# 수화자(受話者) 구별을 위한 PAMD 구현.

서 봉수(徐 鳳隨)
서울통신기술 정보기술그룹
전화 (02)2225-6056 ,팩스 (02)2226-6069
H.P : 016-777-5640

# Implement PAMD for discriminate human and ARS

Bong Su Seo
Information Engineering Group Seoul Commtech Co.Ltd
E-mail: miroo.seo@samsung.com

Abstract

In this Paper, we implement PAMD(Positive Answering Machine Detection) for discrimina-tion human and ARS. We are used Grunt detection, Glitch Noise detection and Tone detection for PAMD. It distinguishes voice signals from ring-back tone and glitch noise respectively. And as a second step, it judges whether human responses or ARS responses after integrating pattern changes like initial response period, the number of voice data, each time of voice data period and glitch noise. The accuracy is about 93% in ASR and about 98% in Mobile phone.

## 1. 서론.

사람이 일반적으로 전화를 받을 때는 "여보세요" 또는 "OOO입니다"라고 말을 한다. 그러나 자동 응답기나 셀룰러나 PCS의 음성 사서함의 안내 멘트는 이 보다는 훨씬 길며 상대방이 말하는 시간을 기다려 주지 않고 자신의 말만을 계속한다. 본 논문에서는 이러한 차이를 이용하여 전화를 받는 상대방이 사람인지 자동 응답기인가를 판단할 수 있는 방법을 구현 하였다.

PAMD를 구현함에 있어서 Grunt Detection, Tone Detection, Glitch Noise Detection등의 방법을 이용하였다.

## 2. Grunt Detection 및 톤 검출.

### 2.1 Grunt Detection

Grunt Detector 는 4 msec (32 sample 구간) 마다 입력 신호의 frame energy 를 구하고, leaky-integrator 로 smoothing 한 결과를 threshold 값과 비교하여 입력 신호의 유무를 판단하는 것이다. 입력 신호의 frame energy 는 4 msec 마다 구하였다.

다음은 frame energy 를 smoothing 하는 leaky-integrator 인데, frame energy 를 입력으로 하고 출력은 smoothing 된 energy 이다. Leak-integrator 는 1 차 infinite impulse response (IIR) filter 인데, average 효과를 주는 filter 로 filter 의 계수를 조정하면 average 구간을 조절할 수 있다. 사용 되는 leaky-integrator 의 식은 다음과 같고, 역시 4 msec 마다 한 번씩 계산하게 된다,

$$H(z) = \frac{\frac{1}{2^p}}{1 - (1 - \frac{1}{2^p}) z^{-1}} \qquad (2\text{-}1)$$

위 식 (2-1) 에서 $2^p$ 의 $p$ 는 leak-integrator 의

average 구간을 결정하는데, $p$ 가 크면 long-term average 가 되고, 반대로 $p$ 가 작으면 short-term average 가 된다. 위에서 설명한 대로 frame energy 를 식 (2-1) 에 주어진 filter 의 입력으로 하여 average 를 구하면 다음과 같은 식이 얻어지고,

$$\bar{E}(m) = (1-\frac{1}{2^p})\bar{E}(m-1)+\frac{1}{2^p}E(m) \qquad (2\text{-}2)$$
$$= \bar{E}(m-1)+\frac{1}{2^p}\{E(m)-\bar{E}(m-1)\}$$

식 (2-2) 에서 구한 $\bar{E}(m)$ 은 smoothing 된 energy 를 나타낸다.

Grunt Detection 은 식 (2-2) 에서 구한 average energy $\bar{E}(m)$ 을 threshold 와 비교하여 음성신호와 배경잡음을 구별하는 것인데, threshold 값을 결정하는 것이 성능에 중요한 영향을 준다. Grunt Detection 의 threshold 값은 신호에 따라 달라질 수 있으므로 고정된 값으로 할 수 는 없고, 상황에 따라 가변 할 수 있도록 한다. 그리고 현재 frame energy 값이 threshold 보다 크면 현재 frame 을 음성신호로 결정하고, threshold 보다 작으면 배경잡음으로 결정한다.

Threshold 값을 정하는 것은 frame energy tracking 에 기반을 두는데, 다음 식으로 구한다.

$$E_{th}(m) = 1.5E_d(m) \qquad (2\text{-}3)$$

식 (4) 에서 m 은 m 번째 frame index 를 나타내고, $E_{th}(m)$ 은 threshold 값이며, $E_d(m)$ 은 배경 잡음의 average frame energy 를 나타낸다. 그리고, $E_d(m)$ 은 frame energy 가 $E_{th}(m)$ 보다 작을 때 (배경잡음 frame) 마다 식 (2) 의 leaky-integrator 를 사용하여 구한다.

여기에서 구한 $E_d(m)$ 은 결국 배경잡음의 average frame energy 가 되고, $E_d(m)$ 은 다음 식으로 계산된다.

$$E_d(m) = (1-\frac{1}{2^p})E_d(m-1)+\frac{1}{2^p}E(m)$$
$$= E_d(m-1)+\frac{1}{2^p}\{E(m)-E_d(m-1)\}$$
$$when\ E(m) < E_{th}(m) \qquad (2\text{-}4)$$

## 2.2 Tone 검출 알고리즘.

Tone 신호를 검출 하는데는 Goertzel-Algorithm을 사용 하였다[1]. Tone 검출을 하려면 주파수 *domain* 에서 spectrum을 구하여 원하는 주파수 성분이 있는지 확인하는 과정이 필요한데, 이를 위하여 Discrete Fourier Transform (DFT)이 흔히 사용된다. 다음은 N 개의 디지탈 신호 $x(m)$ 에 대한 DFT 계산식인데, 기본 DFT 계산식을 단계적으로 변형하면 다음과 같이 정렬할 수 있고, $X_k$ 는 $x(m)$ 의 k번째 주파수 ($\omega = \frac{2\pi}{N}k$) 성분을 의미한다.

$$X_k = \sum_{m=0}^{N-1} x(m)e^{-j\frac{2\pi}{N}km} = \sum_{m=0}^{N-1} x(m)e^{j\frac{2\pi}{N}kN}e^{-j\frac{2\pi}{N}km} \qquad (2\text{-}5)$$
$$= \sum_{m=0}^{N-1} x(m)e^{j\frac{2\pi}{N}k(n-N)}, k = 0,1,\cdots,N-1$$

여기에서 $y_k(n)$ 을 다음과 같이 정의하면

$$y_k(n) = \sum_{m=0}^{N-1} x(m)e^{j\frac{2\pi}{N}k(n-m)} \qquad (2\text{-}6)$$

(2-5)과 (2-6) 식에서 다음의 결과를 얻는다.

$$X_k = y_k(n)\big|_{n=N} \qquad (2\text{-}7)$$

따라서 식 (2-5)을 이용하여 $X_k$ 를 계산하는 대신에, 식 (2-6)를 이용하여 $y_k(n)$ 을 반복 계산하고 마지막에 $y_k(N)$ 값을 구하여도 같은 결과를 얻는다.

여기에서 식 (2-7) 은 $x(n)$ 과 one-pole resonator 의 impulse response와 convolution 이므로, 식 (2-7) 에 z-transformation 을 하면 다음 결과를 얻고,

$$Y_k(z) = X(z)\frac{1}{1-e^{-j\frac{2\pi}{N}k}z^{-1}} \qquad (2\text{-}8)$$

여기에 다시 *inverse z-transformation* 을 취하면 다음의 time difference 식이 얻어진다.

$$y_k(n) = e^{-j\frac{2\pi}{N}k}y(n-1)+x(n) \qquad (2\text{-}9)$$

식 (2-9)는 one-pole filter를 사용하기 때문에 complex 곱셈이 필요한데, 이것은 DSP 에서 구현하

기에 적합하지 않다. 따라서 one-pole filter 대신에 complex conjugate pole $p_{1,2}=e^{\pm j\frac{2\pi}{N}k}$ 을 갖는 two-pole filter를 사용하면 다음과 같이 real 곱셈만 사용하는 time difference 식이 얻어지고,

$$v_k(n)=2\cos(\frac{2\pi}{N}k)\cdot v_k(n-1)-v_k(n-2)+x(n) \quad (2\text{-}10)$$

$y_k(n)$과 $v_k(n)$ 사이에 다음의 관계가 있다.

$$y_k(n)=v_k(n)-e^{-j\frac{2\pi}{N}k}\cdot v_k(n-1) \quad (2\text{-}11)$$

따라서 우리가 구하려는 k번째 DFT, $X_k$ 는 식 (2-10)과 (2-11)을 이용하여 구하는데, 식 (2-10) 을 n=0,…,N 에 대하여 반복 계산하고 마지막 n=N에서 계산한 값이 다음에 보는 바와 같이 우리가 원하는 k번째 DFT 이다.

$$X_k=y_k(N)=v_k(N)-e^{-j\frac{2\pi}{N}k}\cdot v_k(N-1) \quad (2\text{-}12)$$

그런데 Tone Detection의 경우 DFT의 phase 정보는 필요하지 않고 magnitude 정보만 사용하므로, 식 (13) 으로부터 다음의 결과를 얻는다.

$$|X_k|^2=y_k(N)\cdot y_k^*(N) \quad (2\text{-}13)$$
$$=v_k^2(N)+v_k^2(N-1)-2\cos(\frac{2\pi}{N}k)\cdot v_k^2(N)\cdot v_k^2(N-1)$$

### 2.3 Glitch Noise Detection

위의 2.1 에서 설명한 Grunt Detection 으로 음성신호와 배경잡음을 구별하지만, 음성신호라고 판별된 frame 은 경우에 따라 2.2 에서 설명한 대로 ring-back tone 일 수도 있고 혹은 glitch noise 일 수도 있다. glitch noise 는 배경잡음과는 달리 짧은 시간에 큰 크기를 갖는 impulse 성 noise 성분으로 음성신호와 구별되어야 한다.

다음 [그림 1] 은 ARS 에 접속되어 녹음 된 신호를 보여 주는데, 처음 에는 silence 이다가 glitch noise 를 시작으로 음성 신호가 녹음 되어 있다. Grunt Detection 에서는 frame energy 만으로 신호의 유,무를 판별하므로 glitch noise 를 찾기가 어렵다.

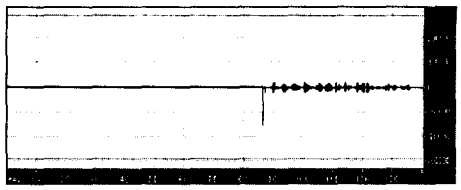

그림 1 Glitch Noise 가 있는 음성 data

Glitch Noise 와 음성 신호를 구별하기 위한 차이점을 찾기 위하여 Glitch Noise 부분을 확대해 보면 다음 [그림 2] 와 같다.

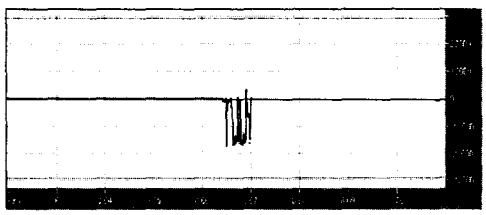

그림 2 Glitch Noise 부분 확대

확대 된 glitch noise 부분을 보면 음성신호와는 다른 특징이 있는데, 일반적으로 음성신호는 positive 와 negative 부분이 균형을 이루는데 glitch noise 는 한 쪽으로 (여기에서는 negative 방향) 치우쳐 있다. 따라서 한 frame 안에서 positive sample 개수와 negative sample 개수의 차이를 구하고, 이것이 threshold 값보다 크면 glitch noise 라고 판단한다.

## 3. PAMD의 구현

PAMD의 구현은 입력된 신호의 frame 을 위에서 설명한 Grunt Detection, Tone Detection, Glitch Noise Detection 을 적용하여 각각 음성신호, ring-back tone, glitch noise, 배경잡음 등으로 구별하고, 그 다음 응답 감지 후 초기 응답 시간, 음성 data 구간의 수, 각각의 음성 data 구간의 시간, glitch noise, 등의 pattern 변화를 종합하여 사람이 응답한 것인지, 아니면 ARS가

응답 한 것인지 판별하게 된다

## 4.실험 및 결론

본 논문에서는 PAMD 검출 데이터로 이동 전화와 기업체의ARS를 대상으로 CT-V16B 보드를 이용하여 발신한 후 PAMD 를 수행하였다.

첫번째 실험은 이동전화를 대상으로 하였다. 이동전화는 통화중, 전화기가 꺼져있는 경우, 전화를 받는 않는 경우 그리고 사람이 받는 경우에 대해서 시험을 하였다. 대상은 20명의 전화에 대해서 5번을 발신하였다. 이경우 Color Ring을 적용한 전화는 제외하였다. 사서함이 받은 경우가 60건 사람이 받은 경우가 40건으로 나누어 실험 하였다.

시험 결과 이동 전화의 사서함 멘트에 대해서는 100%의 판별력을 보였고 사람이 받은 경우에는 95%. 의 판별력을 보였다.

두번째는 각 기업의 고객지원센터의 ARS 시스템을 대상으로 하였다.

각 기업들은 고객을 대상으로 ARS시스템을 운영중이며 보통의 경우 상담원이 받는 경우 보다는 녹음된 멘트를 들려주고 고객이 원하는 경우에만 상담원과 연결이 되도록 되어있다. 이러한 고객센터 50곳의 ARS를 대상으로 5번 실험한 결과 약 93% ARS로 판별하였다.

오류는 ARS시스템에서 안내 멘트가 짧게 나오는 경우와 Ring Back Tone이후에 연결시 다른 톤이 발생하여 이를 사람으로 인식하여 발생하였다.

이와 같이 이동전화와 ARS에 대해서 PAMD 실험을 한 결과 이동전화에 대해서는 약 98%, ARS에 대해서는 93%의 결과를 보였다.

참고 문헌.

1. DTMF Tone Generation and Detection on the TMS320C54x, Texas Instrument, May 2000