0.7 * MZCR \quad (7\text{-B})$$

3.2 실시간 음성구간 검출

사전조정에서 구한 SD을 바탕으로 200 sample(100 sample overlap) 크기의 프레임에 대한 에너지 관련 문턱값을 정한다.

$$\begin{aligned} VoiceStartEnergy &= 13 * SD^2 \\ VoiceEndEnergy &= 30 * SD^2 \\ VowelEnergy &= 300 * SD^2 \\ VoiceRestartEnergy &= 40 * SD^2 \end{aligned} \quad (8)$$

$VoiceStartEnergy$는 시작점을, $VoiceEndEnergy$는 끝점 검출에 이용된다. $VoiceRestartEnergy$는 끝점검출 후 재개되는 시작점 검출에 이용된다. 각 프레임의 에너지는 다음과

같이 구한다.

$$energy_i = \sum_{k=0}^{99}(x_{i,k} - M)^2 + last\_energy_{i-1}$$
$$last\_energy_i = \sum_{k=0}^{99}(x_{i,k} - M)^2 \quad (9)$$

영교차율은 식()과 유사하게 구하나 ASD 대신 SD를 사용한다.

$$zcr_i = \sum_{k=0}^{99} count(x_{i,k}, x_{i,k+1}) + last\_zcr_{i-1}$$
$$last\_zcr_i = \sum_{k=0}^{99} count(x_{i,k}, x_{i,k+1}) \quad (10)$$

영교차율의 문턱값은 다음과 같다.

$$UnvoicedConsonantZCR = 3 * ZCR$$
$$FricativeZCR = 4.4 * ZCR \quad (11)$$

위의 문턱값들과 에너지 및 영교차율의 정의를 이용하여 그 프레임(200 sample)이 음성으로 판명되면 아래 그림과 같이 중간의 100 sample을 저장한다.

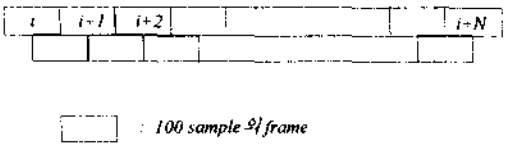

그림 2. 실시간 음성구간 검출

실시간으로 이루어져야 하므로 모든 것이 프레임단위로 적용이되어진다. 우선 초기 6 프레임이 연속해서 음성프레임이라고 판정이 되면 일단 음성이 시작되었다고 가정하고 그 6프레임을 포함해서 이후에 들어오는 프레임들을 버퍼에 저장하고 이어 동시에 끝점검출에 들어간다. 일단 한 프레임이라도 끝점이라고 판정이 되면 이를 끝점으로 간주하고 그 이후 20 프레임을 일단 버퍼에 저장함과 동시에 이 20프레임 중 연속하여 5 프레임이 음성으로 판정되지 않으면 음성검출을 완료한다. 만약 음성이 재개되면 이들을 계속해서 저장하고 끝점검출을 반복한다. 이와는 별도로 시작점과 끝점 부근에서 주로 발생하는 저주파잡음인 숨결잡음(breath noise) 제거 루틴이 추가가 된다.(사운드 카드에는 저주파를 제거하는 고역필터가 없다) 끝점검출이 끝나면 사후처리를 거친다. 사후처리에서는 검출된 총 프레임수가 10 이하이거나, 또는 전체프레임 중 $VowelEnergy$ 보다 큰 프레임의 수가 4 이하이거나, 프레임 전체의 평균 영교차율이 13 이하이거나 (저주파잡음의 경우)이면 이를 잡음으로 간주하고 다시 음성검출에 들어간다.

주변의 잡음의 변화에 적응하기 위하여 음성구간 검출과정에서 그 프레임이 음성이 아니라고 판정이 되면 다음과 같이 SD를 적용 시키다.

$$SD = 0.95 * SD + 0.05 * s_i \quad (12)$$

여기서 s_i는 그 프레임의 표준편차이다. 이를 통하여 에너지 문턱값과 영교차율이 변화된 주변잡음에 적응하도록 한다. 그림 3, 4는 음성구간이 검출된 "쉬프트탭"과 "스페이스바"의 음성파형이다.

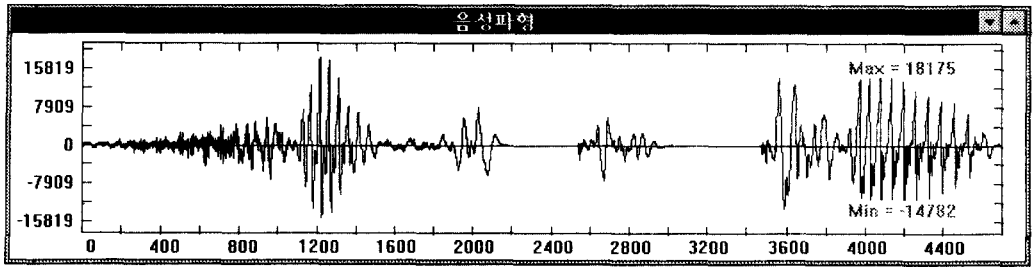

그림 3. 음성구간 검출된 "쉬프트탭"

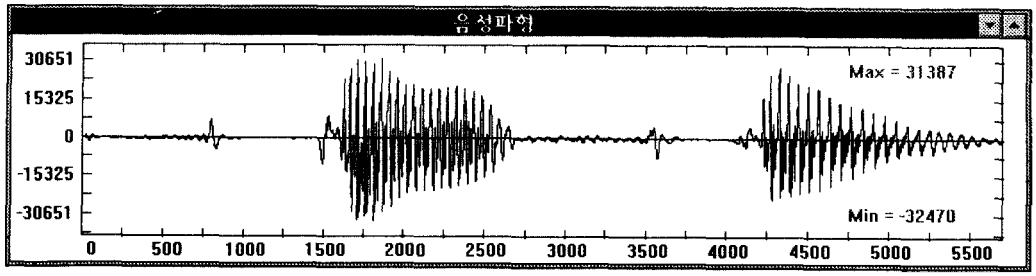

그림 4. 음성구간 검출된 "스페이스바"

4. 결론

본 논문에서는 윈도우즈 환경에서 사용할 수 있는 음성인식 소프트웨어 개발과 관련된 실시간의 음성검출 알고리즘을 제시하였다. 이 알고리즘은 잡음환경에 적용하면서 음성을 검출하므로 상당히 신뢰성 있는 결과를 보여주며, 486DX PC에서 실시간으로 구현하여 매우 실용적인 면을 보이고 있다. 좀더 정확한 음성의 검출을 위해서는 주파수 정보를 음성검출에 이용하고, 음성검출 파라메타 문턱값의 설정시 충분한 음성시료의 통계적 특성으로 부터 구하고, 주변잡음에 적용하는 음성검출 알고리즘이 요구된다.

참고문헌

[1] L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, " Considerations in Dynamic Time Warping algorithms for Discrete Word Recognition,"IEEE, Trans. Acoust., Speech, Signal Processing, vol ASSP-26, No. 6, pp 575-582, 1978

[2] B. S. Atal and L. R. Rabiner, "Pattern recognition approch to Voiced-Unvoice-Silence Classificatin with Applications to speech Recognition," IEEE, Trans Acoust., Speech, Signal Processing, vol. ASSP-24, No. 3, pp 201-211, 1976

[3] Developer Kit Progammer's Guide for Sound Blaster Series, Creative Lab.

[4] Microsoft Windows 3.1 Guide to Programming, Microsoft Press, 1992

[5] Microsoft Windows 3.1 Programmer's Reference, Microsoft Press, 1992