후처리를 이용한 환경음 인식 성능 개선

Improvement of Environmental Sounds Recognition by Post Processing

박준규, 백성준
전남대학교 전자컴퓨터공학부

Jun-Qyu Park(junq14@naver.com), Seong-Joon Baek(tozero@chonnam.ac.kr)

요약

본 연구에 사용된 환경음은 9 가지 상황으로 구분하였으며 생활 속에서 인간의 이동에 따라 변화하는 실제 환경음과 동일한 테스트 데이터 셋을 이용하였다. 실제 환경에서 녹음된 데이터는 Pre-emphasis, Hamming window를 이용하여 전처리하고 MFCC (Mel-Frequency Cepstral Coefficients) 방식으로 특징을 추출한 후 GMM (Gaussian Mixture Model)을 이용하여 분류 실험을 행했다. 후처리가 없는 GMM은 프레임 별로 판정하므로 분류 결과를 보면 상황이 갑자기 변화하는 이상 결과가 나타난다. 이에 본 연구에서는 인접한 프레임 별 확률 값 혹은 분류 순위를 이용해서 갑작스런 상황 변화가 발생하지 않도록 하는 후처리 방식을 제안하였다. 실험 결과에 따르면 GMM 분류방식에 인접 프레임들의 사후확률 값을 이용하는 후처리방법을 적용한 경우 후처리를 적용하지 않은 경우에 비해 10% 이상 평균 인식률이 개선되는 것을 확인할 수 있었다.

■ 중심어 : | 환경음 | 가우시안 믹스처 모델 | 후처리 |

Abstract

In this study, we prepared the real environmental sound data sets arising from people's movement comprising 9 different environment types. The environmental sounds are pre-processed with pre-emphasis and Hamming window, then go into the classification experiments with the extracted features using MFCC (Mel-Frequency Cepstral Coefficients). The GMM (Gaussian Mixture Model) classifier without post processing tends to yield abruptly changing classification results since it does not consider the results of the neighboring frames. Hence we proposed the post processing methods which suppress abruptly changing classification results by taking the probability or the rank of the neighboring frames into account. According to the experimental results, the method using the probability of neighboring frames improve the recognition performance by more than 10% when compared with the method without post processing.

■ keyword : | Environmental Sound | GMM | Post-processing |

I. 서 론

최근 정보기술산업의 발전으로 휴대전자기기의 기능은 수동적으로 정보를 받아 사용자에게 제공하는 수준을 넘어 기기에서 수집 가능한 정보와 사용자가 생성한 정보를 바탕으로 유용한 정보 제공이 가능한 형태로 진

* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (NIPA- 2010-C1090-1011-0008)
접수번호 : #100302-004
접수일자 : 2010년 03월 02일
심사완료일 : 2010년 07월 06일
교신저자 : 백성준, e-mail : tozero@chonnam.ac.kr

화하고 있다. 이러한 휴대전자기기의 발전에 따라 외부에서 입력받은 환경정보와 사용자가 생성한 내부 정보를 토대로 사용자의 환경을 인식하고 서비스를 제공하는 환경인식기능의 중요성이 커지고 있으며 이에 따른 연구가 활발히 진행되고 있다[1].

외부로부터 환경정보를 입력 받는 방법 중의 하나로 휴대전화의 마이크로폰을 통한 소리입력을 고려할 수 있는데 이 경우 부가적인 센서의 사용 없이 외부환경의 음향정보를 입력받을 수 있다는 장점이 있다. 휴대전화에서 환경인식 기능은 극장이나 강의실 같은 사용자의 위치 정보를 바탕으로 공공장소에서 자동으로 에티켓 모드로 전환하는 등의 기능을 제공할 수 있다. 또한 이 휴대전화에서 환경인식 기능은 가까운 미래 휴대폰시장의 트렌드가 될 스마트폰에 지원되는 위성항법장치의 정보와 결합하여 어린이나 노약자 또는 범죄용의자의 위치 정보를 이 전보다 더욱 자세히 제공할 수 있을 것이다. 이와 유사하게 환경음 인식기능에 의한 사용자의 위치 정보는 다른 정보와의 결합으로 새로운 서비스를 제공할 수 있게 할 것이다.

환경음 인식 분야에서는, 기존 음성 인식 분야의 대표적인 특징 추출 방식인 MFCC (Mel-Frequency Cepstral Coefficients)와 LPCC (Linear Prediction Cepstrum Coefficients)를 특징 추출에 주로 이용해 왔다[2-5]. 최근의 연구들에서는 이들을 기반으로 점차 발전된 형태의 특징 추출 기법들을 제안하고 있고, 여기에는 둘 이상의 특징 추출 방식을 조합하여 인식 성능을 개선시킨 연구들이 대부분을 이루고 있다[6-9].

환경음의 특성인 무작위성, 높은 분산성을 비롯한 여러 가지 문제들로 인해 환경음 인식 시스템에서는 환경의 class 수가 증가 할수록 그 인식 성능이 급격히 감소하거나[9], 고려하는 환경의 종류에 따라 인식 성능에 차이가 발생한다. 이러한 환경음 분류의 특성을 고려하여 대표적 기존연구들의 인식 성능을 살펴보면, 먼저 12차 MFCC 특징과 기타 시간, 주파수영역의 특징을 조합하여 이를 바탕으로 5개의 class를 분류한 연구에서 92%의 인식 성능을 보여주고 있다[6]. 또한 64차 MFCC 특징을 PCA(Principal component Analysis)를 이용하여 축소한 후 이를 이용하여 11개의 class를 분류한 작업에서는 77%의 성능을[7], 14개 class를 12차 MFCC특징을 사용하여 분류한 연구에서 70%의 인식 성능을 보여주고 있다[9].

하지만 프레임 별로 분류 판정을 하는 기존의 방식은, 여러 노이즈들이 결합되어 근본적으로 좋은 특징을 추출하기 힘든 환경음의 구조적 특성을 잘 반영하지 못하며, 환경을 인식하고자 하는 인식주체의 움직임 특성을 모델링하지 못하고 있다. 그 결과 기존의 연구결과에서는 유사한 환경음들 간의 분류오류를 발생시키는 것을 크게 개선하지 못하고 갑자기 환경이 변하는 것으로 이상 판정을 하는 현상이 나타나고 있다. 이에 본 연구에서는 현재의 프레임을 최종 판정하기 위해 인접한 프레임의 GMM 확률 값 혹은 분류 순위를 이용한 후처리방식을 제안하고 실험을 통해 제안한 후처리방식이 환경음 인식 성능을 개선시킬 수 있음을 보이고자 한다.

II. 실험 데이터 준비와 특징 추출 및 분류방법

1. 실험 데이터 준비

실험에서 사용한 데이터는 총 9가지 환경에서 휴대용 microphone을 이용하여 획득하였다. 각 환경은 서울의 지하철 7개호선(Subway), 고속철도(KTX), 버스(Urban Bus), 승용차(Vehicle), 극장(Theater), 식당(Restaurant), 강의실(Classroom), 실외 걷기(Outside Walking), 실외 뛰기(Outside Running)이고, 9가지 환경음 데이터 각각의 길이는 대략 60분 전후이고, 샘플링 주파수는 8kHz, 양자화비트는 16bits이다.

본 연구에서는 9가지 상황에서 발생한 환경음 데이터들을 각각 30초 길이로 준비하였고, 이를 3개의 조합으로 연결하여 약 1분 30초 길이로 다수의 테스트 데이터를 구성하도록 하였다. 이를 통해 실생활에서 인간의 이동에 따라 변화하는 실제 상황을 표현할 수 있도록 10가지 상황에 대한 테스트 데이터를 구성하였으며 이를 [표 1]에 나타내었다.

[표 1]의 각 테스트 데이터는 실제 인간이 휴대전화를 가지고 이동할 때 마이크로폰을 통해 입력되는 환경음과 유사하게 9개의 환경 데이터를 이용하여 최소한

의 중복으로 10가지 상황을 구성하여 테스트 셋으로 만든 것으로 분류 실험에 보다 현실적인 방법이라고 볼 수 있다.

표 1. 3 class로 조합된 테스트 데이터 셋

data set	구성
case 1	버스 – 걷기 – 식당
case 2	식당 – 걷기 – KTX
case 3	KTX – 걷기 – 극장
case 4	극장 – 걷기 – 자동차
case 5	수업 – 걷기 – 식당
case 6	극장 – 걷기 – 지하철
case 7	지하철 – 뛰기 – 버스
case 8	걷기 – 뛰기 – 걷기
case 9	자동차 – 뛰기 – 수업
case10	수업 – 뛰기 – 지하철

2. 특징 추출

전처리 과정은 Pre-emphasis, Windowing으로 구성된다. 일반적으로 환경음 신호는 20Hz-18kHz의 주파수를 가지며 300Hz를 최대로 하여 1kHz 이상부터 에너지 크기가 작아지는 현상을 보인다. 이러한 특성을 보상하기 위해 다음과 같은 고대역 통과 특성을 갖는 pre-emphasis 필터를 사용하였다.

$$H(z) = 1 - 0.97z^{-1} \quad (1)$$

Pre-emphasis 과정을 거친 신호는 일정한 길이의 프레임으로 나누는데 이때 각 프레임 사이의 정보 손실을 고려하여 N개의 샘플로 blocking하여 하나의 프레임으로 사용하고 다시 M개의 샘플만큼 이동하여 중첩을 시킨 후, 다시 N개의 샘플로 blocking하여 다음 프레임으로 사용한다. 본 실험에서는 중첩의 길이 M은 N/2로 사용하였다.

프레임으로 나뉜 데이터는 각 프레임 양끝단의 불연속 지점에서 주파수 영역으로 변환 시 원하지 않는 정보를 최소화하기 위해 각 프레임에 다음 식과 같은 Hamming window를 적용하였다.

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad (2)$$

특징 추출을 위해서 본 연구에서는 잡음에 강인한 특성을 보이는 것으로 알려진 MFCC를 사용하였다.

실험에서 사용한 MFCC 추출 과정을 [그림 1]에 나타내었다.

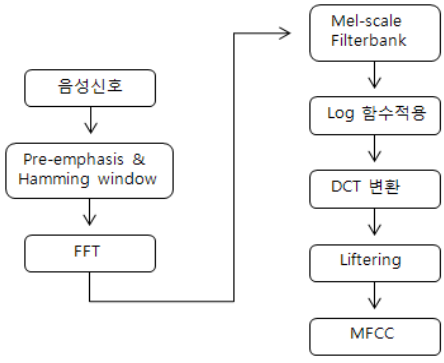

그림 1. MFCC 추출 과정

$c[k]$를 입력 켑스트럼, n이 켑스트럼의 차수, L이 윈도우 계수라고 할 때 본 실험에서 사용한 liftering은 다음 식과 같다.

$$m[k] = \left(1 + \frac{L}{2}\sin\frac{\pi n}{L}\right) c[k] \quad (3)$$

본 실험에서 사용한 윈도우 계수 L은 22이고, 사전 실험에서 MFCC 차수에 따른 인식 성능과 계산량을 고려한 후 12차로 고정하였다.

3. GMM

GMM은 주어진 데이터 집합의 분포밀도를 여러 개의 가우시안 확률밀도함수로 모델링하고 실제 데이터를 기반으로 최대 우도를 가지는 클래스를 선택하는 패턴 인식 방법이다[7][8]. 추출한 특징이 D차라고 할 때 특징 벡터 $\mathbf{X} = \{x_1, x_2, ..., x_N\}$라고 하면 M개의 혼합 성분(Mixture Component)을 가지는 가우시안 확률밀도함수의 우도(Likelihood)는 다음의 식과 같다.

$$p(x_t|\lambda)=\sum_{i=1}^{M}p_i b_i(x_t) \quad (4)$$

$$b_i(x_t)=\frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}}exp\left(-\frac{1}{2}(x_t-\mu_i)^T\Sigma_i^{-1}(x_t-\mu_i)\right) \quad (5)$$

$$\sum_{i=1}^{M}p_i=1,\ 0\le p_i\le 1, \quad (6)$$

GMM 모델의 t 번째 성분 파라미터 λ는 가우시안 혼합 성분 밀도의 가중치(mixture weight) p_i, 평균 벡터(mean vector) μ_i그리고 공분산 행렬(covariance matrix) Σ_i로 나타낼 수 있으며 다음 식을 이용해 반복적으로 구할 수 있다.

$$\lambda=\{p_i,\mu_i,\Sigma_i\},\quad i=1,...,M \quad (7)$$

$$p_i=\frac{1}{T}\sum_{t=1}^{N}p(i|\mathbf{x}_t,\lambda) \quad (8)$$

$$\mu_i=\frac{\sum_{t=1}^{N}p(i|\mathbf{x}_t,\lambda)\mathbf{x}_t}{\sum_{t=1}^{N}p(i|\mathbf{x}_t,\lambda)} \quad (9)$$

$$\Sigma_i=\frac{\sum_{t=1}^{N}p(i|\mathbf{x}_t,\lambda)\mathbf{x}_t^2}{\sum_{t=1}^{N}p(i|\mathbf{x}_t,\lambda)}-\mu_i^2 \quad (10)$$

여기서 i번째 성분의 사후 확률(*a posteriori* probability)은 다음과 같다.

$$p(i|\mathbf{x}_t,\lambda)=\frac{p_i b_i(\mathbf{x}_t)}{\sum_{i=1}^{M}p_i b_i(\mathbf{x}_t)} \quad (11)$$

일반적으로 GMM의 Mixture 수는 인식률과 시스템의 계산량에 직결되는 것으로 적당한 Mixture 수를 고려하는 것은 중요한 문제이다.

III. 후처리 방식

제안 하는 후처리 방식은 한 프레임의 클래스를 결정할 때 해당 프레임에 인접한 프레임들의 정보를 이용하는 방식이다. 여기서 인접한 프레임들이란 과거와 미래의 프레임을 말하고, 이용하는 과거 프레임과 미래 프레임의 수는 양쪽을 동일하게 하여 어느 한 편의 정보에 치우치지 않도록 하였다.

인접 프레임의 정보를 이용하는 데에는 두 가지 방식을 적용할 수 있는데 그 중 하나는 각 프레임 사후확률의 순위를 이용하는 방식이고 나머지 하나는 각 프레임 사후 확률의 값을 이용하는 방식이다. 다음에서는 이 두 가지 방식에 대해 각각 살펴보기로 하자.

1. 순위 방식

일반적인 GMM의 분류에서는 하나의 프레임에 대한 클래스를 정하기 위해서 9개 클래스의 사후 확률을 구하고, 그중 가장 큰 값을 갖는 클래스를 해당 프레임의 클래스로 결정한다. 이에 반해 제안한 후처리 방식에서는 현재 프레임의 클래스를 정하기 위해 인접 프레임의 정보를 동시에 사용한다.

순위를 이용한 후처리에서는 먼저 각 프레임의 9가지 클래스 중 사후 확률이 높은 클래스 순으로 순위를 정하고 그 순위에 따라 순위 값을 할당한다. 다음에는 현재 프레임과 인접 프레임들의 순위 값을 클래스 별로 모두 더한 다음 그 중 가장 높은 순위를 차지한 클래스로 현재 프레임의 클래스를 결정한다. 이 방식은 간접 투표방식과 같은 것으로 갑작스런 클래스 변화를 방지하는 효과를 갖는다.

편의를 위해 시각 t 프레임에서 i번째 클래스의 GMM 사후확률 $P_i(t)$가 전부 미리 계산되어 있다고 하자. i는 클래스의 인덱스이므로 1에서 9까지 변화하고, $rank()$를 입력 배열의 값들을 내림차순으로 정렬하여 순위 값을 반환하는 함수, $minindex()$를 입력된 배열의 값들 중 최소값의 인덱스 i를 반환하는 함수, R_i, S_i 를 초기 값이 0인 배열 이라하면, 인접한 프레임의 사후확률 값 순위를 고려한 후처리 방식을 이용하여 t 프레임의 클래스 $D(t)$는 다음과 같이 얻을 수 있

다. 여기서 고려하는 인접 프레임의 개수 l은 1 또는 2를 사용하였고, 각각 세 프레임과 다섯 프레임을 고려할 때의 설정 값이다.

begin initialize : t, l

$j \leftarrow -l$
do $t = t + j$
$R_i \leftarrow rank(P_i(t))$
$S_i \leftarrow S_i + R_i$
$j \leftarrow j + 1$
until $j \le l$

$D \leftarrow minindex(S_i)$
return D
end

또한 위 과정에서 $minindex()$를 이용하여 클래스를 결정할 때 만약 두 개 이상의 클래스가 동일한 점수를 획득하면 그 전 프레임과 같은 클래스를 부여하였다. 이를 수식으로 표현하면 다음과 같다.

$$R_i(t) = rank(P_i(t)) \tag{12}$$

$$S_i(t) = \sum_{j=-l}^{l} R_i(t+j) \tag{13}$$

$$D(t) = minindex(S_i(t)) \tag{14}$$

2. 사후확률 가중치 방식

사후확률의 값을 이용한 방법에서는 각 프레임 클래스의 순위가 아니라 사후확률 값을 직접 사용한다. 사후확률 값을 사용하는 경우에는 순위를 이용하는 방법과 달리 각 프레임에 가중치를 부여할 수 있으며 동점이 만들어지는 문제를 회피할 수 있다.

사후확률을 이용한 후처리에서는 먼저 각 프레임의 9가지 클래스에 대한 사후확률을 구한다. 그 다음에는 정해진 가중치를 적용하여 현재 프레임과 인접 프레임들의 사후확률을 클래스 별로 모두 더한 다음 그 중 가장 높은 사후확률을 가지는 클래스로 현재 프레임의 클래스를 결정한다. 이 방식은 직접 투표방식과 같은 것으로 간접 투표 방식과 동일하게 갑작스런 클래스 변화를 방지하는 효과를 갖는다.

앞서와 같이 유사하게 $P_i(t)$를 시간 t 프레임의 GMM 사후확률, i를 클래스의 인덱스, $maxindex()$를 입력된 배열의 값 중 최대값의 인덱스 i를 반환하는 함수라고 하고, $w = \{w_{-l}, w_{-l+1}, \cdots, w_0, \cdots, w_{l-1}, w_l\}$를 가중치, T_i를 초기 값이 0인 배열이라고 한다면 사후 확률 가중치를 이용한 후처리 방식은 다음과 같이 나타낼 수 있다.

begin initialize : t, l, $w_j = 1/(2l+1)$

$j \leftarrow -l$
do $t = t + j$
$T_i \leftarrow T_i + w_j \log(P_i(t))$
$j \leftarrow j + 1$
until $j \le l$

$D \leftarrow maxindex(T_i)$
return D
end

여기에서도 i는 1에서 9까지 변화하고, 고려하는 인접 프레임의 개수는 순위방식과 동일하게 1 또는 2를 사용하였다. 사후확률 가중치를 결정하는 방식에는 여러 가지가 있지만 본 연구에서는 순위방식과 비교를 위해 가중치를 모두 동일하게 $w_j = 1/(2l+1)$로 설정하였다.이를 수식으로 표현하면 다음과 같다.

$$T_i(t) = \sum_{j=-l}^{l} w_j \log(P_i(t+j)) \tag{15}$$

$$D(t) = maxindex(T_i(t)) \tag{16}$$

Ⅳ. 실험 방법 및 결과

1. 실험 방법

본 실험에서는 전체 데이터를 임의로 추출하여 훈련 그룹과 테스트 그룹으로 나누었고, 훈련 그룹에서는 GMM을 이용하여 9개의 클래스 각각의 모델을 생성하였다. 테스트 그룹에서는 [표 1]에 나타낸 10 가지의 상

황에 대한 환경음을 묘사하기 위해, 테스트 그룹의 9가지 클래스의 데이터를 각 상황에 맞도록 연결하여 10가지 상황으로 이루어진 테스트 셋을 구성한다. 이때 우리는 공정한 성능평가를 위해 전체 테스트 그룹을 10등분하여 데이터들이 서로 겹치지 않도록 10 세트의 테스트 셋을 준비하여 테스트 하였다. 따라서 실험에서 제시한 인식률은 10개의 상황에 대해 테스트 그룹 내에서 10회의 validation을 수행한 결과이다. 이 훈련 그룹과 테스트 그룹에서 사용한 전체 데이터의 대략적인 크기는 [표 2]와 같다.

표 2. 윈도우 사이즈에 따른 특징 데이터의 개수

윈도우크기 / 그룹	1000ms	750ms	500ms	250ms
테스트 그룹	18000	24000	38000	72000
훈련 그룹	22500	27000	45000	90000

2. 실험결과

후처리가 없는 GMM 방식과 두 가지 후처리를 적용한 방식들에서의 인식 성능을 비교하기에 앞서 GMM 모델의 Mixture 개수를 고정할 필요가 있으므로 우리는 Mixture 개수에 따른 두 가지 후처리 방식의 인식률을 비교해보았고 이를 [그림 2]에 나타내었다. 그래프를 보면 Mixture의 개수가 늘어날수록 각각의 방식에서 인식률은 점차 증가하나 그 증가율이 로그스케일을 따르고 있음을 알 수 있고 그에 비해 계산량은 점차 증가한다. 이에 우리는 각 방식에 따른 인식률을 고려한 후 Mixture의 개수를 14개로 고정하였고, 이를 기준으로 각 방식에서의 인식성능을 비교하였다.

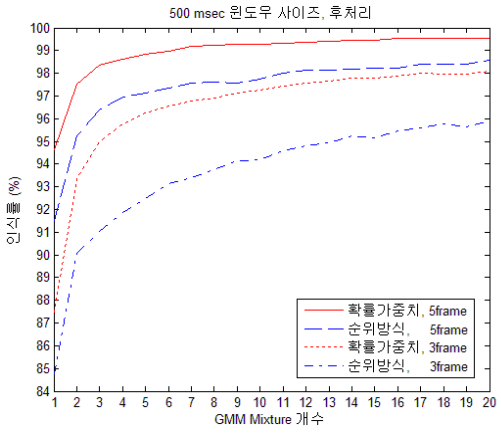

그림 2. GMM Mixture 개수에 따른 인식률

표 3. 후처리 방법을 적용하지 않은 경우 인식률

길이(ms)	250	500	750	1000
분류율	81.03%	83.41%	85.66%	85.84%

위의 [표 3]에는 각각 다른 프레임의 길이로 MFCC를 이용하여 얻은 특징을 후처리 없는 GMM의 결과를 나타내었고, [표 4]에는 이때의 인식 오류를 검토할 수 있는 confusion matrix를 나타내었다. [표 4]를 보면 인식 오류가 주로 시내버스와 KTX, 지하철, 또 실외 뛰기와 실외 걷기 같이 상대적으로 주변잡음이 많으며 서로 유사한 환경음임을 알 수 있고, 기존의 한 프레임만을 고려하는 방식으로 인해 갑자기 환경이 바뀌는 현상이 발생하는 것을 알 수 있다.

표 4. 500msec 12차 MFCC 특징, 14개 Mixture의 후처리 없는 GMM 분류 결과

	버스	강의실	KTX	식당	뛰기	지하철	극장	자동차	걷기
버스	78.33%	0.00%	7.67%	0.00%	4.33%	3.67%	0.67%	4.00%	1.33%
강의실	0.00%	96.00%	0.00%	1.10%	0.00%	0.00%	0.33%	0.00%	2.67%
KTX	6.67%	0.00%	86.33%	0.00%	3.33%	0.00%	0.33%	2.33%	1.00%
식당	0.00%	3.00%	0.00%	90.02%	1.65%	0.67%	1.00%	0.00%	3.67%
뛰기	1.33%	0.00%	2.33%	1.00%	77.00%	3.00%	4.00%	0.33%	11.00%
지하철	2.33%	0.00%	0.00%	0.00%	1.00%	86.67%	2.33%	5.00%	2.67%
극장	0.00%	0.67%	0.00%	1.67%	4.33%	1.67%	84.67%	1.33%	5.67%
자동차	5.00%	0.00%	5.67%	1.00%	1.33%	7.33%	2.33%	75.00%	2.33%
걷기	1.00%	0.67%	0.00%	2.33%	11.33%	1.67%	4.33%	2.00%	76.67%

[표 5]와 [표 6]에서는 두 가지 후처리 방식을 사용하여 분류한 결과를 나타내었다. 순위 방식을 적용한 [표 5]의 결과에서는 750ms의 프레임 길이로 구한 MFCC 특징을 좌우 각각 두 개의 프레임을 고려하였을 때 최적의 인식률인 98.38%를 보여주는데, 이는 [표 3]의 750ms의 프레임 길이로 구한 MFCC 특징을 후처리 없는 GMM을 이용한 인식 성능인 85.66%보다 12.72%의 성능 개선을 이루었음을 확인 할 수 있다. 확률 가중치를 이용한 [표 6]의 결과에서는 500ms의 프레임 길이로 구한 MFCC 특징을 좌우 두 개의 프레임을 고려하였을 때 99.46%의 성능을 보여주고 있는데, 이는 [표 3]의 후처리 없는 방식에 비해 16.05% 향상된 성능을 보여주는 결과이다. 따라서 후처리 방법을 적용하지 않았을 경우에 비해 후처리방법을 적용한 경우에서 환경음 인식 성능이 확연하게 개선되는 것을 알 수 있다.

표 5. 순위 방식 후처리를 적용한 경우 인식률

사용프레임 / 길이 (ms)	3 frames	5 frames
250	94.02%	97.59%
500	95.37%	98.27%
750	95.53%	98.38%
1000	95.11%	98.01%

표 6. 확률 가중치 후처리를 적용한 경우 인식률

사용프레임 / 길이 (ms)	3 frames	5 frames
250	97.34%	98.74%
500	98.08%	99.46%
750	98.31%	99.39%
1000	97.59%	98.64%

[표 5]와 [표 6]를 비교하면 잡음이 매우 커서 사후 확률의 변동이 클 때 순위 방식은 확률 가중치 방식보다 더 좋은 성능을 보일 가능성이 있지만 본 실험과 같이 일반적인 잡음 상황에서는 순위 방식이 확률 가중치를 이용한 방식에 비해 전반적으로 인식률이 약간 떨어지는 것을 알 수 있다. 같은 표에서 인접한 프레임의 개수를 한 개 사용하는 것보다 두 개를 고려하는 경우가 더 좋은 성능을 보이고 있는데 이는 데이터의 길이가 길수록 인식 성능이 조금씩 나아지는 일반적인 경향과 일치하는 것이다.

인식률을 높이기 위해 인접 프레임을 몇 개 사용하는가에는 고려해야 할 문제가 있다. 가령 프레임의 길이가 0.5초인 경우에 반씩 중첩을 시킨다고 하면 인접 프레임을 한 개 혹은 두 개 고려하는 경우 시스템의 응답 지연이 각각 0.25초와 0.5초가 된다. 이것은 현재 프레임을 포함하면 시스템이 결국 0.75초와 1초 간격으로 상황인지를 한다는 것이다. 따라서 응답 지연을 고려할 때 현재의 조건 하에서는 1초를 넘는 시간 지연을 갖는 인접 세 프레임을 고려하는 것은 문제가 될 수 있다.

표 7. 프레임의 개수와 길이에 따른 응답 지연시간

사용프레임 / 길이 (ms)	3 frames	5 frames
250	125ms	250ms
500	250ms	500ms
750	375ms	750ms
1000	500ms	1000ms

이와 관련하여 같은 시간 지연을 갖는 프레임의 길이 1000ms로 후처리 없는 GMM 방식(즉 1초 간격으로 상황인지를 하는 시스템)의 평균 인식률인 85.84%과 역시 1초 간격의 상황인지를 하는 500ms의 특징을 이용한 두 가지 후처리 방식에서의 인식률을 비교했을 때 후자의 인식률이 월등하다는 것은 본 논문에서 제안하는 후처리 방식이 큰 의미가 있다고 할 수 있다. 1frame을 고려하는 기존연구의 경우를 지연시간이 없다고 가정하였을 때 인접 프레임의 개수와 길이에 따른 응답 지연시간을 위의 [표 7]에 제시하였다.

V. 결 론

기존의 GMM 분류를 이용한 환경음 인식은 한 프레임만을 고려하는 방식으로 인해 하나의 연속적인 환경

상황에서 갑자기 다른 환경으로 인식하는 인식오류를 발생시킬 수 있다. 본 연구에서 사용한 상황에서는 이러한 인식 오류가 주로 시내버스와 KTX, 지하철, 또 실외 뛰기와 실외 걷기 같이 상대적으로 주변잡음이 많으며 서로 유사한 환경음이었다. 이와 같은 인식 오류를 개선하는 방법으로 본 연구에서는 기존의 GMM 분류 방식에 인접 프레임의 확률적인 정보를 이용한 후처리의 도입을 제안하였고, 그 결과 GMM에 사후확률 가중치를 이용한 후처리 방식을 적용한 경우에서 평균 인식률을 99% 이상으로 끌어올릴 수 있었다. 이를 통해 본 연구에서는 적절한 후처리 방식의 도입으로 환경음의 인식 성능을 개선시킬 수 있다는 것을 확인하였다.

하지만 환경음 인식의 실제 서비스를 고려해 볼 때 환경음의 종류는 굉장히 방대하므로 이들 전체를 DB로 구성하고 이를 학습시키는 것은 한계가 있음을 알 수 있다. 때문에 인식하고자 하는 특정 환경음 이외의 환경음이 마이크로폰에 입력된다는 가정 하에 이를 처리하는 방법이 필요하며, 앞으로의 연구에서는 실제 서비스를 고려하여 환경음의 도메인이 정해져 있지 않다는 가정 하에 인식성능을 유지하는 방법을 강구할 것이다.

참 고 문 헌

[1] http://www.teco.edu/tea/

[2] L. Ma, B. P. Milner, and D. Smith, “Acoustic environment classification,” ACM Transactions on Speech and Language Processing, Vol.3, No.2, pp.1-22. 2006.

[3] Y. Toyoda, J. Huang, S. Ding, and Y. Liu, “Environmental sound recognition by multilayered neural networks,” International Conference on Computer and Information Technology, pp.123-127, 2004.

[4] L. Couvreur and M. Laniray, “Automatic noise recognition in urban environments based on artificial neural networks and hidden Markov models,” InterNoise, Prague, Czech Republic, pp.1-8. 2004.

[5] N. Sawhney, “Situational awareness from environmental sounds,” MIT Media Lab. Technical Report, 1997.

[6] S. Chu, S. Narayana, C.-C. J. Kuo, and M. J. Mataric, "Where am I? Scene recognition for mobile robots using audio features," in Proc. ICME, 2006.

[7] R. G. Malkin and A. Waibel, "Classifying user environment for mobile applications using linear autoencoding of ambient audio," in Proc. ICASSP, 2005.

[8] A. Eronen, V. Peltonen, J. Tuomi, A. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," IEEE Trans. on Audio, Speech, and Language Processing, Vol.14, No.1, pp.321-329, 2006(1).

[9] S. Chu, S. Narayanan, and C.-C. Jay Kuo “Environmental Sound Recognition With Time-Frequency Audio Features,” IEEE Trans. on Audio, Speech, and Language Processing, Vol.17, No.6, pp.1-16, 2009.

[10] C. M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, UK, 1995.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classfication*, Jone Wiley & Son, Inc. 2001.

저 자 소 개

박 준 규(Jun-Qyu Park) 준회원

- 2009년 2월 : 전남대학교 전자컴퓨터공학부(공학사)
- 2009년 3월 ~ 현재 : 전남대학교 전자공학과(석사과정)

<관심분야> : 디지털 신호처리, 패턴인식

백 성 준(Seong-Joon Baek) 정회원

- 1989년 2월 : 서울대학교 전자공학과(공학사)
- 1992년 2월 : 서울대학교 전자공학과(공학석사)
- 1999년 2월 : 서울대학교 전자공학과(공학박사)
- 2002년 3월 ~ 현재 : 전남대학교 전자공학과 교수

<관심분야> : 의료, 통신, 음성 관련 디지털 신호처리