DOI: 10.3745/KIPSTB.2009.16-B.2.151

FM 라디오 환경에서의 실시간 음악 판별 시스템 구현

강 현 우†

요 약

본 연구에서는 GMM 기반의 음성/음악 판별 방법을 응용하여 FM 라디오 방송에서 순수한 음악 구간만을 판별하는 시스템을 구현하였다. 본 시스템에서는 음성, 음악, 광고 음악, 기타 여러 가지 사운드가 혼합되어 있는 오디오 방송 프로그램에서 순수한 음악만을 판별하여 자동으로 저장하고자 한다. 음악의 시작 부분과 끝 부분을 보다 정교하게 검출하고자 순수한 음악으로 판별된 구간의 시작 부분과 끝 부분에 대해 후처리 과정을 추가하였다. PC 환경에서 FM 라디오 방송을 이용하여 구현된 시스템을 실시간으로 테스트한 결과 우수한 성능을 보임을 확인하였다. 또한 SoC 구현을 고려하여 고정소수점 연산을 수행한 결과 3MIPS 이하의 적은 연산량으로 부동소수점 연산일 때와 동일한 결과를 얻을 수 있었다.

키워드 : 음성/음악 판별, 가우시안 혼합 모델, 고정소수점 연산, 후처리 과정

Implementation of Music Signals Discrimination System for FM Broadcasting

Kang Hyun-Woo†

ABSTRACT

This paper proposes a Gaussian mixture model(GMM)-based music discrimination system for FM broadcasting. The objective of the system is automatically archiving music signals from audio broadcasting programs that are normally mixed with human voices, music songs, commercial musics, and other sounds. To improve the system performance, make it more robust and to accurately cut the starting/ending-point of the recording, we also added a post-processing module. Experimental results on various input signals of FM radio programs under PC environments show excellent performance of the proposed system. The fixed-point simulation shows the same results under 3MIPS computational power.

Keywords : Speech/Music Discrimination, GMM, Fixed-Point Simulation, Post-Processing

1. 서 론

최근 들어 음성, 음악, 배경 음악, 효과음 등 다양한 형태의 사운드가 포함된 오디오 신호를 특정 카테고리로 판별(audio discrimination)하거나 여러 가지 카테고리로 분류(audio classification)하는 연구가 진행되고 있다[1-4].

음성의 경우 주기적 특성의 유성음과 비주기적 특성의 무성음이 연결되어 구성되고 발성 중간에 묵음 구간이 존재하지만, 음악의 경우는 음성에 비해 무성음과 묵음의 비중이 낮고 음성의 주기적 특성인 피치보다 넓은 주기의 특성인 리듬을 갖기 때문에 오디오 신호를 음성과 음악으로 판별하는 것이 가능하다[5]. 음성과 음악의 고유한 특성을 이용하여 여러 가지 특성 파라미터를 만들 수 있고, 이러한 특성 파라미터로부터 음성과 음악을 최종적으로 판별하는 방법은 이분법 판별 방법과 통계적 모델 기반 방법 등이 있다. 통계적 모델 기반 방법은 각 파라미터의 분포를 통계적 모델로 만들어 확률을 비교하는 방법으로 NN(Neural Network), HMM(Hidden Markov Model), GMM(Gaussian Mixture Model) 등의 방법이 있고, 이 중 음성 인식, 화자 인식, 음성 변조 등의 음성 신호처리 분야에 GMM이 가장 널리 이용되고 있다[6].

본 연구에서는 이러한 음성/음악 판별 방법을 응용하여 FM 라디오에서 수신된 신호 중 순수한 음악 구간만을 판별하고, 이를 선택적으로 저장할 수 있는 시스템을 개발하였다. FM 라디오 방송 환경의 프로그램에서는 여러 화자가 동시에 이야기하는 경우나 박수 소리 등을 음악으로 판별할 가능성이 높고, DJ 멘트가 배경 음악과 혼합되는 부분도 음악으로 판별할 가능성이 높다. 또한 광고 음악(Commercial Song)이나 로고송(Logo song)을 순수한 음악으로 잘못 판단하기 쉽다. 이러한 여러 가지 문제점 때문에 FM 라디오 프로그램에서 순수한 음악 구간만을 검출하는 것은 일반적인 음성/음악 판별보다 훨씬 더 복잡한 문제이다. 따라서

※ 본 논문은 2008년도 강남대학교 교내연구비 지원에 의한 것임.
† 정 회 원 : 강남대학교 컴퓨터미디어공학부 교수
논문접수 : 2008년 11월 1일
수 정 일 : 1차 2008년 12월 10일
심사완료 : 2008년 12월 26일

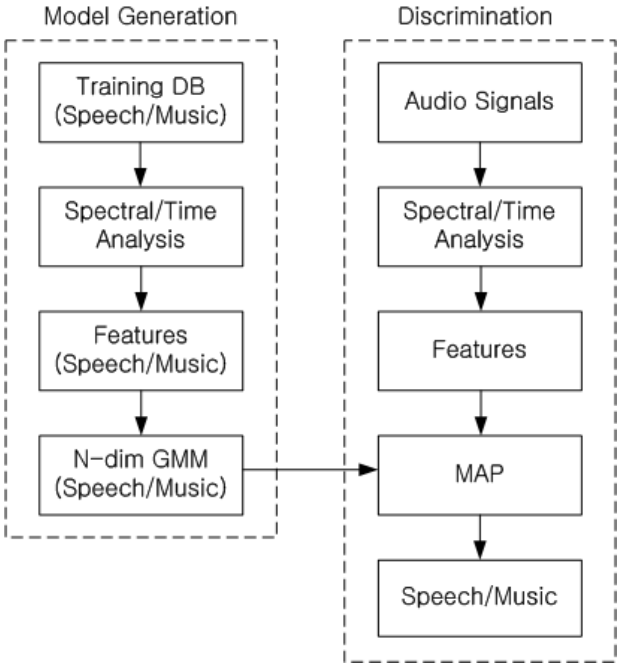
음성과 음악의 특성 파라미터를 이용한 음성/음악 판별 방법에 방송 프로그램의 부가적인 특징을 찾아내어 규칙을 추가할 필요가 있다. 본 논문에서는 이동 방송 수신기 형태로 이와 같은 기능을 구현할 것을 고려하여 적은 연산량만으로 구현할 수 있는 알고리듬을 고려했으며, 시간 영역의 특징 파라미터를 이용하기 때문에 성능이 우수하고 연산량이 비교적 적은 GMM[7,8]을 기반으로 하였다. 음악의 시작 부분과 끝 부분을 보다 정교하게 검출하고자 확실한 음성이나 음악이라고 판별하기 어려운 경우 천이(transition) 상태로 구분하고, 후처리 과정(post-processing)에서 최종 결정을 하였다. 이와 관련된 국내외 연구 결과의 대부분은 오디오 신호의 장르를 판별하기 위한 전처리 단계로 사용하거나[2], 또는 음성/음악을 제대로 판별하는 비율에 중점을 둔 논문들[1,3]이고, 실제 응용을 고려하여 음악만을 따로 저장하기 위한 후처리 과정을 구현하는 경우는 없었다.

본 논문의 구성은 다음과 같다. 2장에서는 GMM 기반의 음성/음악 판별 방법에 대해 설명하고, 3장에서는 PC 환경에서 실시간으로 음악 구간만을 판별하여 저장하는 시스템에 대해 설명한다. 4장에서는 구현한 시스템의 실험 결과를 살펴보고, 5장에서는 SoC로 구현하기 용이하게 하기 위한 고정소수점(fixed-point) 연산 결과에 대해 설명한다. 끝으로 6장에서 결론을 맺는다.

2. GMM 기반의 음성/음악 판별 방법

GMM(Gaussian Mixture Model)은 음성 신호처리 분야에 가장 널리 이용되는 통계적 모델 기반의 클러스터링 방법이다. GMM은 특성 파라미터의 확률 분포를 여러 개의 정규분포의 합으로 나타내는 방법으로 연산량이 비교적 적고 통계 모델을 잘 반영하는 장점을 가지고 있다. 특히 상관관계가 적은 여러 개의 특성 파라미터를 사용하는 다차원 GMM을 이용하면 카테고리간의 중복 구간을 최소화할 수 있어서 판별 성능을 향상시킬 수 있다[7,8].

음성과 음악을 판별하기 위한 알고리즘은 <그림 1>에서와 같이 크게 훈련(training) DB로부터 판별 기준이 되는 모델을 생성하는 과정과 입력 오디오 신호를 음성 또는 음악으로 판별하는 과정으로 구성된다. 먼저 음성과 음악 각각의 특성을 잘 반영하는 특성 파라미터를 여러 개 선택한다. 모델 생성 과정에서는 엄선된 음성과 음악 각각의 훈련 데이터로부터 정해진 단위 길이마다 시간 또는 주파수 영역 분석을 통해 특성 파라미터들을 계산하고, 이 값들의 확률 분포를 이용해 음성과 음악 각각의 N차원 GMM을 생성한다. 음성/음악 판별 과정에서는 입력된 오디오 신호로부터 단위 길이마다 동일한 특성 파라미터를 계산하고, 이를 음성과 음악 각각의 GMM 모델과 비교하여 각각의 확률 값을 구하게 된다. 이 확률 값을 MAP(Maximum a Posteriori) 방법으로 비교하여 입력 오디오 신호가 음성인지 음악인지를 최종 결정하게 된다[7].



〈그림 1〉 GMM 기반의 음성/음악 판별 알고리즘

3. 실시간 소프트웨어 구현

3.1 특성 파라미터

본 연구에서는 GMM 모델을 생성하기 위한 훈련 데이터로 라디오 방송에서 녹음한 음성과 음악 데이터를 각각 사용하였다. 음성의 경우 20명의 아나운서 또는 DJ들의 음성을 녹음한 72분 분량을 사용했으며, 음악의 경우 다양한 장르의 음악 26곡을 편집하여 사용하였다. 음성과 음악의 고유한 특성을 반영하는 여러 가지 파라미터 중 본 연구에서는 MLER(Modified Low Energy Ratio), HZCRR(High Zero-Crossing Rate Ratio), JEZ(Joint of the Energy and ZCR), SRPF(Speech Rate in the Pitch Frequency)의 4가지를 선택했으며, 이들 조합에 의한 음성/음악 판별 성능은 이미 [4]에서 검증되었다. 이 때 GMM의 mixture 수는 성능과 계산량을 고려하여 16으로 정하였다.

각 특성 파라미터의 정의는 다음과 같다. MLER은 식 (1)과 같이 일정 단위 길이에서 평균 단구간 에너지의 δ 배보다 작은 에너지 크기를 갖는 단구간 개수의 비로 정의된다[1].

$$MLER = \frac{1}{2N}\sum_{n=1}^{N}\left[sgn\left(\delta E_{mean} - E_n\right)+1\right].$$
$$E_{mean} = \frac{1}{N}\sum_{n=1}^{N} E_n. \quad (1)$$

여기에서 n 은 프레임 인덱스, E_n 은 n 번째 프레임의 평균 단구간 에너지, N은 프레임의 개수, δ는 제어 변수, $sgn[]$는 sign 함수를 각각 나타낸다.

HZCRR은 식 (2)와 같이 일정 단위 길이에서 영교차율

(Zero-Crossing Rate)이 평균 영교차율의 1.5배보다 큰 프레임 수의 비로 정의된다[2].

$$HZCRR = \frac{1}{2N}\sum_{n=1}^{N}\left[sgn\left(ZCR_n - 1.5\,ZCR_{mean}\right)+1\right]. \quad (2)$$

여기에서 ZCR_n 은 n 번째 프레임의 영교차율, ZCR_{mean} 은 평균 영교차율을 각각 나타낸다.

JEZ은 식 (3)과 같이 단구간 에너지 크기와 영교차율의 곱을 단위 길이 구간 동안 더한 값을 $2E_{\max} - E_{\min} - E_{mean}$ 으로 정규화한 값으로 정의된다[9].

$$JEZ = \frac{\sum_{n=1}^{N} E_n\, ZCR_n}{2E_{\max} - E_{\min} - E_{mean}}. \quad (3)$$

여기에서 $E_{\max}$는 단구간 에너지의 최대값, $E_{\min}$은 단구간 에너지의 최소값을 각각 나타낸다.

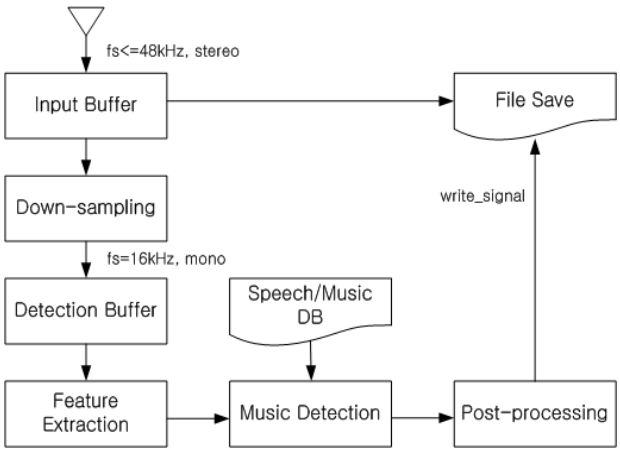
SRPF은 식 (4)와 같이 정의되며[4], 이는 신호의 주기적인 특성을 나타내는 기본 주파수를 이용해 단위 길이 구간에서 음성의 주파수 범위에 포함되는 주기값을 갖는 단구간의 개수를 나타낸다.

$$SRPF = \frac{M_{SR}}{M - M_0 + \alpha}. \quad (4)$$

여기에서 M_{SR} 은 주파수 범위에 포함되는 기본 주파수를 갖는 주기적 특성의 단구간 개수, M은 단 구간의 총 개수, M_0는 기본 주파수를 갖지 않는 비주기적 특성의 단구간 개수, α 는 임의의 상수를 각각 나타낸다.

3.2 실시간 소프트웨어 구현

Pentium-IV 3GHz 사양의 컴퓨터를 사용했으며, Visual Studio 환경에서 C 언어와 API(Application Programming Interface)를 이용하여 구현하였다. 구현된 실시간 음악 판별 시스템의 블록 다이어그램을 <그림 2>에 나타내었다. 실시간 구현 시스템에서는 컴퓨터 사운드카드의 라인 입력 단자(Line-in)를 통해 입력되는 오디오 신호를 실시간으로 디지털 신호로 변환(Analog-to-Digital Conversion)하여 입력 버퍼(input buffer)에 저장하였다. 이 때 샘플링 주파수는 44.1kHz, 샘플당 16bit, 스테레오(stereo)로 샘플링했다. 일반적으로 음성/음악 판별 시 한 프레임의 길이를 1초로 하지만, 본 시스템에서는 정확도를 높이기 위해 0.5초로 하였다. 준비된 두 개의 버퍼를 통해 저장과 처리를 번갈아 가면서 연속적으로 처리하는 더블 버퍼링(double buffering) 과정을 쓰레드(thread)를 이용하여 구현하였다. 매 프레임마다 특성 파라미터를 추출하기 위해 16kHz, 모노(mono)로 다운샘플링(down-sampling)하여 판별 버퍼(detection buffer)에 저장



〈그림 2〉 실시간 음악 판별 시스템의 블록 다이어그램

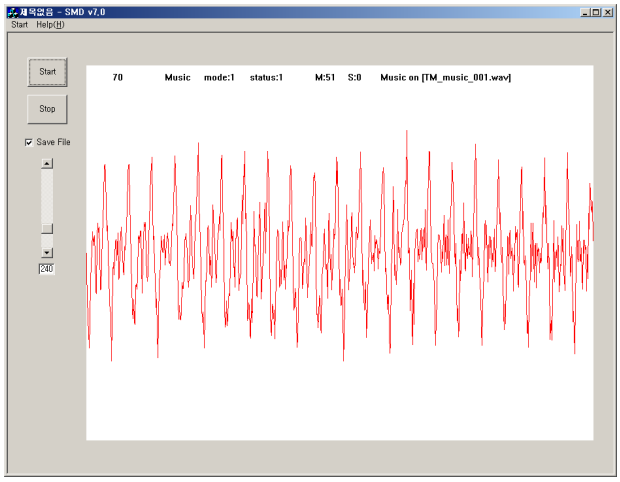
한 후 이 데이터를 이용하여 이 구간의 오디오 신호가 음성인지 음악인지를 판별하여 모니터에 다른 색깔의 파형으로 표현한다. 그리고 프레임마다 판별된 음성/음악 정보를 이용해서 순수한 음악 구간만을 파일로 저장하고자 한다.

라디오 프로그램의 경우 저장하고자 하는 순수한 음악 이외에도 광고 음악이나 로고송 등이 존재한다. 또한 배경 음악과 함께 DJ가 멘트를 하는 경우나 여러 명의 DJ나 게스트(guest)들이 동시에 이야기를 하는 경우는 물론 박수 소리 등의 비음악을 음악으로 잘못 판별할 가능성이 높다. 뿐만 아니라 프레임에 기초한 음성/음악 판별 알고리듬의 경우 음성 프레임을 음악으로 잘못 판별하는 경우나, 반대로 순수한 음악 프레임을 음성으로 잘못 판별되는 경우가 존재할 수 있는데 이는 시스템의 성능에 커다란 악영향을 미치게 된다. 즉 기본적으로 음악 프레임이 수 프레임 이상 연속되면 음악 구간이라고 판단하여 저장을 시작하고, 음성 구간이 수 프레임 이상 연속되면 음악이 끝났다고 판단하여 저장을 중지하는 방법만으로는 성능에 한계가 있게 된다. 따라서 후처리 과정을 통해 잘못된 판별에 의한 영향력을 최소화하고 순수한 음악의 시작 부분과 끝 부분을 보다 정교하게 검출할 필요가 있다.

이를 위하여 천이(transition) 상태를 도입하여 어떤 프레임이 확실히 음성인지 음악인지 판별하기 어려운 경우 천이 상태로 구분하고, 후처리 과정을 통해 최종적으로 결정을 하였다. 후처리 과정에서는 연속된 음악/음성 프레임의 수, 에너지의 변동 폭, 특성 파라미터의 통계적 특성 변화 등을 고려하였다. 또한 순수한 음악의 경우 대부분 음악의 끝부분에서 소리가 줄어들기 때문에 에너지가 서서히 줄어드는 경우를 음악의 끝부분이라고 판별하였다.

4. 실험 결과

<그림 3>은 실시간 음악 판별 시스템의 동작 화면을 캡쳐(capture)한 것이다. 현재 프레임이 음악 프레임이며 음악 검출 알고리즘에서 현재의 상태를 순수한 음악으로 판단하



〈그림 3〉 실시간 음악 판별 시스템의 동작 화면

여 wave 파일로 저장하고 있음을 보여준다.

후처리 과정에서 연속된 음악/음성 프레임의 수, 에너지의 변동 폭, 특성 파라미터의 통계적 특성 변화 등을 이용할 때 다양한 환경에 대응할 수 있도록 튜닝(tuning) 과정을 통해 변인들의 세부적인 문턱치(threshold)를 결정하였다. 이러한 튜닝 과정으로도 모든 장르의 음악을 정확히 분류해 내기는 쉽지 않다. 클래식과 가곡의 경우 가요나 팝송과는 달리 음악의 시작과 끝 부분에서 레벨이 아주 낮으며 곡의 중간에도 아주 낮은 레벨을 보이는 경우가 많다. 가요와 팝송에 맞게 변인들을 튜닝한 후처리 과정을 클래식과 가곡에 적용을 하면 음악의 중간이나 앞, 뒤 부분이 잘리는 경우가 많이 발생하였다. 따라서 입력 데이터가 가요/팝송인 경우와 클래식/가곡인 경우로 구분하여 후처리 과정을 튜닝 하였으며 그 결과로 실험을 진행하였다. 곡의 길이를 조사한 553곡의 음악 중 2분 이하의 곡은 단 2곡(0.36%)이었고, 대부분 음악의 길이가 2분 이상이었기 때문에 광고 음악을 일차적으로 걸러내기 위해 연속된 음악 프레임이 2분 이상 되는 경우에만 순수한 음악이라고 판별하였다. 연속된 음악이 2분 이내인 경우 음악이 중간에 잘리는 것을 방지하기 위해 음악의 끝이라고 판단하는 문턱치를 높게 설정하였고, 음악이 2분 이상 연속되는 경우 문턱치를 낮게 설정하였다.

2007년 11월 21일 낮 12시부터 3시간 동안 KBS2 FM 라디오 방송과 SBS FM 라디오 방송, 2008년 3월 17일 오전 11시부터 3시간 동안 MBC FM 라디오 방송, 그리고 2008년 3월 18일 낮 12시부터 5시간 동안 KBS1 FM 라디오 방송 신호로 실험한 결과를 <표 1>에 나타내었다. KBS2, SBS, MBC 음악은 모두 가요와 팝송이었고, KBS1의 경우는 모두 클래식과 가곡이었다. 여기에서 Grade A는 한 곡을 정확히 저장한 경우, Grade B는 한 곡을 저장하지만 음악 앞 또는 뒤에 순수한 음악이 아닌 사운드가 저장되어 편집이 필요한 경우, Grade C는 한 곡의 일부분이 잘리는 경우, Grade D는 순수한 음악이 아닌 사운드가 저장된 경우로 구분하여 평가하였다. 순수한 음악이 아닌 사운드에는 상업 광고의 배경으로 사용하는 음악(Commercial song)이나 로고송

〈표 1〉 실시간 음악 판별 실험 결과
(단위: 곡(%))

Grade	가요/팝송			클래식/가곡	합계	설명
	KBS2	SBS	MBC	KBS1		
A	14(73.7)	3(21.4)	3(17.6)	12(36.4)	32(38.6)	한 곡을 정확히 저장
B	3(15.8)	10(71.4)	10(58.8)	19(57.6)	42(50.6)	한 곡 저장, 편집 필요
C	2(10.5)	1(7.1)	4(23.5)	2(6.1)	9(10.8)	곡의 일부가 잘림
D	0(0.0)	0(0.0)	0(0.0)	0(0.0)	0(0.0)	순수한 음악이 아님
합계	19	14	17	33	83	

(Logo song)뿐만 아니라 DJ의 음성이나 박수 소리 등도 포함된다.

Grade A가 가장 좋은 결과이며, 일반적으로 Grade B를 허용할만한 수준이라고 할 수 있다. 그리고 Grade C와 D는 좋지 않은 경우라고 할 수 있다. KBS2 데이터의 경우에는 전체 19개 곡 중에서 Grade A와 Grade B가 17곡(89.5%)을 차지할 정도로 성능이 우수했고, 곡의 일부분이 잘리는 Grade C의 경우 2곡(10.5%)에 불과하였다. SBS 데이터의 경우에는 전체 14개 곡 중에서 Grade A와 Grade B가 13곡(92.8%)을 차지하였고, 곡의 일부분이 잘리는 Grade C의 경우 1곡(7.1%)에 불과한 것을 확인하였다. MBC 데이터의 경우에는 전체 17개 곡 중에서 Grade A와 Grade B가 13곡(76.4%)을 차지해서 무난한 성능을 보이는 것을 볼 수 있었다. 또한 클래식과 가곡으로 구성된 KBS1 데이터의 경우도 33곡 중 Grade A와 Grade B가 31곡(94.0%)을 차지할 정도로 성능이 뛰어났다. 네 가지 경우 모두 순수한 음악이 아닌 사운드가 저장되는 Grade D의 경우는 한 건도 없었으며, 반대로 순수한 음악을 저장하지 못하는 경우도 없었다.

편집이 필요한 Grade B의 경우는 전주와 함께 시작하는 DJ의 멘트나 음악이 끝나기 전의 DJ 멘트, 여러 화자(DJ, guests)가 동시에 말하는 소리, 음악 뒷부분에 이어지는 박수 소리 등이 음악의 앞/뒤에 추가되는 경우가 대부분이었다. 특히 MBS 데이터의 경우 두 명의 DJ가 코믹하게 진행하는 상황이라 음악의 시작이나 끝 부분과 중첩되게 두 명이 동시에 멘트를 하는 경우가 많아서 성능이 떨어졌다고 판단된다. 클래식/가곡의 경우에는 연속되는 2~3곡의 음악을 분리하지 못하고 이어서 저장한 경우도 있었으며, 이 경우 Grade B로 계산하였다. 음악의 일부분이 잘리는 Grade C의 경우 음악의 중간에 fade out되는 것처럼 레벨이 급격히 줄어드는 경우나 수 초 이상 비음악적인 요소가 강하게 나타나는 경우에 음악의 앞부분이나 뒷부분에서 수 초~수십초 정도 잘리는 경우가 대부분이었다.

Grade B의 경우 몇 초의 사운드가 음악의 앞/뒤에 추가되었는지, Grade C의 경우 음악의 앞/뒤에서 몇 초가 잘리는지를 분석한 결과를 <표 2>에 나타내었다. 음악의 앞과

〈표 2〉 Grade B에서 사운드가 추가되는 시간과 Grade C에서 음악이 잘리는 시간 분석 (단위: 곡(%))

시간	Grade B	Grade C
1~10초	21(70.0)	4(36.4)
11~20초	4(13.3)	0(0.0)
21~30초	0(0.0)	1(9.1)
31초 이상	5(16.7)	5(45.5)
합계	30	11

뒤에 모두 사운드가 추가되거나 잘리는 경우는 각각의 시간을 더했고, 한쪽에서 추가되고 다른 쪽에서 잘리는 경우는 각각 따로 계산하였다. 또한 2~3곡의 음악이 분리되지 못하고 이어서 저장된 6가지 경우는 고려하지 않았다. 그 결과 Grade B의 경우 10초 이하의 짧은 사운드가 추가되는 경우가 70.0%로 대부분이었고, 11~20초 사이가 13.3% 그리고 31초 이상이 16.7%로 어느 정도 나타났다. Grade C의 경우 10초 이하의 짧은 길이가 잘리는 경우가 36.4%고, 21~30초 사이가 9.1% 그리고 31초 이상 과도하게 잘리는 경우가 45.5%나 되었다.

5. 고정소수점 연산 구현

고정소수점 연산(Fixed-point simulation)을 위해 ITU-T에서 표준화한 STL-2000 라이브러리[10]에서 제공하는 함수들을 사용하였는데, 이 함수들은 16 비트 프로세서를 기반으로 C 언어로 정의되어 있다. 표준 라이브러리를 사용하면 일반화된 명령어를 사용하기 때문에 프로세서에 따라 고유하게 정의된 명령어에 의존하지 않을 수 있다. 또한, 각 프로세서마다 고유하게 설계되어야만 하는 에뮬레이터(emulator) 혹은 시뮬레이터(simulator)가 없어도 알고리즘 구현에 필요한 복잡도 및 메모리 사용량 등을 미리 예측할 수 있다는 장점도 있다. STL에서는 복잡도를 계산하기 위해 WMOPS(Weighted Millions Operations Per Second) 단위를 사용하는데, 이는 각 명령어를 수행하기 위해 일반적으로 필요한 사이클을 가중치로 표시하여 전체 연산량을 계산하는 방식으로써 각 명령어를 수행하는 시간이 프로세서마다 다를 경우에도 쉽게 조정 가능하다. <표 3>은 STL에서 사용하는 명령어와 각 명령어 수행에 필요한 가중치에 대한 예이다.

3장에서 설명한 바와 같이 제안된 알고리즘은 특징 벡터 추출, GMM likelihood 연산 및 비교, 그리고 후처리 단계로 구성되어 있다. <표 4>는 각 단계 별로 필요한 연산량을 정리한 값이다. <표 4>에서 사용한 가중치는 일반적인 16비트 DSP 프로세서의 MIPS(Millions Instructions Per Second)과 비슷한 값을 갖도록 조정한 것으로써 결국, 제안된 알고리즘은 3 MIPS 이하의 매우 낮은 연산량이 필요함을 확인할 수 있다. 또한, 전체 연산량 중 약 73%는 특징 벡터 추출 단계에서 필요하므로 알고리즘의 성능뿐만 아니라 연산량 측면에서도 특징 벡터를 추출하는 단계가 전체 시스템의 성능을 결정짓는 가장 중요한 단계라고 판단된다.

〈표 3〉 STL-2000 16비트 연산 명령어 및 가중치

명령어	가중치
add(Word16 var1, Word16 var2);	1
sub(Word16 var1, Word16 var2);	1
abs_s(Word16 var1);	1
shl(Word16 var1, Word16 var2);	1
shr(Word16 var1, Word16 var2);	1
mult(Word16 var1, Word16 var2);	1
L_mult(Word16 var1, Word16 var2);	1
negate(Word16 var1);	1
extract_h(Word32 L_var1);	1
extract_l(Word32 L_var1);	1
round(Word32 L_var1);	1
L_mac(Word32 L_var3, var1, var2);	1
L_msu(Word32 L_var3, var1, var2);	1
L_macNs(Word32 L_var3, var1, var2);	1
L_msuNs(Word32 L_var3, var1, var2);	1
L_add(Word32 L_var1, L_var2);	2
L_sub(Word32 L_var1, L_var2);	2
L_add_c(Word32 L_var1, L_var2);	2
L_sub_c(Word32 L_var1, L_var2);	2
L_negate(Word32 L_var1);	2
mult_r(Word16 var1, Word16 var2);	2
L_shl(Word32 L_var1, Word16 var2);	2
L_shr(Word32 L_var1, Word16 var2);	2
shr_r(Word16 var1, Word16 var2);	2
mac_r(Word32 L_var3, Word16 var1, Word16 var2);	2
msu_r(Word32 L_var3, Word16 var1, Word16 var2);	2
L_deposit_h(Word16 var1);	2
L_deposit_l(Word16 var1);	2
L_shr_r(Word32 L_var1, Word16 var2);	3
L_abs(Word32 L_var1);	3
L_sat(Word32 L_var1);	4
norm_s(Word16 var1);	15
div_s(Word16 var1, Word16 var2);	18
norm_l(Word32 L_var1);	30

이와 같이 구현된 고정소수점 연산 프로그램을 이용하여 <표 1>의 데이터에 대해 동일한 실험을 한 결과 부동소수점 연산 프로그램의 결과와 동일한 결과를 얻을 수 있었다.

〈표 4〉 각 프로세싱 단계별 연산량

프로세싱 단계	연산량 (WMOPS)
특징 벡터 추출	2.10
GMM likelihood 연산 및 비교	0.57
후처리	0.21
합계	2.88

이는 부동소수점 연산 프로그램을 고정소수점 연산으로 변환할 때 발생할 수 있는 성능 저하가 전혀 없음을 의미한다. 따라서 본 논문에서 구현된 고정소수점 연산 프로그램을 고정소수점 연산을 위한 DSP나 DoC 형태로 그대로 구현이 가능할 것이다.

6. 결 론

본 논문에서는 GMM 기반의 음성/음악 판별 방법을 응용하여 라디오 방송에서 순수한 음악 구간만을 판별하고 이를 선택적으로 저장할 수 있는 시스템을 개발하였다. 라디오 방송의 경우에는 여러 가지 다양한 상황이 존재할 수 있기 때문에 일반적인 음성/음악 판별 알고리즘보다 훨씬 복잡한 문제를 가지고 있다. 따라서 라디오 환경에 적합하게 음악 판별 알고리듬을 구현하였으며, 후처리 과정을 통해 보다 정교하게 음악의 시작 부분과 끝 부분을 검출할 수 있도록 하였다.

구현한 시스템을 16시간 분량(83곡)의 FM 라디오 방송으로 테스트한 결과 순수한 음악 한 곡을 정확히 저장하는 경우와 음악의 앞이나 뒷부분에 비음악이 어느 정도 포함되는 경우가 89.2%(74곡), 음악의 앞부분이나 뒷부분에서 수 초~수 십초 정도 잘리는 경우가 10.8%(9곡)로 나타났다. 광고음악, 로고송, DJ 멘트 등의 순수한 음악이 아닌 사운드를 저장하는 경우는 전혀 없었으며, 순수한 음악을 저장하지 못하는 경우도 없었다. 하지만 DJ 멘트의 특성에 따라 비음악이 일부 포함되는 성능 차이가 나타나는 것을 확인하였다. 특히 클래식/가곡의 경우 음악이 과도하게 잘리는 경우가 발생하였고, 음악이 2~3곡 이어서 저장되는 경우가 발생하였는데, 이에 대한 성능 보완이 필요한 것으로 판단된다.

본 논문에서는 후처리 과정에서 음악적 장르의 특성에 따라 가요/팝송, 클래식/가곡의 두 가지로 구분하여 튜닝을 하였다. 실제 응용 시스템 구현을 위하여 기존의 장르 판별(genre classification) 알고리듬에서 가요/팝송 음악과 클래식/가곡 음악을 구분할 수 있는 특징 벡터를 찾아내어 두 가지 장르를 자동으로 분류할 수 있는 연구가 보완된다면 더욱 성능을 높일 수 있을 것이다.

실시간 구현을 위해 고정소수점 연산을 수행한 결과 2.88MIPS의 매우 낮은 연산량으로 부동 소수점 연산 결과와 동일한 결과를 얻을 수 있었다. 이 과정에서 특징 벡터 추출 과정의 연산량이 전체 연산량의 73%를 차지할 정도로 중요한 요소임을 알 수 있었다. 본 시스템에 음악을 MP3 포맷으로 압축하여 플래쉬(flash) 메모리 등에 저장하는 기술을 추가하고 DSP나 SoC 형태로 구현한다면 다양한 응용분야에 적용할 수 있을 것이다.

참 고 문 헌

[1] W. Q. Wang, W. Gao, D. W. Ying, "A Fast and Robust Speech/Music Discrimination Approach," *IEEE ICICS-PCM,* pp.1325-1329, Dec., 2003.

[2] Lie Lu, Hong-Jiang Zhang, Hao Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing*, Vol.10, No.7, pp.504-516, Oct., 2002.

[3] K. El-Maleh, M. Klein, G. Petrucci, P. Kabal, "Speech/Music Discrimination for Multimedia Application", *Proc. IEEE Int. Acoustics, Speech, Signal Processing, Istanbul*, pp.2445-2448, Jun., 2000.

[4] 정기훈, 이봉진, 강현우, 강홍구, "음악/음성 판별 시스템의 특성 파라미터 조합에 따른 성능 분석," 음향학회 추계학술발표대회논문집, 제25권, 제2(s)호, pp.247-250, 2006년 11월.

[5] Thom F. Quatieri, *Discrete-Time Speech Signal Processing Principles and Practice*, Prentice Hall PTR, 2002.

[6] Rongqing Huang, "Advances in Unsupervised Audio Classification and Segmentation for the Broadcast News and NGSW Corpora," *IEEE Transactions on Speech and Audio Processing*, Vol.14, No.3, pp.907-919, May, 2006.

[7] Reynolds, D. A. Rose, R. C, "Robust Text-independent Speaker Identification using Gaussian Mixture Speaker Models", *IEEE Transaction on Speech and Audio Processing*, Vol.3, No.1, pp.72-83, Jan., 1995.

[8] 신옥근, "화자독립 음성인식을 위한 GMM 기반 화자 정규화," 정보처리학회논문지, Vol.12-B, No.4, pp.437-442, Aug., 2005.

[9] Costas Panagiotakis and George Tziritas, "A Speech/Music Discriminator based on RMS and Zero-Crossings," *IEEE Transactions on Multimedia*, Vol.7, No.1, pp.155-166, Feb., 2005.

[10] ITU-T Software Tool Library 2000, STL-2000. Release 3 version, http://www.itu.int/rec/T-REC-G.191-200012-S/en.

강 현 우

e-mail : hwkang@kangnam.ac.kr

1991년 연세대학교 전자공학과(학사)

1993년 연세대학교 전자공학과(공학석사)

1997년 연세대학교 전자공학과(공학박사)

1997년~1999년 (주)현대전자 멀티미디어 연구소 선임연구원

1999년~현 재 강남대학교 컴퓨터미디어공학부 교수

관심분야 : 적응 신호처리, 음성 신호처리, 멀티미디어 응용 등