Journal of Korea Multimedia Society Vol. 10, No. 7, July 2007(pp. 817-826)

음악의 클라이맥스 추출을 이용한 내용 기반 장르 분류

정명범[†], 고일주[††]

요 약

기존의 음악 분류 연구는 음악에서 임의 20초 구간 또는 40%~45% 지난 부분으로부터 20초 구간을 얻은 후 여러 가지 신호적 특징을 추출하여 장르 분류에 사용해왔다. 본 논문에서는 기존 연구의 성공률을 높이기 위해 음악의 클라이맥스 구간을 추출하여 장르 분류하는 것을 제안한다. 음악은 도입과 진행, 클라이맥스 부분으로 나뉘며, 클라이맥스는 음악이 강조하는 부분으로서 그 음악의 특징을 가장 잘 나타낸다. 즉, 음악을 분석하거나, 분류할 때 클라이맥스 부분을 이용하면 보다 효과적인 결과를 얻을 것이다. 음악의 클라이맥스는 FFT를 이용하여 박자와 마디 정보를 얻은 후 마디별 파형 집중도로부터 추출할 수 있다. 논문에서는 기존의 연구에 사용된 방법과 제안한 방법인 클라이맥스를 이용하여 장르 분류 실험을 하였다. 기존 방법은 47%의 성공률을 보이는 반면 제안한 방법은 56% 향상된 성공률을 얻을 수 있었다.

Content-Based Genre Classification Using Climax Extraction in Music

Myoung-Bum Chung[†], Il-Ju Ko[††]

ABSTRACT

The existing a music genre classification research used signal feature of the part which gets 20 seconds interval of the random or the 40%~45% after in the music. This paper propose it to increase the accuracy of existing research to classify music genre using climax part in the music. Generally the music is divided to three parts; introduction, progress and climax. And the climax is the part which the music emphasizes and expresses the feature of the music best. So, we can get efficient result if the climax is used, when the music classify. We can get the climax in the music finding the tempo and node which uses FFT and the maximum waveform from each node. In this paper, we did a genre classification experiment which uses existing research method and proposing method. The existing method expressed 47% accuracy. And proposing method expressed 56% accuracy which is improved than existing method.

Key words: Climax of Music(음악의 클라이맥스), Music Retrieval(음악 검색), Music Genre Classification(음악 장르 분류), Audio Signal Processing(오디오 신호 처리)

1. 서 론

최근 컴퓨터 환경의 발달과 폭 넓은 인터넷의 확산으로 사용자들은 보다 풍부하고 깊이 있는 정보를 접할 수 있게 되었으며, 정보의 내용 또한 이미지, 오디오, 비디오 등 다양한 멀티미디어 데이터 형식으로 제공되게 되었다. 따라서 멀티미디어 데이터에 대한 사용자들의 검색 요구도 증가하였으며, 이를 위한 멀티미디어 검색 시스템에 대한 연구가 활발히 진행되고 있다. 그러나 이러한 멀티미디어 데이터들은 데이터가 가지고 있는 특징 그대로 시스템에 색인되어 저장되는 것이 아니라, 기존 분류되어 있는 틀에 맞추어 수작업을 통해 각각의 제목이나, 내용에 맞게 텍스트 기반의 분류 시스템에 저장되어 왔다. 따라서

※ 교신저자(Corresponding Author) : 정명범, 주소 : 서울시 동작구 상도동 숭실대학교(156-743), 전화 : 02)882-1061, FAX : 02)882-0236, E-mail : nzin@ssu.ac.kr
접수일 : 2007년 1월12일, 완료일 : 2007년 5월 2일

[†] 정회원, 숭실대학교 미디어학부
[††] 정회원, 숭실대학교 미디어학부
(E-mail : andy@ssu.ac.kr)
※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

기존 검색 시스템은 사용자에게 맞는 적절한 내용을 제공하지 못하는 경우가 빈번하다.

그와 달리 내용 기반의 정보 검색 시스템은 정보의 내용을 수학적으로 분석하여 구조화된 기준에 따라 대표적인 특징을 추출하고, 그러한 특징을 토대로 데이터를 체계적인 구조로 색인화 하고 있다. 이러한 색인화는 멀티미디어 데이터의 신속하고 정확한 정보와 특징을 얻어낼 수 있고, 데이터 고유의 특징으로부터 사용자에게 적합한 내용을 제공할 수 있다. 특히 음악은 영상·음향·음성 등 멀티미디어 데이터들이 공통으로 포함하고 있는 정보 매체로서 내용을 분석하여 장르를 분류하고 검색하는데 핵심적인 역할을 한다.

내용기반 음악 장르 분류 및 검색 연구는 크게 미디(MIDI) 파일의 음악 표기 정보를 이용하는 방법과 디지털 신호처리(Digital Signal Processing) 기술을 이용하는 두 가지 방법이 있다. 미디를 이용한 방법은 논문 [1]에서 미디 파일 내의 멜로디를 이용하여 멜로디 내의 연속된 음들을 UDS 문자열로 표현하여 데이터베이스 내에 곡들과 비교 검색 한다. 그러나 문자열 정합을 위한 검색 속도에 대한 문제점이 있어 이를 개선하기 위한 다양한 연구가 발표되고 있다[2,3]. 디지털 신호처리를 이용한 방법으로는 논문 [4]에서 동물소리, 기계소리, 악기소리, 음성 등의 음향 효과들에 대한 신호크기, 음의 높낮이, 밝기, 대역폭, 하모니 등의 특징을 추출하여 유사한 오디오를 검색하였다. 논문 [5,6]에서는 음악 장르의 계층적인 자동 분류를 위해서 STFT(Short Time Fourier Transform) 기반의 오디오 특성과 웨이블릿 변환 기반의 리듬, 음의 높낮이 등의 오디오 특성을 추출하여 Classic, Jazz, Fork, R&B 등 음악 장르를 분류하였다.

기존 연구들은 음악 장르를 분류하기 위해 음악적 특징 벡터를 얻는 방법에 치중하였다. 음악을 분석하기 위한 범위는 임의의 20초 구간을 정하거나, 모의실험을 통하여 장르별 특성이 가장 잘 나타나는 부분을 선택해 음악의 40%~45% 구간으로부터 20초를 얻어 실험하였다[7]. 그러나 음악은 도입부분, 진행부분, 클라이맥스 부분 등으로 나뉘며 클라이맥스 부분이 그 음악의 특징을 가장 잘 나타내는 부분이다. 클라이맥스는 곡 전체에서 일정한 유사도 내에서 3번 이상 반복하는 특징과 그 음악 중 최고 절정 선율을 나타내는 특징을 가지고 있다[8,9]. 또한 클라이맥스는 그 음악에서 강조하려는 부분이며 그 음악에 사용되는 모든 악기가 나타난다. 즉 클라이맥스는 그 음악의 성격이나 특징을 결정짓는다고 말할 수 있다. 따라서 음악의 클라이맥스 부분을 자동으로 추출하여 장르 분류에 사용하면 보다 효과적인 결과를 얻을 수 있다.

본 논문에서는 음악의 클라이맥스 추출을 이용한 내용기반 장르 분류를 제안한다. 음악의 클라이맥스 추출 방법은 FFT(Fast Fourier Transform)와 파형 집중도를 이용하여 구하였다. 장르 분류는 Ballad, Dance, Hip-hop, Rock으로 각각 250곡을 선택하여 각각 100곡은 학습 데이터로, 150곡은 테스트 데이터로 사용하였으며, 특징 벡터는 STFT 기반의 특징들과 MFCC(Mell Frequency Cepstrum Coefficient)의 특징 벡터를 추출하였고 장르 분류 알고리즘은 k-NN(K-nearest neighbor)을 이용하였다. 논문의 구성은 다음과 같다. 2장에서 클라이맥스 추출을 위한 방법을 설명하고, 3장에서 내용기반 장르 분류 및 검색에 사용되는 특징 벡터 추출 및 장르 분류 알고리즘을 설명한다. 4장에서는 클라이맥스 추출과 장르 분류에 대한 실험과 분석을 하였으며, 5장에서 결론을 제시한다.

2. 클라이맥스 추출 방법

일반적으로 음악은 도입 부분과 진행 부분, 클라이맥스 부분으로 나눌 수 있다. 클라이맥스는 그 음악의 특징을 가장 잘 나타내는 부분이며, 다음과 같은 두 가지 특징을 가지고 있다. 첫 번째 특징은 음악에서 반복 선율이 일정 유사도내에서 3번 이상 반복하는 것이다. 두 번째 특징은 그 음악의 최고 정점을 나타낸다는 것이다. 우리는 두 번째 특징을 이용하여 음악의 클라이맥스 부분을 추출하도록 하였다. 클라이맥스 영역은 음악의 마디를 구하고 마디로부터 최대 파형 집중도를 찾음으로 구할 수 있다.

다음 그림 1은 클라이맥스 추출을 위한 구조도를 나타낸다. 클라이맥스 추출을 위해서는 다음과 같이 크게 2단계로 나뉜다. 첫 번째 단계에서 음악의 마디 길이와 마디 시작점을 찾아내며, 두 번째 단계에서 마디별 높은 파형 집중도 추적을 통해 찾아냄으로서 클라이맥스를 찾게 된다. 다음은 두 단계에 관한 방법을 차례로 설명하였다.

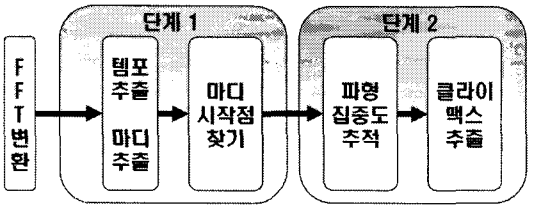


그림 1. 클라이맥스 추출을 위한 구조도

2.1 음악의 템포, 마디 길이, 마디 시작점 찾기

음악의 템포, 마디 길이, 마디 시작점을 찾는 방법은 논문[10]에서 제안한 방법을 이용하였으며, 그 방법은 다음과 같다.

CD 음질의 웨이브 파일은 대부분이 44.1KHz, 16Bit로 샘플링 된다. 그러한 파일의 4분 정도에 해당하는 크기는 40MByte정도이며 스테레오로 녹음되어있다. 대부분의 음악들은 멜로디를 담당하는 부분과 리듬을 담당하는 부분으로 구성 되어 있으며 그 중 리듬을 구성하는 부분은 규칙성을 가진다. 여기서는 마디 구간과 마디 시작점을 찾기 위해 드럼파트를 구성하고 있는 스네어 드럼이라는 악기에 초점을 맞추었다. 이 악기는 음악의 리듬에서 2번째와 4번째 박자에 규칙적으로 나타나며, 표현되는 주파수가 거의 일정하다는 특성을 가진다.

대부분의 음악들은 도입부분에서 특징을 나타내지 않기 때문에 40초가 지난 다음부터 곡들의 마디 시작점을 찾는다. 일반적으로 마디의 크기는 220,000개의 샘플링 이하라는 특징을 고려하여 40초가 지난 다음부터 262,144개(1024*256)의 샘플을 채취한다. 추출한 샘플로부터 256-point FFT를 1024번 수행하였으며, 이 때 변환된 데이터에서 스네어 악기의 소리는 37번째 주파수 대역에서부터 43번째 주파수 대역 사이에서 잘 나타나는 사실을 알 수 있다. FFT를 사용하여 변환된 값들로부터 스네어 드럼의 소리 후보를 찾기 위해 1024번 수행한 구간을 32개의 구간으로 나누고 각 구간에서 37번째에서 43번째까지의 각 주파수 대역별로 최대값을 찾는다. 32개의 구간으로 나누는 이유는 음악에서의 스네어드럼 간격은 한마디에 32회 이상 연주되지 않기 때문이다. 각 구간별 최대값을 정렬하여 각 주파수에서 상위 12개의 값들을 후보로 선정한 후, 선정된 값들 중 시간 위치 값이 32 간격 보다 작은 후보들을 비교하여 값이 낮은 쪽을 후보에서 삭제한다. 이는 32 구간으로 나누었을 때 그 간격 안에 스네어 드럼이 두 번 연속으로 나타날 수 없기 때문에 낮은 쪽 후보를 삭제하는 것이다. 그림 2는 SG워너비의 "내 사람"이라는 곡으로부터 샘플을 추출하여 37번째에서 43번째까지의 최대값 12개 후보를 선정한 것이다.

각 주파수별로 남겨진 후보들을 다른 주파수 후보들과 비교하여, 시간 위치 차이가 ±10이하인 위치 값들을 찾아서 투표한 후, 위치 값들의 평균값을 구한다. 따라서 그 평균값이 스네어 드럼이 나타나는 구간이라 예상 할 수 있다. 투표 결과가 6 이상인 후보들만을 추출한 후 시간 순서대로 나열하여, 후보들 사이의 시간 간격을 구한 뒤, 이들 중 가장 큰 값과 가장 작은 값을 뺀 나머지 값들의 평균을 구한다. 이 때 구해진 평균이 템포를 의미하며, 템포의 2배가 우리가 찾고자 하는 마디의 길이이다. 그림 3의 a는 앞서 구한 12개의 후보들 중 32 간격 안에 겹치는 후보를 제외하고 남은 후보의 위치 값을 나타낸다. 그림 3의 b는 후보들로부터 얻은 값을 시간 순서대로 나열하여, 시간 간격과 시간간격 평균을 구한 것이다.

스네어 드럼의 주기는 2번째와 4번째 박자에 규칙적으로 나타난다. 즉 첫 번째 스네어 드럼이 나타난 지점에서 마디 길이의 1/4 지점이 마디의 시작점이다. 따라서 마디의 시작점은 앞에서 찾은 후보들 중 가장 큰 값을 선정하고, 템포의 1/2을 뺀 값이 마디의 시작점이다. 그림 4는 FFT를 이용하여 템포와 마디 길이, 마디 시작점을 계산하는 방법을 도식화 한 것이다.

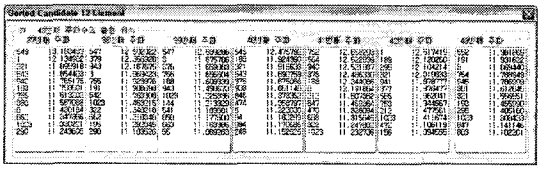


그림 2. 37번-43번 주파수 값이 최대가 되는 12개 후보 선정

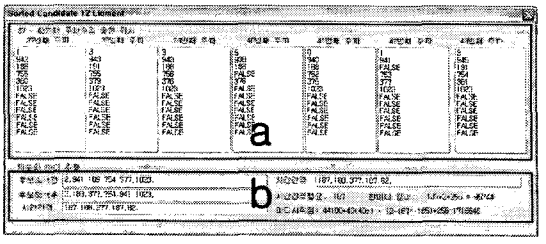


그림 3. 후보 삭제 및 시간 간격과 시간 간격 평균 계산

그림 4. FFT를 이용한 템포, 마디 길이, 마디 시작점 추출 순서도

2.2 파형 집중도 추적을 통한 클라이맥스 추출

클라이맥스는 2.1절에서 찾은 마디 길이와 마디 시작점을 이용하여 검색한다. 클라이맥스는 그 음악의 최고 정점을 나타낸다는 특징으로부터 마디별 최고 정점이 있는 곳을 찾는다. 최고 정점이라는 것은 그 음악 중 최대한의 소리를 내는 것이며 파형의 크기가 가장 높은 것을 나타낸다. 연구에서는 최대 파형 집중도를 마디로 나누어 측정하였으며, 집중도가 가장 높은 곳을 음악의 클라이맥스 시작 지점으로 구하였다. 그리고 시작 지점으로부터 템포에 8을 곱한 여덟 마디가 우리가 찾아낸 클라이맥스 부분이다[11].

마디별 최대 파형 집중도를 구하기 위해서는 먼저 음악 전체로부터 파형이 가장 큰 값을 찾는다. 그 후 2.1절에서 찾은 마디의 시작 지점에서부터 각각의 마디별로 최대 파형의 90% 이상 되는 파형들의 집중도를 구한다. 이때 최대 파형의 90% 이상을 선택한 이유는 다양한 실험을 한 결과 가장 높은 정확도를 나타냈기 때문이며, 최대 파형의 집중도가 가장 많은 마디가 찾고자 하는 클라이맥스의 시작 지점이다.

음악 계층 구조에서 악곡을 구성하는 가장 작은 단위는 동기(motif)이며, 동기 두 개가 모여서 작은악절(phrase)이 되고, 작은악절 두 개가 모여서 큰악절(period)이 된다. 음악 구성 형식에서 큰악절 두 개로 구성된 음악(두 도막 형식, 8개의 동기)의 경우, 앞 큰악절의 일부분을 뒤 큰악절에서 대조를 주어 변화시키고, 뒤 큰악절에서 앞 큰악절의 일부분을 모방해 반복시켜 통일성을 갖춘다[12]. 즉, 일반적인 음악의 클라이맥스는 여덟 마디로 구성된다. 따라서 2.1절에서 찾은 오디오 데이터의 템포를 이용하여 마디 길이를 구하고 2.2절에서 얻은 클라이맥스 시작점을 얻어 시작점에서 마디 길이에 8을 곱하면 클라이맥스 구간을 찾아낼 수 있다.

3. 특징 벡터 추출 및 장르 분류 알고리즘

3.1 특징 벡터 추출

각 특징 벡터의 분석 및 추출은 다른 여러 연구에서 이용되는 방법을 그대로 사용하였다[7]. 이는 음악을 장르별로 분류하거나, 음악 고유의 특징을 얻기 위해 현재 가장 많이 사용되는 방법이기 때문이다. 음악은 44.1KHz, 16bit, 스테레오로 샘플링 되었으며 실험에 사용된 음악 클립은 임의의 영역 선택 경우 전체에서 20초 분량을 임의로 선택하고, 40%~45% 위치의 음악의 경우 그 위치에서부터 20초 분량을 선택하였으며 클라이맥스 영역의 경우 추출한 여덟 마디 분량을 선택하였다. 선택한 음악 클립은 23ms 크기의 Hamming window를 중복되지 않게 이동하면서 각 23ms 프레임으로부터 특징을 추출하여 평균과 분산 값을 조합해서 총 32차의 특징 벡터를 구하였다[13,14]. 본 논문에서 사용된 특징 벡터들은 다음과 같다.

Spectral Centroid는 STFT의 크기 스펙트럼의 중심을 뜻하며, 스펙트럼의 형태를 측정하는 방법 중의 하나이다. 여기서 $M_t[n]$은 프레임 t와 주파수 Bin (Frequency Bin) n에서의 스펙트럼 크기에 해당한다. C_t는 프레임 t의 Centroid 값을 나타내며, N은 프레임 t내에 있는 주파수Bin의 최대값이다. Centroid의 값이 높을수록 더 높은 주파수에서 선명한 음질을 나타낸다.

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

Spectral Rolloff는 크기 스펙트럼의 형태와 낮은 주파수 영역에 신호의 에너지가 얼마나 집중되어 있는지를 보여준다. 아래의 수식은 스펙트럼 분포의 80%가 집중되어 있는 주파수 R_t을 구하기 위한 것이다. R_t는 크기 스펙트럼 전체를 더하여 80% 값을 얻은 후 다시 크기 값을 1에서부터 R_t까지 합으로 비교하여 구할 수 있으며, 이때의 R_t는 크기 분포의 80%가 집중되어 있는 주파수이다.

$$\sum_{n=1}^{R} M_t[n] = 0.8\times\sum_{n=1}^{N} M_t[n] \quad (2)$$

Spectral Flux는 연속된 스펙트럼의 분포에서 스펙트럼의 변화의 양을 측정하는 것이다. 아래의 수식은 연속된 스펙트럼 분포에서 정규화 된 크기들 간의 차이를 제곱해서 Spectral Flux를 구하기 위한 것이다. $N_t[n]$과 $N_t[n-1]$은 주파수 Bin n과 n-1에 해당하는 크기 값을 정규화 한 것이며, 이로 부터 스펙트럼의 변화량인 F_t를 구한다.

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

MFCC(Mell Frequency Cepstrum Coefficient)는 인간의 청각 특성을 모델링한 방법이다. 사람의 귀는 낮은 주파수에서는 작은 변화에도 민감하게 반응하지만, 높은 주파수로 갈수록 민감도가 작아지는 특성이 있다. 이러한 사람의 귀의 반응은 로그 스케일과 비슷하므로, MFCC는 그와 비슷한 멜(Mel) 스케일을 이용하여 켑스트럼 계수를 추출한다. MFCC를 얻는 방법은 오디오 신호의 크기 스펙트럼을 로그 스케일(log scale) 한 후 FFT bin을 그룹화 하여 멜 주파수(Mel-Frequency) 스케일로 변환하여 얻을 수 있다. 아래의 그림 5는 MFCC를 추출하는 일반적인 과정을 나타낸다.

장르 분류 실험에서 MFCC의 사용은 기존의 분류 실험에서 많이 사용되어 왔기 때문에 본 논문에서도 MFCC의 여러 계수중 성능이 가장 우수하게 나타나는 13차 계수를 사용하였다.

3.2 장르 분류 알고리즘

오디오 데이터의 장르 분류 알고리즘은 Ballad, Dance, Hip-hop, Rock등 4개의 오디오 장르 중 하나로 분류하는데 사용된다. 장르 분류를 위한 분류기는 k-NN, Gaussian 분류기, GMM(Gaussian Mixture Model) 분류기 등이 있으나 본 논문에서는 분류를 위한 데이터 값의 성능을 비교하는 것이므로 비교적 계산이 쉽고 빠른 k-NN 분류기로 사용하였다[15]. k-NN 분류기는 가장 기초적인 분류 방법 중 하나로, 모든 데이터가 n-차원공간 Rn상의 점들로 대응된다고 가정한다. 한 데이터와 다른 데이터 사이의 최근접은 표준 유클리드 거리(standard Euclidean distance)를 통해서 정의된다. r번째 특징 $a_r(x)$를 가진 임의의 x가 다음의 특징 벡터를 가진다고 가정하면

$$< a_1(x), a_2(x), ..., a_n(x) >$$

와 같이 표현되고, 두 데이터 x_i 와 x_j 사이의 거리 $d(x_i, x_j)$는 다음과 같다.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \quad (4)$$

최근접 학습에서 목표 함수는 이산값 (discrete-value)이거나 실수값(real- value)이 된다. 그림 6은 k-NN 알고리즘이 데이터가 2차원 공간상의 한 점이고 목표함수가 이항 값을 가지는 경우에 어떻게 작동하는가를 보여준다. 그림 3에서 '+'는 양성인 데이터를, '-'는 음성인 데이터를 각각 의미한다. k-NN 알고리즘에서 k는 가장 근접해 있는 원소의 개수를 나타내며, 그 개수들 중 가장 많은 원소로 분류되게 하는 것이다. 그림 3에서 k=1일 때 가장 근접해 있는 원소가 양성을 나타냄으로 x_q는 양성으로 분류된다. 반면에 k=5일 때는 가장 근접해 있는 원소가 양성 2, 음성 3개로써 x_q는 음성으로 분류된다.

따라서 3.1절에서 언급한 음악 각각의 고유한 특징 벡터들로부터 k-NN 알고리즘을 이용하여 음악의 장르를 분류하는 실험을 할 수 있다. 아래의 표 1은 실험에 사용된 음악 각각의 고유한 특징 벡터를 추출한 것으로 장르에 따라 하나씩 나열하였다.

그리고 표 1의 음악 중 "나에게로 떠나는 여행 : 버즈 (Rock)"의 경우 주변의 특징 벡터 값들이 아래 표 2와 같은 값을 가지므로 Rock으로 분류되어진다.

그림 5. MFCC 추출 과정

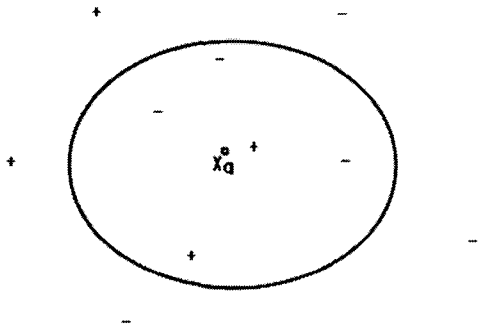


그림 6. k-NN 알고리즘의 동작 예

표 1. k-NN 알고리즘에 사용하기 위해 추출된 음악 각각의 특징 벡터 예

노래 제목 : 가수 (장르) MFCC 1차(평균,분산), 2차(평균,분산), ... , 13차(평균,분산), Spectral Centroid(평균,분산), Spectral Rolloff(평균,분산), Spectral Flux(평균,분산)
나의 사랑 나의 신부 : UN (Dance) -3.231444, 1.425880, -2.911192, 0.532459, -3.076864, 0.530777, -0.346156, 0.422847 -4.209618, 0.332688, -1.006735, 0.272735, -2.984929, 0.207991, -1.919845, 0.149125 -1.899662, 0.190117, -2.136086, 0.168896, -1.339029, 0.188681, -1.486132, 0.145000 -1.109830, 0.098528, 127.599236, 1.608336, 140.451218, 227.744385, 0.948822, 7.577481
혼자만의 겨울 : 강수지 (Ballad) -4.127745, 0.644089, -2.462150, 0.440625, -2.975618, 0.418005, -0.612523, 0.314531 -4.504005, 0.329062, -0.250909, 0.310219, -2.809120, 0.204638, -2.245306, 0.181694 -1.709603, 0.262543, -2.215042, 0.254990, -1.205766, 0.159998, -1.634923, 0.110851 -1.242730, 0.079867, 127.927208, 0.051366, 135.319305, 91.106415, 1.218667, 13.743856
나에게로 떠나는 여행 : 버즈 (Rock) -3.452116, 1.286351, -2.827275, 0.298868, -3.809378, 0.498503, -0.976887, 0.374962 -3.971190, 0.298838, -0.742623, 0.219051, -2.938645, 0.168876, -1.831205, 0.167476 -1.864434, 0.199179, -2.102812, 0.290635, -1.352014, 0.200475, -1.722061, 0.134949 -1.258282, 0.078147, 127.803238, 0.523599, 136.237076, 57.754826, 1.721653, 27.260227
Word Up : Side B (Hiphop) -4.513717, 1.811275, -2.100210, 0.884710, -2.569916, 0.586154, -1.248310, 0.698325 -3.454865, 0.445167, -1.426042, 0.313039, -2.583387, 0.193159, -1.986977, 0.187092 -1.912023, 0.159311, -2.050760, 0.166754, -1.440745, 0.142641, -1.635351, 0.108980 -1.173671, 0.076572, 127.603233, 1.202247, 142.793549, 373.479065, 0.374924, 3.737888

4. 실험 및 분석

클라이맥스 추출은 다음과 같이 이루어졌다. Ballad, Dance, Hip-hop, Rock 각각의 장르로 분류되는 오디오 데이터 250곡씩 총 1000곡을 사용하였으며, 분류 구분은 국내 음악 서비스 업체 중 하나인 벅스 뮤직 (Bugs Music)의 국내 가요 분류를 따랐다. 또한 클라이맥스 추출의 결과에 대한 정확성 여부는 논문[8,9]에서 정의하고 있는 반복 선율이 3번 이상으로 나타나는 부분이 포함 되어 있고, 최고 정점이 나타나는 경우 검색이 된 것으로 결과를 측정하였다. 즉, 수동으로 찾은 실제 클라이맥스와 제안한 알고리즘에 의해 추출한 클라이맥스를 비교하였다.

클라이맥스 추출은 표 3과 그림 7의 결과를 얻을 수 있었다. 표 3에서 보는 바와 같이 전체 곡에 대하여 템포를 찾은 음악은 737곡이었다. 이때 실제 클라이맥스와 일치하는 클라이맥스 추출은 586곡으로 템포를 찾은 음악의 79.5%에 해당하는 결과를 얻었다.

그림 7은 각각 클라이맥스 추출 시 템포를 찾아낸 곡, 클라이맥스를 찾아낸 곡, 찾아 낸 클라이맥스가 실제 클라이맥스와 일치하는 곡을 장르별로 비교한 그림이다. Hip-hop이 가장 큰 정확도를 나타내었는데 이는 Hip-hop의 구성이 일정하고 정확한 리듬을 가진 특징 때문이다. Rock 장르는 다른 장르에 비해 정답률이 낮았는데, 간주 중 기타 연주부분이 강렬하게 들어가기 때문에 그 부분을 클라이맥스로 찾은 오류를 나타내었다. 전체적으로 정답이 되지 않은 약 20%의 곡은 높은 파형 집중도 부분을 찾아내었지만, 여러 복합적인 소리가 섞인 부분이나, 기타연주, 폭발음과 같은 효과음으로 인한 결과였다.

본 논문에서는 앞에서 추출한 클라이맥스를 이용하여 장르 분류가 더 잘 되는가를 검증하기 위하여

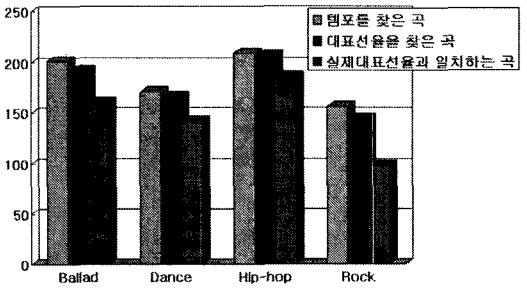


그림 7. 클라이맥스 추출 결과

Ballad, Dance, Hip-hop, Rock 4장르에 대한 내용기반 장르 분류 실험을 하였다.

실험을 위한 오디오 데이터는 Ballad, Dance, Hip-hop, Rock 각각의 장르별로 100곡씩 총 400곡을 학습 데이터로 사용하였고, 테스트 데이터는 장르별로 150곡씩 총 600곡을 사용하였다. 특징을 추출하는 영역은 임의 구간 추출을 통한 20초, 일반적으로 특징 추출에 사용하는 방법인 음악의 40%~45% 이후 부분부터의 20초, 그리고 제안하고자 하는 클라이맥스 구간으로 하였다. 이때 클라이맥스를 찾지 못한 데이터는 40%~45% 지난 부분에서부터 20초 분량을 선택하게 하였다. 특징 추출은 음악의 Spectral Centroid, Rolloff, Flux등 STFT 기반의 특징 계수들과 MFCC의 13차 특징 계수를 추출하였다.

먼저 Ballad와 Dance 두 개의 장르만으로 비교한 실험 결과는 표 4와 같다.

표 4의 결과에서 40% 이후 20초 구간과 클라이맥스 구간의 정확도에 차이가 많지 않음을 알 수 있다. 다음으로 Ballad, Dance, Hip-hop 3장르로 비교한 실험 결과는 표 5와 같다.

표 5에서는 장르가 3가지로 늘어나면서 각각의 구간에 대한 정확도가 떨어짐을 알 수 있으며, 임의 20초와 40% 이후 20초 구간에 비해 클라이맥스 구간의 정확도가 더 나음을 알 수 있다. 마지막으로 Ballad, Dance, Hip-hop, Rock 4장르로 분류한 실험 결과는 표 6과 같다.

표 6에서는 앞의 실험결과와 연결해서 알 수 있듯이 장르가 늘어남에 따라 정확도는 계속 떨어짐을 알 수 있었다. 그러나 모든 실험에서 클라이맥스 구간이 임의 20초, 40% 이후 20초 구간에 비해 높은 정확도를 나타냈다. 이로부터 클라이맥스 구간을 이용한 장르 분류가 보다 효과적임을 알 수 있었다. 실험에서 임의 20초 구간의 결과가 40% 이후 20초 구간의 결과보다 높거나, 비슷한 정확도를 나타내는 것은 임의 20의 구간이 클라이맥스 부분을 얻어온 경우가 많았을 것으로 추측된다.

표 2. k=7에 의한 음악 분류 시 주변 특징 벡터 값과 장르

	32차 특징 벡터	장 르
1	-2.264803, 0.821657, -1.690358, 0.341683, -3.564268, 0.364506, -0.996289, 0.335643, -3.481775, 0.308496, -1.394370, 0.173946, -3.211224, 0.192043, -1.988770, 0.161563, -2.412110, 0.139417, -1.498401, 0.133352, -1.699932, 0.144511, -1.363075, 0.098361, -1.478799, 0.064482, 127.780457, 0.196326, 131.620483, 9.118141, 1.054802, 12.141028	Ballad
2	-2.241071, 0.985050, -1.398446, 0.632243, -3.886135, 0.422360, -1.352893, 0.412740, -3.709383, 0.270711, -1.222635, 0.185251, -2.646103, 0.141024, -1.546199, 0.155983, -2.365346, 0.150043, -1.544612, 0.171527, -1.768026, 0.119493, -1.126648, 0.119994, -1.331100, 0.080658, 127.886459, 0.087888, 131.158218, 6.281412, 0.593484, 14.294839	Rock
3	-1.795399, 0.927754, -1.390768, 1.215676, -3.410659, 0.403854, -0.531288, 0.401364, -3.542469, 0.363648, -0.962649, 0.218209, -2.787440, 0.165378, -1.622843, 0.176729, -2.241978, 0.218101, -1.323790, 0.179115, -1.617438, 0.156698, -1.074244, 0.115433, -1.285114, 0.090986, 127.875641, 0.088537, 131.564667, 11.387774, 1.414308, 15.935861	Ballad
4	-2.430016, 0.651146, -0.974542, 1.003383, -3.144571, 0.452063, -0.745389, 0.429499, -3.413504, 0.378184, -0.769675, 0.242435, -2.855322, 0.151637, -1.406407, 0.191060, -2.007724, 0.191700, -1.532968, 0.177897, -1.448982, 0.196940, -1.075758, 0.120455, -1.339242, 0.106166, 127.879013, 0.100859, 132.193939, 8.104801, 0.908280, 10.219082	Rock
5	-2.387352, 0.821812, -1.822020, 0.603403, -3.507619, 0.656792, -0.432234, 0.604624, -3.487535, 0.427327, -0.757873, 0.295707, -2.655880, 0.190681, -1.957351, 0.141545, -2.425233, 0.199684, -1.768188, 0.163069, -1.916278, 0.121649, -1.293506, 0.088520, -1.291375, 0.081535, 127.584229, 2.909235, 131.290268, 11.072625, 1.040777, 15.869842	Rock
6	-3.162851, 1.461876, -2.342339, 0.291565, -4.380234, 0.681271, -1.136846, 0.262745, -3.179206, 0.267397, -0.902120, 0.179813, -3.016815, 0.158071, -1.594977, 0.165696, -2.104217, 0.128954, -1.618125, 0.176386, -1.698178, 0.185332, -1.258463, 0.125576, -1.367657, 0.083958, 127.947769, 0.014016, 133.444427, 10.073431, 0.937373, 11.418728	Rock
7	-2.409095, 0.678877, -2.373830, 0.569508, -3.578449, 0.490934, -0.554714, 0.340882, -4.305418, 0.399084, -0.809846, 0.284530, -2.836823, 0.157790, -1.972246, 0.151485, -2.294287, 0.174894, -1.904917, 0.167517, -1.771796, 0.106219, -1.325189, 0.084372, -1.336998, 0.070921, 127.760689, 0.870835, 132.006104, 11.818720, 1.268673, 16.662960	Rock

표 3. 템포와 클라이맥스 추출 결과

	전체곡	템포를 찾은 곡	클라이맥스 검색	정답	정확도 (전체곡에서의 정확도)
Total	1000	737	709	586	79.5% (58.6%)

표 4. Ballad와 Dance 장르 분류 결과

음악 장르	임의 20초		40% 이후 20초		클라이맥스 구간	
	정인식	오인식	정인식	오인식	정인식	오인식
Ballad	106	44	113	37	109	41
Dance	103	47	113	37	120	30
정확도	70 %		75.3 %		76.3 %	

기존의 논문에서는 각 음악의 40% 지점에서부터 20초를 추출하여 STFT 기반의 특징들과 MFCC, LPC 등의 특징 벡터들을 추출 후 SFS를 적용하고, k-NN 분류기 등을 이용하여 약 80%의 장르분류 성공률을 보였다[7]. 그러나 기존 논문에서 사용한 실험 데이터는 Hip-hop, Rock과 같은 음악 간의 분류만을 한 실험이 아니라 Speech 같은 음악과 전혀 다른 요소를 분류에 사용하였고, Classic과 같은 음성이 없는 장르를 추가하므로 높은 성공률을 보일 수 있었다. 그에 비해 본 논문에서는 실제 음악 서비스에서 사용되고 있는 음악 장르를 분류하기 위해 Ballad, Dance, Hip-hop, Rock등 4개의 장르를 사용하였다. 이를 기존 방법인 40초 이후의 20초를 선택하여 분류해 본 결과 47%의 성공률을 보이는 반면, 제안한 방법인 클라이맥스 부분을 이용한 방법에서는 56%의 성공률을 얻었다. 또한 여기서는 기존 논문에서 제안한 SFS 기술을 적용하지 않았지만, 추후 실험에서 적용한다면 보다 높은 성공률을 얻을 수 있을 것이다.

5. 결론과 향후 연구 방향

본 논문에서는 음악의 클라이맥스 추출을 이용하여 내용 기반 장르 분류를 하였다. 제안한 방법에 의해 음악의 클라이맥스를 추출할 수 있었으며, 실험을 통하여 기존의 연구에 사용하고 있는 구간보다, 클라이맥스 구간을 이용했을 때 장르 분류의 성공률이 증가함을 볼 수 있었다. 따라서 클라이맥스 구간을 이용하여 장르를 분류하면 보다 효과적임을 알 수 있다. 이러한 장르 분류 연구는 방대한 양의 음악을 특징별로 분류할 때 사용할 수 있을 것이다.

추후 연구로 장르 분류 실험은 음악 데이터의 크기가 크기 때문에 특징 파악을 위한 시간이 오래 걸리는 단점이 있다. 음악의 특징 벡터 종류는 다양한 종류가 있다고 한다. 이러한 다양한 특징 벡터 중 장르 분류에 적합한 특징 벡터가 무엇이 있는지를 알아

표 5. Ballad, Dance, Hip-hop 장르 분류 결과

음악 장르	임의 20초		40% 이후 20초		클라이맥스 구간	
	정인식	오인식	정인식	오인식	정인식	오인식
Ballad	96	54	100	50	104	46
Dance	80	70	76	74	98	52
Hip-hop	84	66	83	67	83	67
정확도	57.8 %		57.6 %		63.3 %	

표 6. Ballad, Dance, Hip-hop, Rock 장르 분류 결과

음악 장르	임의 20초		40% 이후 20초		클라이맥스 구간	
	정인식	오인식	정인식	오인식	정인식	오인식
Ballad	76	74	75	75	86	64
Dance	65	85	59	91	88	62
Hip-hop	80	70	80	70	79	71
Rock	68	82	72	78	81	69
정확도	48.2 %		47.7 %		55.7 %	

내고 짧은 시간에 특징 벡터를 추출할 수 있는 방법에 대한 연구가 필요하다.

또한 클라이맥스 추출 시 템포의 추출이 안 되는 음악을 보안하기 위한 새로운 템포 추출 연구와 클라이맥스 추출의 정확도를 높이기 위하여 반복 선율이 일정 유사도내에서 3번 이상 반복하는 클라이맥스 특징을 찾을 수 있는 연구가 필요하다. 특히 Classic과 같이 스네어 드럼이 포함되지 않은 장르에 대해서 템포 추출을 위한 방법을 고려하여야 한다.

본 논문에서는 클라이맥스 부분을 이용한 장르 분류가 보다 효과적이라는 것을 나타내기 위해 장르 분류 알고리즘인 k-NN을 이용하였다. 클라이맥스로부터 다양한 특징 벡터를 추출할 때 추출한 특징 벡터에 적합한 장르 분류 알고리즘을 적용하면 정확도는 보다 높아질 것이다. 따라서 장르 분류의 정확도를 높이기 위해 특징 벡터 계산을 이용한 개선된 장르 분류 알고리즘 연구가 필요하다.

참 고 문 헌

[1] A. Ghias, J. Logan, D. Chamberlin, and B. Smith, "Query by Humming: Musical Information Retrieval in an Audio Database," *ACM Multimedia*, pp. 231-236, 1995.

[2] M. Melucci and N. Orio, "Musical Information Retrieval using Melodic Surface," *ACM Multimedia*, pp. 152-160, 1999.

[3] R. J. McNab, L. Smith, I. H. Witten, and C. L. Henderson, "Tune Retrieval in the Multimedia Library," *Multimedia Tools and Applications*, Vol. 10, pp. 113-132, 2000.

[4] E. Wold, T. Blum, D Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia*, Vol. 3, No. 3, pp. 27-36, 1996.

[5] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for browsing and annotation," *ASPAA*, pp. 103-106, 1999.

[6] G. Tzanetakis and P. Cook, "Musical Genre Classification of audio Signal," *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 5, pp. 293-302, 2002.

[7] 윤원중, 이강규, 박규식, "내용기반 오디오 장르 분류를 위한 신호 처리 연구," 전자공학회논문지, 제41권, 제6호, pp. 271-278, 2004.

[8] Alan Smaill, Geraint Wiggins, and Mitch Harris, "Hierarchical music representation for composition and analysis," *Computers and the Humanities*, Vol. 27, No. 11, pp. 7-17, 1993.

[9] Harris M, Smaill AD, and Wiggins G, "Representing Music Symbolically," *In the Proceedings of IX Colloquium on Musical Informatics*, Italy, 1991.

[10] Jae-Won Lee, Chan-Yun Cho, and Sang-Kyoon Kim, "Music Genre Classification using Time Delay Neural Network," *Korea Multimedia society*, No. 4-5, pp. 414-422, 2001.

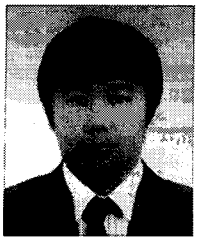
[11] Belkin, Alan, "A Practical Guide to Musical Composition," http://www.musique.umontreal.ca/personnel/Belkin/bk/, 1995-1999.

[12] 구경이, 임상혁, 이재헌, 김유성, "내용 기반 음악 정보검색을 위한 음악 구성 형식을 고려한 대표 선율의 추출 및 색인," 정보처리학회논문지, 제11-D권, 제3호, pp. 495-508, 2004.

[13] J. M. Gray, *An Exploration of Musical Timbre*, PhD thesis, Dept. of Psychology, Stanford University, 1975.

[14] M. Slaney, *A critique of pure audition*, Computational Auditory Scene Analysis, Canada, 1995.

[15] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, University of Illinois at Urbana, 2001.



정 명 범

2004년 숭실대학교 미디어학부 졸업 (공학사)
2006년 숭실대학교 미디어학과 졸업 (공학석사)
2006년~현재 숭실대학교 미디어학과 박사과정
관심 분야 : 디지털 신호처리, 감성인식, 콘텐츠 공학

고 일 주

1992년 숭실대학교 전산학과 (공학사)
1994년 숭실대학교 전산학과 (공학석사)
1997년 숭실대학교 전산학과 (공학박사)
2003년~현재 숭실대학교 미디어학부 조교수
관심 분야 : 콘텐츠, 정보검색, 감성공학