Journal of Korea Multimedia Society Vol. 10, No. 3, March 2007(pp. 391-400)

부분 정보에 기반한 효과적인 음악 무드 분류 방법

박근한[†], 박상용[††], 강석중[†††]

요 약

기술의 발전으로 인하여, 대용량의 음악 데이터들을 저장하고 검색하는 것이 중요하게 되었다. 그러나 음악데이터들을 손쉽게 분류하고 검색하기 위한 방법론에 대한 집중적인 연구는 이루어 지지 않고 있다. 본 논문에서는 내용기반의 음악 분류/검색에 대한 새로운 방법론을 제안한다. 기존의 분류화 (classification) 방법들이 음악파일 전체에 대해서 수행하는데 비해 음악파일의 부분만을 분석하여 비슷한 성능을 낼 수 있다는 것을 보여 주었고, 소리의 톤 (tone) 표현에 기반한 새로운 피쳐를 제안하여 기존의 피쳐들에 비해 효과적으로 분류를 할 수 있다는 것을 보여주었다. 또한 속도향상을 위한 여러가지 방법론들을 적용하여 실 제품 적용 시 보다 효과적인 방법론이 될 수 있음을 보여주었다. 제안한 방법론을 MuSE (Music Search/Classification Engine)엔진으로 구현함으로써 PC와 PDA상에서 잘 동작함을 보여주었다.

Effective Mood Classification Method based on Music Segments

Gunhan Park[†], Sang-Yong Park[††], Seok-Joong Kang[†††]

ABSTRACT

According to the recent advances in multimedia computing, storage and searching technology have made large volume of music contents become prevalent. Also there has been increasing needs for the study on efficient categorization and searching technique for music contents management. In this paper, a new classifying method using the local information of music content and music tone feature is proposed. While the conventional classifying algorithms are based on entire information of music content, the algorithm proposed in this paper focuses on only the specific local information, which can drastically reduce the computing time without losing classifying accuracy. In order to improve the classifying accuracy, it uses a new classification feature based on music tone. The proposed method has been implemented as a part of MuSE (Music Search/Classification Engine) which was installed on various systems including commercial PDAs and PCs.

Key words: Music Classification(음악분류), Mood Search(무드검색), Contents Analysis(내용분석), Bark-Scale Frequency Cepstral Coefficients(바크 캡스트럴 계수), Support Vector Machines(서포트 벡터 머신)

1. 서 론

기술의 발전과 더불어, 오디오 데이터들이 증가함에 따라 다양한 오디오 활용에 대한 요구들이 증가되어 왔다. 그러나 기존의 방법들은 텍스트기반의 오디오 정보를 이용함으로써, 근본적인 두가지의 문제점을 가지고 있다. 텍스트 데이터에 대한 효과적인 검색능력에도 불구하고, 실제 텍스트 데이터를

※ 교신저자(Corresponding Author) : 강석중, 주소 : 서울시 노원구 월계동 447-1 비마관 515-1호(139-701), 전화 : 02)940-5761, FAX : 02)941-5730, E-mail : sjkang@cs.kw.ac.kr
접수일 : 2006년 10월 11일, 완료일 : 2007년 1월 22일

[†] 삼성전자 DM 연구소
(E-mail : gunhan.park@gmail.com)
[††] 삼성전자 DM 연구소
(E-mail : sy302.park@samsung.com)
[†††] 정회원, 광운대학교 컴퓨터과학과

수동으로 작성하는 것은 거의 불가능한 일로서 대용량의 오디오 데이터에의 적용에는 한계가 있다. 또한 텍스트를 모두 작성한다고 해도, 작성자마다 각기 다른 형태의 텍스트로 표현할 수 있기 때문에 동일성을 유지하기가 쉽지 않다. 이와 더불어 텍스트로는 표현할 수 없는 음악의 내부 특성들이 있다. 즉 ID3 태그와 같은 텍스트 정보를 이용하여 미리 아티스트, 발표년도, 앨범 등을 알 수 있다고 해도 그 음악의 느낌 (emotion)등을 알기는 불가능하다. 이에 본 연구에서는 오디오 데이터를 효과적으로 관리/활용하기 위한 새로운 음악 자동 분류기법을 제안한다.

음악 자동분류에 대한 연구들은 크게 두 가지로 나눌 수 있다. 먼저 첫번째는 MIDI 파일 등의 음계를 가지는 음악데이터를 기반으로 하는 분류 기법들과 두번째는 MP3 파일등의 오디오 시그널을 가지는 음악데이터를 기반으로 하는 분류기법으로 나눌 수 있다. MIDI파일로 구성되어 있는 파일에 대한 분석기법들은 그 명료함으로 인해 많은 장점을 가지고 있으나, 실제 데이터들이 MIDI파일로 존재하지 않고 또한 활용도에 있어서 제한점이 있다는 문제점을 가지고 있다. 이에 본 연구에서는 아무런 제약을 가지지 않고, 실제 가장 많이 사용하고 있는 오디오 시그널(예, MP3, OGG)로 되어 있는 음악파일에 대한 자동분류 방법을 제안한다.

본 연구에서 수행한 중요한 내용은 기존 특징값 분석, 새로운 필터인 Bark-Scale 서브 밴드 필터 적용, 무드 분류에 최적인 특징값 선택, SVM (Support Vector Machine) 기반 새로운 분류기, 음악의 부분을 이용한 무드 분류 (Segments of Music), 무드에 적합한 키워드 정의를 통한 검색 (Query by Keyword), 엔진 구현 및 응용 프로그램 구현 (PC, PDA 기반) 이다. 이와 같은 부분에 대한 다양한 실험과 테스트를 통해 새로운 알고리즘을 제안하였다.

본 논문의 구조는 다음과 같다. 먼저 2장에서는 음악자동분류와 관련된 이전의 연구들에 대해서 살펴보고, 3장에서는 본 연구에서 사용되는 특징값들에 대해서 자세히 설명하고, 4장에서는 제안한 무드 분류 방법에 대해서 살펴본다. 5장에서는 제안한 방법의 성능 측정을 위한 실험과 결과에 대한 분석을 기술하고, 마지막으로 6장에서는 결론을 내린다.

2. 관련연구 (Related Work)

음악 분류는 사람에게나 컴퓨터에 있어서 매우 어려운 작업이다. 어떤 범주에 대한 음악적 특성이라는 것은 정확하고 분명하게 설명하기가 쉽지 않다. 여러 가지 관점에서 음악의 무드라는 것은 매우 주관적일 수 있는데 이것은 문화, 교육, 경험과 같은 많은 요소들에 의존적일 수 밖에 없다. Hevner의 모델[1]과 같이 이전 인지과학적/심리학적 측면에서의 분류에 대한 연구가 진행되었으나, 이러한 분류들은 계산적인 모델링을 하기 매우 힘들고, 그 기술(description)이 매우 애매하다.

이와 같은 불명확한 부분이 있음에도 불구하고, 자동 음악 분류는 사람보다 더 빠르고 일관성있게 음악을 분류할 수 있다는 장점도 가지고 있다. 즉 컴퓨터는 실험적 결과에 영향을 줄수 있는 사람의 선호도/선입견/편견 등을 제거할 수 있다. 이러한 관점하에서의 무드 모델링을 통한 음악 무드 자동 검출 방법들이 많이 연구 되고 있다.

Tzanetakis의 연구[2]에서는 음색 (Timbre) 특징값, 리듬 (Rhythm) 특징값, 하모닉 (Harmonic) 특징값을 사용하고 있으며, 분류기 (Classifier) 로는 K-NN, Gaussian Mixture 분류방법을 사용하였다. Li의 연구[3]에서는 이러한 특징값들 이외에 새로운 값으로 DWT (Discrete Wavelet Coefficients) 특징값을 사용하였고, Tzanetakis의 특징값/분류기와의 비교 분석연구를 수행하였다. 또한 Beranzweig의 연구에서는 분류기를 뉴런망 (neural network)를 사용해서 12개의 클래스를 구분하는 연구를 수행하였다[4]. McKay의 연구에서는 심볼릭 데이터 (symbolic data)에 대해서 109개의 상위레벨 특징값 (high-level feature)들을 구현하고 있으며, MIDI파일을 이용하여 다양한 종류의 장르를 구분하고 있다[5]. 최근들어 장르뿐만 아니라 무드등의 음악내의 감성을 기반으로 한 구분방법들이 연구되어 발표되고 있다[6].

Pachet[7]의 연구에서는 현재 진행중인 음악 장르 분류 방법에 대한 여덟가지 방법들에 대한 장단점을 분석하였다. 이 방법들은 공통적으로 2단계의 방식을 취하고 있다. 먼저 음악들을 프레임 (frame)으로 불리는 작은 시간 단위로 나누고, 각 프레임마다 음색, 리듬등의 특징값 벡터를 계산한다. 다음으로 기계 학습/분류 방법을 사용하여 감독 학습 (supervised

learning)을 사용하여 장르를 구분하고 있다.

또 다른 특징값으로 비트, 리듬에 대한 연구들이 있다. 음악에 있어서 비트, 리듬이 중요한 요소임에 틀림없지만, 실제 음악 파일에서 정확하게 구해내기가 쉽지 않다. Tzanetakis의 연구에서 비트 히스토그램 (Beat Histogram)을 사용하고 있고, Foote의 연구에서 비트 스펙트럼 (Beat Spectrum)을 이용하여 리듬특징 및 템포값을 나타내고 있다[8]. 비트 히스토그램과 비트 스택트럼은 같은 특징을 표현하는 방식으로, 특정한 비트에서 상대적인 크기와 템포를 나타내줄 수 있다. 그러나 현재로서는 성능이 좋지 못한 상태이며 많은 개선이 요구되고 있다.

또한 몇가지의 연구들에서는 MP3파일에서 빠른 특징값 추출을 위해서 전체 디코딩을 하지 않고 비트스트림 파싱과 역양자화 (de-quantization)만을 수행하는 부분 디코딩만으로 특징값들을 추출하고 있는데, 대표적인 연구로 Pye의 MP3CEP 방법론을 들 수 있다. 이는 비슷한 성능에 4.5배의 속도향상을 가져옴을 보이고 있다[9,10].

이와 같은 이전연구들에서 사용된 특징값들을 분석해 보면, 크게 3가지 영역으로 나눌 수 있는데 스펙트럴 방법 (Spectral Method), 시간적 방법 (Temporal Method), 캡스트럴 방법 (Cepstral Method)으로 나눌 수 있다. 스펙트럴 방법은 스펙트럴 중심 (Spectral Centroid), 스펙트럴 플럭스 (Spectral Flux)와 같은 특징값들을, 시간적 방법은 제로 교차율 (Zero Crossing Rate)과 같은 특징값을, 캡스트럴 방법은 MFCC (Mel-Frequency Cepstral Coefficients), LPC (Linear Prediction Coding) 캡스트럼 (cepstrum) 등을 들 수 있다[11,12]. 이러한 음악 자동 분류에 대한 기존연구들은 기본적으로 음성인식 (Automatic Speech Recognition) 분야의 기술들을 이용하고 있다. 이는 오디오 자체의 정보분석에 대한 연구 보다는 음성인식 분야에 대한 분석 연구들이 집중적으로 수행되었기 때문이다.

본 연구에서는 이러한 기존연구들을 분석하여 보다 성능이 좋은 새로운 방법론을 제안한다. 제안 하는 방법론은 전체 곡의 통계치를 이용하지 않고 부분 만으로 성능을 비슷하게 유지 함으로써 추출 시간을 획기적으로 줄였으며, 기존의 특징값들에 비해 성능이 좋은 새로운 특징값들을 제안하였다. 또한 커널 기반 기계학습 방법인 SVM (Support Vector Machine)을 이용하여 보다 분류 정확도를 높일 수 있었다. 이에 대해서는 다음 절에서 자세하게 설명한다.

3. 특징값 (Features)

본 장에서는 다양한 특징값들중 본 연구에서 사용하는 특징값들에 대해서 설명한다. 앞서 살펴본 바와 같이 특징값들은 크게 나누어보면 로우레벨 특징값 (low-level feature)와 하이레벨 특징값 (high-level feature)의 두가지 분류로 나눌 수 있다. 로우레벨 특징값으로는 신호처리특성을 많이 가지고 있는 스펙트럴 중심, 스펙트럴 플럭스등을 들 수 있고, 하이레벨 특징값으로는 템포 (tempo)등의 음악적 요약특성 (musical abstraction)들이 있다. 이러한 특징값들은 자동분석기술의 한계로 실제 음악의 의미론적 특성과 정확히 일치하지는 않지만, 음악이 가지는 기본적인 특성들을 잘 반영해 주고 있으며, 현재 많은 연구들에서 장르나 무드 분류/검색에 이용되고 있다. 이와 같은 많은 특징값들중 본 연구에서는 다음에서 설명할 다섯종류의 특징값을 사용하였다.

3.1 스펙트럴 중심 (Spectral Centroid)

스펙트럴 중심은 주파수 대역에서 에너지 분포의 평균 지점이다. 이 특징값은 음정에 대한 인지 척도로 사용된다. 즉, 음의 높낮이에 대한 주파수 내용을 판단하는 척도이다. 스펙트럴 중심은 신호 에너지의 대부분이 집중하는 주파수 영역을 결정하며, 식 (1)과 같이 계산된다[2].

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] \times n}{\sum_{n=1}^{N} M_t[n]} \quad (1)$$

$M_t[n]$은 프레임 t와 주파수 범위에 따른 영역 n에서 퓨리에 변환의 크기 (magnitude)를 나타낸다.

3.2 스펙트럴 롤오프 (Spectral RollOff)

스펙트럴 롤오프 지점은 주파수 대역에서 에너지의 85%가 어디에서 얻어지는가를 결정한다. 이 특징값은 스펙트럴 모양을 측정하는데, 음정의 분포 정도를 나타낼 수 있기 때문에 서로 다른 음악을 구분하는데에 유용하게 사용할 수 있다. 음악의 경우 그 노래특성에

따라 주파수 대역의 전 범위에 걸쳐 더 잘 분포되어 있거나 모여 있을 수 있는데 이를 구분할 수 있게 된다. 스펙트럴 롤오프 지점은 식 (2)와 같이 계산된다.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 \times \sum_{n=1}^{N} M_t[n] \quad (2)$$

스펙트럴 롤오프 주파수 R_t는 크기 (magnitude)분포의 85%인 지점의 주파수로 정의된다.

3.3 스펙트럴 플럭스 (Spectral Flux)

스펙트럴 플럭스는 2개의 연속하는 주파수 대역의 에너지 분포의 변화를 나타낸다. 음악의 특성에 따라 에너지 분포의 변화가 크거나 작을수 있으므로 이러한 변화를 이용하여 각 음악을 구분하는 특징으로 사용하게 된다. 연속되는 스펙트럴 분포의 정규화된 (normalized) 크기사이의 차이값의 제곱으로 정의되며, 식 (3)과 같이 계산된다.

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \quad (3)$$

$N_t[n]$은 프레임 t에서의 퓨리에 변환의 정규화된 크기를 나타낸다.

3.4 BFCC (Bark-Scale Frequency Cepstral Coefficients)

음성인식에서 사용되는 많은 특징값들이 내용기반 음악 분석에도 사용되고 있는데 그 대표적인 예가 캡스트럼 (Cepstrum) 분석 방법이다. 캡스트럼 분석 방법은 일반적으로 목소리의 특성에 따른 음색 차이를 잘 나타내 준다고 알려져 있으며, 음악에도 적용되어 좋은 성능을 내고 있음이 알려져 있다[11].

캡스트럼 분석은 일반적으로 다음과 같은 순서를 통하여 계산되게 된다. 먼저 PCM 데이터 값을 일정한 크기의 프레임으로 나누고 해밍 윈도우 (Hamming window)를 통하여 엣지효과 (edge effect)를 제거한다. 각 프레임의 값을 퓨리에 변환을 통하여 주파수 도메인으로 변환한다. 퓨리에 변환의 크기 (magnitude)를 구하고 이 값들에 대해 청각특성적 필터 뱅크 (filter banks)를 이용하여 주파수 분해를 한다. 주파수 분해후 로그 (log)를 취하여 값을 구하고, 이 값들의 상호관계성(correlation)을 제거하기 위해 DCT (Discrete Cosine Transform)을 수행한다. 이러한 일련의 과정을 통하여 캡스트럴 계수 (Cepstral Coefficients)를 구하게 된다.

Bark Frequency Cepstral Coefficients (BFCC)는 이러한 캡스트럼 특징을 이용하는 방법으로, 비균일 필터 뱅크 (non-uniform filter banks)중에서 발성 (speech articulation)에 똑같은 기여를 하는 밴드 (band)로 구분 하는 크리티컬 밴드 스케일 뱅크 (critical band scale filter banks)중에서 톤 인식 (tone perception)을 주파수에 적용한 방법이다. 이와 같이 Bark-Scale은 톤을 기반으로 하기 대문에 주관적 피치 구분등에 사용되는 다른 스케일 필터들보다 음악 분석에 적합하다. 톤 (tone)은 기본적으로 음색 (timbre)를 나타내는 것으로 목소리/악기등을 구분짓게 하는 소리의 중요한 요소이다.

Bark-scale 필터는 기본적으로 인간의 가청범위를 약 24개의 밴드로 나눈다. 특정대역 (예, 1000Hz) 이하에서는 linear하게 증가하다가, 특정대역 이상에서는 로그함수 (logarithmic)로 증가한다. 본 연구에서 사용한 Bark-scale은 24 밴드, 최대 15,500 Hz의 범위를 가지고 있다[13]

3.5 BFCC 차이값 (Delta of BFCC)

차이값 (Delta)은 많은 특성분석들에서 사용되는 계산값이다. 본 연구에서는 음색을 나타내는 BFCC의 각 계수들의 차이값을 이용하여 음색의 변화정도가 어느 정도 인지 나타낸다. 이에 대한 계산식은 다음과 같다.

$$\delta_i = BFCC_i - BFCC_{i-1} \quad (4)$$

$BFCC_i$는 i번째 BFCC 계수값을 의미한다.

4. 무드 검출 (Mood Detection)

본 장에서는 내용기반 음악 검색/분류에 대한 전체적인 프로세스에 대해서 설명한다. 그림 1에서 보는 바와 같이 전체 프로세스는 두개의 부분으로 나눌 수 있는데, 첫번째는 특징값을 추출하고 분류하는 부분이고, 두번째는 추출된 특징값 및 분류결과를 이용하여 사용자의 요구에 맞는 음악파일을 찾아주는 검색 부분이다.

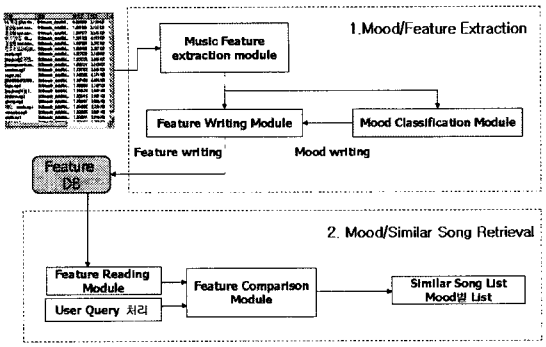


그림 1. MuSE 전체 프로세스

이러한 전체 구조에서 가장 중요한 부분인 특징값 추출 및 분류 방법에 대하여 자세하게 설명한다. 특징값 추출 및 분류에 대한 전체적인 순서도는 그림 2에 설명되어 있다. 먼저 인코딩된 음악파일을 디코딩하고 정규화 하는 전처리 과정이 있고, 이러한 과정을 통해 정규화 된 값들을 본 논문에서 제안하는 특징값 추출 방법에 따라 다섯 가지의 특징에 대해 계산을 수행하는 추출 과정이 있다. 특징값 추출을 통해 구해진 값들은 미리 만들어져 있는 분류기 모델을 통하여 어떤 무드를 가지는지 분류되게 된다. 본 연구에서는 무드를 4가지 (차분한, 감미로운, 신나는, 정열적인)로 나누고 이에 대한 분류기를 구현하였다.

4.1 데이터 전처리 (preprocessing)

특징값 추출을 하기전에 여러가지 압축포맷과 샘플링 특성등에 대한 영향을 제거 하기 위하여 몇가지 기본적인 데이터 전처리 과정을 수행한다.

- 샘플링율 변환 (Sampling Rate Conversion) : 모든 음악파일은 특정한 샘플링율을 갖도록 변환된다. 이러한 변환을 하는 이유는 크게 두가지인데 먼저 샘플링율이 특징값에 영향을 주기 때문이다. 두번째는 음악파일에 유용한 정보의 대부분이 저주파수 대역에 있다는것이다. 그러므로 다운샘플링을 하는 것은 특성값을 구하는 시간을 줄여줄 수 있을 것이다.
- 정규화 (Normalization) : 샘플링된 수치값을 정규화 하는 것은 소리크기 (loudness)등의 영향을 최소화 하는데 매우 중요하다.
- 채널병합 (Channel Merging) : 스테레오 (혹은 다채널)로 녹화된 음악은 모노로 바꾸어서 특징값을 계산한다. 일정한 특징값을 얻을 수 있고 계산시간을 줄일 수 있다.
- 구간선정 (Windowing) : 특성값 분석을 위한 최소의 단위의 구간을 선정하고, 분석구간을 설정한다. 이 분석구간은 오버랩을 포함하여 연속적인 특성들에 대한 계산을 수행한다.

본 연구에서는 22050 hz, Mono, PCM값을 정규화 포맷으로 사용하고 있다. 따라서 어떠한 형태의 음악파일이 입력되어도 본 정규화 과정을 통하여 항상 같은 특징값을 구할 수 있게 된다.

4.2 특징값 추출 알고리즘 (Feature Extraction Algorithm)

본 절에서는 음악분류에서 사용하는 특징값들을 어떻게 추출하는지 자세하게 설명한다. 기본적으로 특징값은 512 샘플의 크기를 가지는 기본 단위로 처리 된다. 이 기본 단위는 그림 3에서 보는 바와 같이 분석윈도우 (analysis window)라고 부른다. 22050 Hz의 정규화된 데이터를 사용하므로, 이 값은 대략 23msec정도의 시간 단위를 가지는 크기이다. 이 단위들에 대해 단시간 퓨리에 변환 (Short Time Fourier Transform)을 통하여 음악파일의 특징값들을 계산하게 된다.

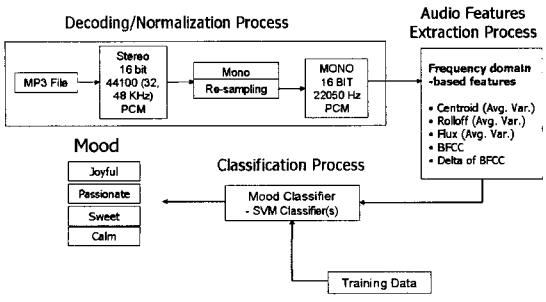


그림 2. MuSE 특징값 추출 프로세스

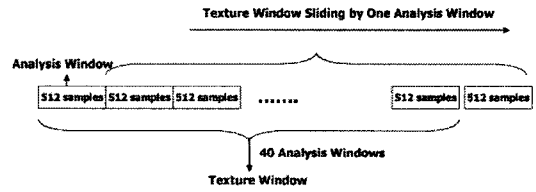


그림 3. 기본 처리/분석 방법

기본단위들은 다수개 처리되어 그 분산값과 평균값을 계산하게 되는데, 본 연구에서는 40개의 단위를 하나의 텍스처윈도우 (texture window)로 처리한다. 이 텍스처윈도우는 오버랩을 포함하여 하나의 분석 윈도우 단위로 옮겨가며 값들을 계산하게 된다. 이러한 값들을 전체적으로 평균하여 최종적인 특징값을 계산하게 된다. 이에 대한 설명이 그림 3에 나타나 있다. 이러한 기본처리단위들을 결정하는 것은 계산량과 성능에 관계가 있으며, 본 연구에서는 기존의 연구들과 다양한 테스트를 통해 그 값들을 정하였다.

기본적인 특징값들은 분석윈도우마다 구하게 된다. 그림 4는 특징값들을 어떤 순서로 추출하는지와 내부적으로 어떤 작업들이 일어나는지를 자세히 설명한 모듈 다이어그램이다. 먼저 특징값 추출을 위한 기본적인 테이블값을 메모리에 저장하고, 분석윈도우에 포함되어 있는 PCM에 대해 해밍 윈도우를 통해 값을 변환시킨다. 이 변환된 값을 이용하여 주파수 대역으로의 퓨리에 변환을 수행하고, 그 크기값을 구한다. 기본적으로 이 크기값을 이용하여 스펙트럴 값들을 계산하고, 같은 크기값을 바탕으로 Bark-Scale 필터를 통과시킨뒤 로그값과 상호관계성을 제거한뒤 특징값들을 추출한다. 분석윈도우 마다 추출된 특징값들은 텍스처 윈도우 단위로 평균과 분산값을 구하고, 구해진 각각의 값들에 대해 전체 분석 음악 구간에 대하여 평균을 구하여 하나의 곡에 대한 특징값으로 결정된다. 이 특징값들은 미리 정의된 분류모델을 통해 무드를 분류하고, 검색에 사용하기 위해 그 값들을 본 연구에서 지정한 데이터 포맷으로 저장한다.

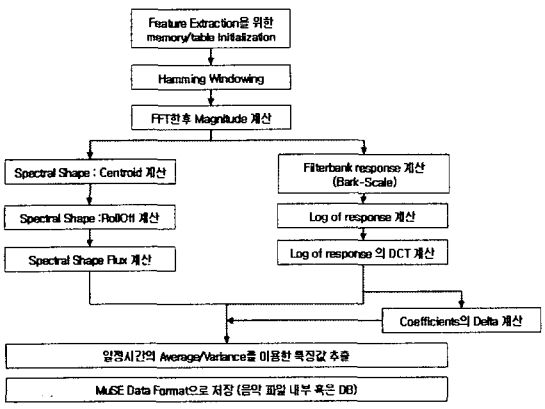


그림 4. 특징값 추출 모듈

4.3 SVM (Support Vector Machine) 분석

본 절에서는 음악무드를 결정하기 위해 사용하는 기계학습방법에 대해서 설명한다. 앞 절에서 설명한 특징값들은 음악의 평균 에너지, 에너지가 모여있는 지점, 에너지 분포의 변화, 음색구분 필터뱅크에 대한 계수값, 음색구분 필터 뱅크에 대한 계수값의 변화들을 설명하고 있고, 이것은 음악에 있어서 대략적인 음진행의 변화, 높낮이와 사용된 악기의 음색등을 표현할 수 있다. 그러나 음악의 무드를 단순히 음이 높고 낮음과 같은 방식으로 결정할 수 없으므로, 결정트리 (decision tree)와 같이 범위기반 분기방식으로는 음악 무드를 결정하기 어렵다. 음악의 무드란 것은 사람이 느끼는 방식에 의해 분류가 되어야 하고, 이를 위해서는 다양한 종류의 음악들을 훈련데이터 (training data)로 이용한 기계학습 (machine learning)방법으로 그 경계선을 구하는 것이 최선의 방법이다. 본 논문에서는 이 분류 경계선를 정하기 위하여 최적의 분류 평면 (classification plane)을 구하는 방식중 좋은 성능을 내고있는 SVM (Support Vector Machines)을 이용한다.

SVM은 커널 기반의 기계학습방법으로 감독 학습의 한가지 방법이다. 간단한 수식만을 가지고서도 복잡한 패턴인식 문제를 쉽게 해결할 수 있는 명료한 이론적 근거에 기반하고 있으며, 입력으로부터 어떠한 학습방법을 이용하는가에 대한 직관적인 해석을 제공 해준다. 실제 응용에서 복잡한 구조를 가지는 패턴의 분류를 위해 SVM 기법은 입력 공간인 높은 차수의 비선형 특징 벡터공간을 선형적으로 투영하여 해석할 수 있도록 해주고 각 특징 벡터 사이의 최적의 경계 분리면 (maximum margin hyperplane)를 제시한다. 최근 연구들에서 얼굴인식, 영상분류등의 패턴인식 분야에서 실험적으로 Nearest Neighbor (NN)이나 Gaussian Mixture Model (GMM), Hidden Markov Model (HMM) 등보다 좋은 결과를 보이고 있음을 보여주고 있다.

SVM은 다음과 같은 방법으로 구현된다. 여기에 설명하는 방법은 일대일 분류방법에 대한 설명으로, 멀티클래스 분류기를 위해서는 이러한 일대일 분류기를 여러개 구성하여 구현하게 된다. 먼저 양성 (positive)과 음성 (negative) 특성의 두 개의 클래스 속하는 훈련데이터를 다음 식과 같이 정의한다.

$$(\mathbf{x}_1, y_1),...,(\mathbf{x}_k, y_k),\ \mathbf{x}_i \in R^n,\ y_i \in +1,-1 \quad (5)$$

$\mathbf{x}_i$는 i번째 샘플의 n차원의 특징값 벡터를 나타낸다. 본 연구에서는 앞 절에서 설명한 스펙트럴 중심, 스펙트럴 롤오프, 스펙트럴 플럭스, BFCC, BFCC의 차이값을 $\mathbf{x}_i$로 사용한다. y_i는 i번째 데이터의 클래스 라벨을 나타내며, 기본적인 SVM프레임워크에서 양성 특성의 데이터와 음성 특성의 데이터를 다음과 같은 하이퍼플레인으로 분리한다.

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0,\ \mathbf{w}, \mathbf{x} \in R^n, b \in R \quad (6)$$

SVM은 훈련데이터들을 이러한 두 개의 클래스들로 정밀하게 나누는 "최적의" 하이퍼플레인을 찾는다. 최적의 하이퍼플레인을 찾는다는 것은 다음과 같은 동일한 최적 문제를 푸는 것과 같은 문제가 된다.

$$Minimize\ \Phi(\mathbf{w}) = \frac{1}{2(\mathbf{w} \cdot \mathbf{w})},$$
$$subject\ to\ y_i[(\mathbf{w} \cdot \mathbf{x}_i) - b] \geq 1,\ i = 1,...,k \quad (7)$$

라그랑지 곱셈 방법 (Lagrange Multiplier Method)에 의해 다음과 같은 또 다른 동일한 최적화 문제를 얻는다.

$$Maxmize\ W(\alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j),$$
$$subject\ to\ \alpha_i \geq 0,\ i = 1,...,k,\ and\ \sum_{i=1}^{k} \alpha_i y_i = 0 \quad (8)$$

이 식을 만족시키는 계수를 찾는 것이 SVM에서 구하는 하이퍼플레인이 되고, 이것을 분류기 모델이라고 부른다. 훈련데이터들에 의해 구해진 분류기에 의해 실제 데이터값들을 분류하게 된다. SVM은 위와 같은 선형적인 모델의 내적 ($\mathbf{x}_i \cdot \mathbf{x}_j$)을 대치하여 커널 펑션 $K(\mathbf{x}_i, \mathbf{x}_j)$을 사용할 수 있으며, 어떤 커널을 사용하느냐에 따라 선형 혹은 비선형 모델을 구할 수 있다.

이러한 SVM은 훈련데이터 (training data)의 구성/순서/개수와 분류 벡터의 값의 범위/순서등에 따라 조금씩 다른 특성모델이 생성되고, 결과에 영향을 미친다. 본 연구에서는 다양한 실험을 통하여 무드 분류에 적합한 분류 모델을 생성하였고, 이는 간단한 계산을 통하여 빠르게 음악들을 분류할 수 있음을 보였다. 본 연구에서는 SVM Light 라이브러리[14]를 사용하였다.

4.4 유사도 계산 (Similarity Measures)

본 연구에서 사용하는 음악 특징값들은 벡터의 형태로 표현된다. 따라서 유사한 곡을 선택하기 위한 유사도는 벡터들의 거리값으로 구할 수 있다. 즉 거리가 가까운 벡터들은 비슷한 음악이라고 할 수 있다. 이러한 벡터 거리를 구하는 방법은 시티블럭 거리, 제곱근 거리 등 기본적인 거리 방법부터 EMD (Earth Mover's Distance)[15]와 같은 두 벡터의 성분의 특징 및 전체분포에 대한 성질을 이용한 방법도 사용 할 수 있다. 본 연구에서는 기본적인 벡터 거리 계산방법 중 간단하면서 상대적으로 좋은 성능을 나타내고 있는 시티블럭 거리 값을 사용한다. 이는 차이값에 대한 절대값의 합으로, 계산방법은 식 (10)에 나타나 있다.

$$M = (f_1, f_2, f_3, ..., f_n) \quad (9)$$

$$D(M, M') = \sum_{i=1}^{n} |f_i - f'_i| \quad (10)$$

식 (9)에서 보는 바와 같이 한 곡의 음악 M은 여러 개의 특징값 f_i를 가진 벡터로 표현할 수있고, 이에 대한 유사도는 식 (10)에서 보는 바와 같이 간단한 형태의 벡터 거리값으로 구할 수 있다.

표 1. MuSE 성능

성능 관련 사항	Extraction/Classification	Similar Search	Mood Search
Time (P4, 3.0Ghz, 512M RAM)	0.13 sec/song	0.42 sec/1000 songs	0.41 sec/1000 songs
Execution Code Size	196 KB	80 KB	80 KB
Memory	430 KB	15 KB	10 KB
CE Device	ARM9 500 MHz 6 sec/song	ARM9 200 MHz 0.07 sec/1000 songs (using DB)	ARM 9 200 MHz 0.06 sec /1000 songs (using DB)

5. 실험 결과 (Experimental Results)

5.1 실험 (Experiments)

본 절에서는 본 연구에서 제안한 방법론의 검색/분류 정확도 측면의 결과와 이를 구현한 MuSE (Music Search/Classification Engine)엔진에 대한 성능적 측면의 결과에 대해서 설명한다. 먼저 표 1은 전체적인 성능에 대해서 설명하고 있다. 전체 성능은 크게 두가지의 측면, 즉 특징값을 추출하고 분류하는 것과 이를 이용하여 검색하는 것에 대해서 나누어 생각할수 있다. 표에서 보는 바와 같이 구현한 MuSE는 작은 실행 메모리를 가지고 충분히 빠른 시간에 특징값 추출과 검색이 가능하다.

표 2는 비슷한 음악 검색 성능을 나타내고 있다. 비슷한 음악 검색에 대한 테스트는 전체 353곡에 대해서 미리 비슷한 음악들을 정의해 놓고 실제 검색을 했을때 그 음악들이 상위에 얼마나 나타나느냐를 테스트한 것이다. 이는 MP3 플레이어와 같은 작은 디스플레이상에서 적어도 하나는 원하는 곡이 검색되었을 확율을 검색 정확도로 계산한 것이다. 표 2에서 보는 바와 같이 상위 5개 안에 하나라도 있을 확율은 정답 348개로 98.58%를 나타내고 있으며, 상위 3위 안에 나타날 확율은 정답 343개로 97.17%를 나타내고 있다. 이는 원하는 범위안에서 좋은 성능을 나타내고 있음을 알 수 있다.

표 3은 무드 분류에 대한 검색 정확도를 나타내고 있다. 테스트는 3명의 다른 성향의 사용자에 대해서 만족도 테스트를 수행하였으며, 4가지 무드 (차분한, 감미로운, 신나는, 정열적인)에 대해서 테스트 하였다. 표 3에서 보는 바와 같이 각각 82.3%, 66.7%, 92.7%의 정확도(만족도)를 나타내었으며, 전체적으로 1753곡에 대해 83.2%의 정확도를 나타내었다. 표 3에서 보는 바와 같이 사용자 마다 편차가 나타났으며, 이는 개인의 곡에 대한 느낌이 다를 수 있다는 것을 나타내고 있다. 공통적으로 만족도가 떨어지는 경우 이었으며, 이러한 경우를 본 연구에서 제안한 경우는 전혀 반대의 곡이 반대의 무드로 분류 되었을 특징값 이외의 다른 특징값 (예를 들어 템포와 같은 빠르기 정보)을 이용하여 향후 개선시킬 수 있을 것이라 생각된다.

표 2. 비슷한 음악 검색 성능 (353곡 대상 테스트)

성능 분류	정답곡수	정확도
Top 5에 하나라도 있을 확률	348	98.58%
Top 3에 하나라도 있을 확률	343	97.17%

표 3. 무드 분류 정확도

검사자	전체 곡수	정답 곡수	정확도
사용자 1	1418	1167	82.3%
사용자 2	75	50	66.7%
사용자 3	260	241	92.7%
전 체	1753	1458	83.2%

그림 5는 Tzanetakis의 방법[2]과 본 연구에서 제안하는 방법의 성능 비교 및 전체 곡을 모두 이용하는 방법과 부분 만을 이용하는 방법의 성능비교를 나타내고 있다.

성능비교는 Tzanetakis의 논문에서와 같이 장르 구분 방법을 이용하였다. 그림에서 보는 바와 같이 제안하는 방법은 전체 곡을 이용하는 Tzanetakis방법과 비슷한 성능을 내고 있음을 알 수 있고, 부분만을 이용하는 Tzanetakis방법에 비해서는 6%이상의 성능향상이 있음을 알 수 있다. 본 연구에서 제안하는 방법은 기존의 전체 곡을 이용하는 방법에 비해 평균 24배 이상의 빠른 특징값 추출 속도를 가지며, 성능은 4%정도 낮아짐을 알 수 있다. 그러나 부분만을 사용하는 기존의 방법에 비해서는 6%의 성능 향상을 가져왔음을 알 수 있다. 이는 내용기반 음악 검색의 실제 적용을 위해 가장 필요한 요소인 추출 시간 단축을 실현하며, 성능도 향상 시켰음을 보여주고 있다.

5.2 구현 (Implementation)

본 연구에서는 제안한 방법론을 실제 제품에 적용하기 위해 엔진을 구현하였으며, 이 엔진을 이용하여 다양한 형태의 응용 프로그램을 구현하였다. 먼저 키

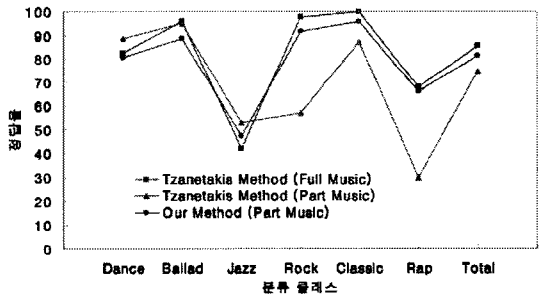


그림 5. 분류 성능 비교

워드 정의를 통한 무드검색을 수행하는 MuSE 플레이어를 구현하였고(그림 6), 같은 성능의 플레이어를 PDA에 구현하였다. 그림에서 보는 바와 같이 프로그램은 무드 입력을 받아, 300개 이상의 키워드의 무드를 매칭하여 그 결과를 랜덤하게 보여줌으로써 각기 다른 결과를 보여준다. 이는 기존의 랜덤플레이에 비해 사용자의 성향을 좀더 반영할 수 있는 무드플레이 기능을 제공하여 준다. 이는 음악의 무드에 대한 정의는 사람마다 다양하게 나올 수 있음을 반영한 것이다. 즉 미리 정의된 네가지의 무드만으로 분류할 수 있으나 다양한 응용을 위하여 무드를 나타내는 단어의 집합을 정의하여 향후 키워드 검색/음성인식 검색등의 관련 기능으로 활용할 수 있을 것이다. 각 단어의 어미 변형를 포함하여 300개 이상의 단어에 대해서 무드를 처리해 줄 수 있으며 계속해서 단어들을 보강하고 있다.

6. 결론 및 향후 연구 (Conclusions and Future Work)

본 연구에서는 음악파일의 내용을 분석하여 효과적인 무드 검색/분류 방법에 대해서 분석하였다. 무드를 4가지 (차분한, 감미로운, 신나는, 정열적인)로 나누고 이에 대한 분류기를 구현하였다. 구현한 분류기는 1753개의 음악 파일 테스트와 세 그룹의 서로 다른 사용자 테스트를 통하여 평균 83.2%의 정답율을 보이고 있음을 알 수 있었다. 또한 부분 음악만으로 효과적인 음악분류/검색이 가능함을 보였으며, 새로운 특징 값들을 제안함으로써 이전 방법보다 평균 6%이상의 성능향상이 있었음을 알 수 있었다. 제안한 알고리듬은 실제 PC/PDA상에서 구현함으로써 실제 적용가능함을 보였으며, 다양한 실험적 방법에 의해 그 효과가 만족스러웠음을 보였다.

그림 6. 제안한 방법론을 이용한 PC응용 프로그램 (MuSE)

개발한 MuSE엔진은 향후 PMP, MP3P, 셋탑박스, DTV등 다양한 제품에, 음악 추천/무드 셔플 플레이/뮤직맵/자동 이퀄라이져 세팅등 다양한 응용으로 활용될 수 있을 것이다.

향후 연구로는 템포등의 새로운 특징값들을 이용하여 분류성능의 만족도를 높이고, 전체 디코딩을 하지 않는 방법을 이용하여 추출 시간을 향상 시키는 방법에 대해서 진행할 계획이다. 또한 허밍질의(Query by Humming)와 같은 기술 개발을 수행할 예정이다.

참 고 문 헌

[1] D. Liu, L. Lu, and H.-J. Zhang, "Automatic Mood Detection from Acoustic Music Data," *Proceedings of the International Symposium on Music Information Retrieval (ISMIR 2003)*, pp. 81-87. 2003.

[2] G. Tzanetakis, "Manipulation, Analysis, and Retrieval Systems for Audio Signals," *Doctoral Dissertation*, Princeton University, 2002.

[3] T. Li, M. Ogihara, and Q. Li, "A Comparative Study on Content-Based Music Genre Classification," *Proceedings of the ACM SIGIR'03*, pp. 282-289, 2003.

[4] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, "A Large-Scale Evaluation of Acoustic and Subjective Music Similarity Measures," *Proceedings of the International Symposium on Music Information Retrieval (ISMIR 2003)*, pp. 99-105, 2003.

[5] C. McKay and I. Fujinaga, "Automatic Genre Classification using Large High-Level Musical Feature Sets," *Proceedings of the International Conference on Music Information*

Retrieval (ISMIR 2004), pp. 525-530, 2004.

[6] T. Li and M. Ogihara, "Content-Based Music Similarity Search and Emotion Detection," *Proceedings of The IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, pp. V705-V708, 2004.

[7] J.-J. Aucouturier and F. Pachet, "Representing Musical Genre: A State of the Art," *Journal of New Music Research*, Vol. 32, No. 1, pp. 83-93, 2003.

[8] J. Foote and S. Uchihashi, "The Beat Spectrum: A New Approach to Rhythm Analysis," *Proceedings of International Conference on Multimedia and Expo (ICME)*, pp. 1088- 1091, 2001.

[9] D. Pye, "Content-Based Methods for The Management of Digital Music," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 2437-2440, 2000.

[10] G. Tzanetakis and P. Cook, "Sound Analysis using MPEG Compressed Audio," *Proceedings of the International Conference on Audio, Speech and Signal Processing (ICASSP)*, pp. 761-764, 2000.

[11] B. Logan and A. Salomon, "A Music Similarity Function based on Signal Analysis," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2001)*, pp. 745-748, 2001.

[12] B. Logan and A. Salomon, "A Content-Based Music Similarity Function," *Technical Report*, Compaq Cambridge Research Laboratory, June, 2001.

[13] L. Rabiner and B,-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall Pub., pp. 183-190, 1993.

[14] T. Joachims, "Making Large-scale Support Vector Machine Learning Practical," *Advances in Kernel Methods - Support Vector Learning*, pp. 169-184, MIT Press, 1999.

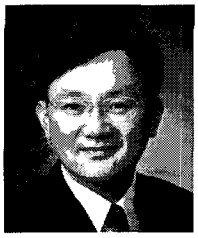
[15] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, Vol. 40, Issue 2, pp. 99-121, 2000.

박 근 한

1994년 한국과학기술원 전산학과 (공학사)
1996년 한국과학기술원 전산학과 (공학석사)
2004년 한국과학기술원 전자전산학과 전산학전공 (공학박사)
2001년~2004년 서치솔루션 연구원
2004년~2006년 삼성전자 DM연구소 책임연구원
관심분야 : 음악 무드 분류/검색, 내용기반 이미지 검색, 멀티미디어 정보 검색/데이터 마이닝

박 상 용

2005년 국민대학교 컴퓨터과학과 (이학사)
2005년~현재 삼성전자 DM연구소 연구원
관심분야 : 내용기반 음악 분류, BD Java, 멀티미디어 프로세싱



강 석 중

1988년 Indian University, Computer Science Dep. (이학사)
1991년 Indian University, Computer Science Dep. (이학석사)
2003년 University of California, Irvine, Electrical Engineering & Computer Science Dep. (공학박사)
1991년~1998년 선임연구원, 한국국방연구원
2003년~2004년 Lecturer 겸 Research Staff, University of California, Irvine, Electrical Engineering & Computer Science Dep.
2004년~2006년 수석연구원, Application SW Lab, Digital Media 연구소, 삼성전자
2006년~현재 조교수, 광운대학교 전자정보공과대학 컴퓨터 과학과
관심분야 : 멀티미디어 시스템, 분산실시간 시스템, 소프트웨어공학