

# 음악 정보검색 시스템을 위한 효율적인 특징 벡터 추출에 관한 연구

## A Study on the Efficient Feature Vector Extraction for Music Information Retrieval System

윤 원 중\*, 이 강 규\*, 박 규 식\*  
(Won-Jung Yoon\*, Kang-Kyu Lee\*, Kyu-Sik Park\*)

\*단국대학교 정보·컴퓨터학부 컴퓨터과학 전공  
(접수일자: 2004년 7월 28일; 채택일자: 2004년 8월 31일)

본 논문에서는 Classic, Hiphop, Jazz, Rock 4개의 장르로 곡을 구분하여 각 장르별 60곡씩 총 240곡의 음악 DB를 대상으로 예제 질의 (QBE) 방식의 음악 정보 검색 시스템을 제안하였다. 제안된 시스템은 입력 질의로부터 spectral centroid, rolloff, flux 등 STFT 기반의 특징들과 MFCC, LPC, Beat 정보 등의 총 60차의 특징 벡터들을 추출한 후 Euclidean 유사도를 측정해서 DB내의 해당 음악을 검색한다. 실제 검색에 사용되는 특징 벡터는 SFS (Sequential Forward Selection) 기법을 사용하여 10차 특징 벡터로 최적화되며 검색 실험결과 평균 84% Hit Rate 와 0.63 MRR 의 성공률을 보이고 있어 기존의 연구 결과보다 약 10% 이상의 성능 향상을 보였다. 한편 본 논문에서는 실제 시스템 사용 환경을 고려하여 임의 질의 구간과 임의 질의 길이에 대한 시스템 성능 평가를 수행하였으며 실험 결과 이러한 임의성에 기인한 검색 성능의 불안정성을 지적하였다.

**핵심용어:** 음악 정보 검색, SFS, 예제 질의 방식, Random query pattern, query length

**투고분야:** 음향 신호처리 분야 (1.1)

In this paper, we propose a content-based music information retrieval (MIR) system base on the query-by-example (QBE) method. The proposed system is implemented to retrieve queried music from a dataset where 60 music samples were collected for each of the four genres in Classical, Hiphop, Jazz, and Rock, resulting in 240 music files in database. From each query music signal, the system extracts 60 dimensional feature vectors including spectral centroid, rolloff, flux base on STFT and also the LPC, MFCC and Beat information, and retrieves queried music from a trained database set using Euclidean distance measure. In order to choose optimum features from the 60 dimension feature vectors, SFS method is applied to draw 10 dimension optimum features and these are used for the proposed system. From the experimental result, we can verify the superior performance of the proposed system that provides success rate of 84% in Hit Rate and 0.63 in MRR which means near 10% improvements over the previous methods. Additional experiments regarding system performance to random query patterns (or portions) and query lengths have been investigated and a serious instability problem of system performance is pointed out.

**Keywords:** Music information retrieval, SFS, QBE, Random query pattern, Query length

**ASK subject classification:** Acoustic signal processing (1.1)

### I. 서론

최근 인터넷의 급격한 발전으로 사용자들은 웹 상에서

이미지, 오디오, 비디오 등이 결합된 다양한 형태의 멀티미디어 정보를 손쉽게 접할 수 있게 되었고, 그 중 음악은 하루에도 수천 곡씩 쏟아지고 있으며 이러한 증가 추세는 앞으로도 계속 될 것으로 보인다.

현재 음악 정보들은 MP3를 기반으로 ID3 태그를 수작업으로 입력하고 DB에서 텍스트를 기반으로 음악을 검색하고 있다. 그러나, 이러한 방식으로는 검색하고자 하

책임저자: 박 규 식 (kspark@dankook.ac.kr)  
서울시 용산구 한남동 산 8번지 140-714  
단국대학교 정보·컴퓨터학부 컴퓨터과학 전공  
(전화: 02-709-2728; 팩스: 02-796-2970)

는 음악에 대한 제목, 가수, 작곡자, 장르 등의 텍스트 정보가 필요하며, DB화 하는데 많은 인적 자원이 소모된다. 최근 이러한 문제점을 보완하고 대용량 음악 DB에 효율적으로 적용할 수 있는 내용기반 음악 정보 검색 기법이 주목을 받고 있다.

내용기반 음악 정보 검색 시스템은 기존의 수작업으로 이루어지던 분류 시스템과 달리 음악 정보의 내용(contents)을 수학적으로 분석하여 구조화된 기준에 따른 대표적인 특성을 추출하고 컴퓨터를 통해 데이터를 체계적인 구조로 색인화 한다. 이러한 색인화 및 분류, 검색 작업은 데이터의 신속하고 정확한 정보 취득을 제공할 수 있기 때문에 멀티미디어 관련 응용 연구에 필수적인 기반 기술로 사용될 수 있다. 특히 음악을 포함한 오디오는 영상·음향·음성 등 멀티미디어 데이터들이 공통으로 포함하고 있는 정보 매체로서 입력 정보의 내용을 분석하여 장르를 분류하고 검색하는데 핵심적인 역할을 한다.

기존의 내용기반 오디오 및 음악 정보 검색에 관련된 연구는 1997년 미국 Audible Magic 사의 MuscleFish[1]가 그 효시로서, 최근에는 ACM ISMIR 학술회를 중심으로 다양한 연구 논문이 발표되고 있으며 크게 예제 질의(QBE: Query By Example) 방식과 허밍 질의(QBH: Query By Humming) 방식 2가지로 구별된다. QBE 방식은 질의 음악 클립을 디지털 신호처리 기법을 사용하여 음악의 음색이나 지각 특징을 추출하여 DB내의 음악과 비교 검색하는 방법으로 비교적 구현 방법이 복잡한 반면 모든 오디오 파일 포맷에 적용 가능하다는 장점이 있다. 반면 QBH 방식은 사용자의 휘파람이나 허밍을 질의로 사용하는 것으로 입력 받은 허밍의 음높이(pitch)의 변화를 감지하여 검색하는 방법이다. QBH는 주로 MIDI 파일 내의 멜로디 음표나 pitch 변화를 대상으로 하여 비교적 검색 시간이 짧고 구현이 쉽다는 장점이 있으나 적용범위가 제한되는 단점이 있다.

QBE 방식으로 E. Wold[1]는 MuscleFish를 통해 15초 미만의 동물소리, 기계소리, 악기소리, 음성 등의 음향 효과들에 대하여 신호의 크기(loudness), pitch, 밝음(brightness), 대역폭, 하모니 등의 특징들을 추출하여 DB내에서 유사한 오디오를 검색한다. 약 80%의 검색 성공율을 보이고 있으나 검색 대상이 비교적 각 오디오 특성이 잘 구별되는 특수 음향을 대상으로 하고 있다. T. Chang[2]는 HMM을 이용한 계층적 구조의 오디오 분류 및 검색 시스템을 제안하였는데 역시 음성, 박수 소리,

폭발음, 새소리 등의 배경음 등을 대상으로 하여 약 80%의 성공률을 보였다. G. Tzanetakis[3]는 STFT 기반의 오디오의 표면적 특성과 Wavelet Transform 기반의 rhythm, pitch 등의 음악 특성 정보들을 추출 조합하여 약 74%의 성공률을 보였다. Guo 와 Li[4]는 SVM(Support Vector Machine) 기반의 이진 트리 기법을 이용하여 검색 오류율을 줄일 수 있는 방법을 제안하였다.

QBH 방식으로 논문 [5]는 MIDI 파일 내 멜로디를 이용하여 멜로디 내의 연속된 음들을 U(높음), D(낮음), S(같음)로 표현하여 DB내 곡들과 UDS 문자열을 비교하여 검색한다. 그러나 이 경우 문자열 매칭을 위한 검색 속도에 대한 문제점이 있어 이를 개선하기 위한 다양한 연구가 발표되고 있다[6-8].

본 논문에서는 Classic, Hiphop, Jazz, Rock 4개의 장르로 곡을 구분하여 각 장르별로 60곡씩 총 240곡의 음악 DB를 대상으로 QBE 방식의 음악 정보 검색 시스템을 제안한다. 입력 음악 질의로부터 디지털 신호처리 기법을 이용하여 centroid, rolloff, flux 등의 STFT 기반의 특징들과 MFCC, LPC, Beat 정보 등의 특징 벡터들을 추출한 후 Euclidean 유사도 측정법을 이용해서 DB와 비교하여 해당 음악을 검색한다. 음악 정보 검색에 사용되는 음악 특징 벡터는 SFS(Sequential Forward Selection) 기법을 사용하여 최적화되며 전체적으로 약 86% 이상의 검색 성공률을 보이고 있어 기존 방법보다 약 5~10% 가까운 성능 향상을 보이고 있다. 특히 본 논문에서는 실제 사용자 환경을 감안하여 임의 구간 질의(Query) 입력과 질의 길이에 따른 검색 시스템 성능을 모의 실험하여 문제점을 분석하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 본 논문에서 사용하게 될 특징 벡터 추출 및 최적 특징 벡터 선정 방법에 대하여 설명한다. 3장에서는 검색 실험을 위한 환경과 성능 평가 방법을 설명하고 4장에서는 컴퓨터 모의 실험을 통한 비교 분석을 수행하였다. 마지막으로 결론으로 글을 맺는다.

## II. 특징 벡터 추출 및 최적 특징 벡터 선정

음악 정보 검색 시스템의 동작과정은 크게 음악 특징 벡터 추출 과 유사도 측정 및 검색 등의 2가지 과정으로 구분될 수 있다. 특징 벡터 추출은 트레인용 DB 음악과

표 1. 프레임 별 특징 벡터  
Table 1. Feature vectors in frame.

특징 계수	표기	설명
spectral centroid	Cent	STFT의 magnitude 스펙트럼의 중심을 뜻하며, 스펙트럼의 형태를 측정하는 방법중의 하나이다.
spectral rolloff	Roll	스펙트럼의 형태와 낮은 주파수 영역에 신호의 에너지가 얼마나 집중되어 있는지를 보여 준다.
spectral flux	Flux	연속된 스펙트럼 분포에서 정규화된 magnitude들 간의 차이를 제곱해서 구할 수 있으며, 스펙트럼 변화의 양을 계산할 수 있다.
Zero Crossing Rate	ZCR	오디오 신호 파형의 위상이 중심축을 통과하는 횟수를 나타내며, 신호의 주파수 내용을 측정하는 가장 간단한 특징으로 음성인식에서 유·무성음의 판별에 사용된다.
Mel Frequency Cepstrum Coefficient	MFCC	인간의 청각 특성을 모델링 하는 방법으로 오디오 신호의 magnitude 스펙트럼을 log scale 한 후 FFT bin을 그룹화하여 인간의 청각 특성에 맞는 Mel-Frequency 스케일로 변환한 것이다.
Linear Prediction Coefficient	LPC	인간의 발성 모델에 입각해서 음성 신호를 부호화하는 방법으로 오디오 파형의 샘플 값에서 필터 계수를 구하여 상대에서 입, 코까지의 성도 특성을 8~12차의 전극형 (All-pole) 필터에 근사 시키는 방법이다.
Beat Histogram	Beat	오디오 신호의 리듬 정보를 수치적으로 산출하는 계수로서, Wavelet변환 후 대역 별 상관도를 구하여 Beat 히스토그램을 만들고 Beat정보를 추출한다(3).

입력 질의 (Query)로 부터 해당 음악을 가장 잘 대표할 수 음악의 표면적, 지각적 특징 벡터들을 추출하여 메타 데이터 형식으로 변환한다. 유사도 측정은 질의 음악과 트레인 DB 음악 데이터 간의 유사성을 수치화 하는 기술로서 검색 성능 전체에 영향을 준다.

2.1. 특징 벡터 추출

특징 벡터를 추출하는 이유는 해당 음악을 효과적으로 대표할 수 있는 벡터 열을 구성하여 비교하는 것이 검색 시간을 절약할 수 있으며, 해당 음악의 물리적, 지각적, 감성적 표현 특징 등을 분석하여 사용자에게 보다 유용한 정보를 전달할 수 있기 때문이다.

각 음악 신호로부터 특징 벡터를 추출하기 전 각 음악 신호들은 평균값이 0 분산이 1 이 되도록 정규화 과정을 거친다. 이러한 정규화 이유는 각 특징 벡터 값 들의 조그만 편차에 기인하는 검색 성능상의 오류를 없애기 위함이다. 각 특징 벡터의 분석 및 추출은 다음과 같은 조건에서 수행하였다. 음악 신호는 22050Hz, 16bits, mono로 샘플링 되었으며 실험에 사용된 오디오 클립은 20초 분량에 해당한다. 20초 분량의 오디오 신호는 23ms의 hamming window 를 한 프레임으로 하여 중복 되지 않게 이동하면서 각 프레임으로부터 특징을 추출하여 평균과 분산 값을 조합해서 총 60차의 특징 벡터를 구하게 된다. 각 프레임에서 추출하게 되는 특징 계수는 spectral centroid, rolloff, flux 그리고 ZCR 각 1개, LPC 10차 계수, MFCC 13차 계수로서 총 27개 계수들의 평균과 분산 54개와 6개의 Beat 계수들을 조합하게 되면 총 60차 특징 벡터가 존재하게 된다. 각 특징 벡터들

을 추출하는 방법은 신호 및 음향 처리 분야에서 잘 알려져 있는 기법들이므로 여기서는 각 특징들에 대한 간단한 설명만을 표 1 에 나타내었다.

2.2. SFS(Sequential Forward Selection)를 이용한 최적 특징 벡터 선정

전 절에서 구한 총 60차의 특징 벡터는 상당부분 특징 데이터간에 높은 상관성을 가지고 있어 이러한 상관성을 제거 함으로서 특징 벡터 차원을 줄일 수 있다. 특징 벡터 차원을 줄임으로서 검색 시간과 계산량을 줄일 수 있으며 동시에 합리적인 규모의 검색 DB 구축이 가능하다. 예를 들어 60차 특징 벡터는 모의 실험을 위한 소규모의 DB에서는 큰 문제가 되지 않지만, 10만~100만 곡 정도 되는 방대한 규모의 DB에서는 검색 속도에 큰 영향을 줄 수 있다. 따라서 본 논문에서는 [9]에서 음악 장르분류 목적으로 소개되고 있는 SFS 기법을 사용하여 특징 벡터의 차원을 줄여 1/6 에 해당하는 10차의 최적 특징 벡터만을 선정해서 검색에 사용하게 된다.

SFS 방법은 먼저 각 특징 계수들을 개별적으로 사용하여 검색을 해본 후, 가장 좋은 성공률을 나타내는 특징부터 순차적으로 나열하여 새로운 60차의 특징 벡터 열을 만든다. 다음으로는 첫 번째 특징 벡터의 성분부터 순차적으로 특징 계수의 수를 늘려가면서 혹은 전체 60개의 특징 벡터 열에서 하나씩 특징 계수를 감소시키면서 최적의 성공률을 나타내는 특징 벡터 열만을 찾아낸다. SFS를 사용한 최적 특징 벡터의 선정과 실험결과는 4장의 모의 실험에서 자세히 다루기로 한다.

### III. 실험 환경 및 검색 성능 평가 방법

#### 3.1. 실험 환경

실험에 사용된 음악 DB는 인터넷 전문 음악 사이트나 CD로부터 Classic, Hiphop, Jazz, Rock 등 4개 음악 장르에 대해 각 장르별로 60곡을 선정하여 총 240개의 음악 파일을 트레인 DB와 질의로 사용하였으며 22050Hz, 16bits, mono 형식으로 변환하였다.

음악은 주로 일정한 시나리오를 가지고 연주되기 마련인데 한 곡 내에서도 음악이 속한 장르의 특성이 잘 나타나는 부분이 있고, 그렇지 못한 부분이 있을 것이라 사료되어, 모의 실험을 통하여 음악의 장르별 특성이 가장 잘 나타나는 부분을 실험한 결과 곡의 처음 시작 부분부터 약 40%~45% 지점에서 가장 좋은 성능을 보였다. 위 결과를 토대로 본 논문에서는 실험을 위한 특징 벡터를 각 음악의 40%지점에서부터 20초를 추출하여 사용하였다.

#### 3.2. 검색 및 성능 평가 방법

제안된 검색 시스템의 성능평가를 위한 구체적인 절차는 다음과 같다. 먼저 각 장르별로 음악 데이터를 랜덤하게 분할하여 70%를 DB로 사용하고 나머지 30%를 음악 질의 데이터로 사용하여 해당 질의 음악을 DB에서 검색하였다. 또한 제안된 DB에서 신뢰성 있는 결과를 도출하기 위해서 위 과정을 100번 반복하여 각 검색 결과를 산술 평균하여 최종 결과를 도출하였다.

해당 질의 음악에 대한 검색은 트레인 DB에 구축되어 있는 음악 데이터와 벡터 공간에서의 상대적인 거리를 비교하여 거리가 가장 가까운 곡이 질의와 같은 곡이라고 판단하게 된다. 보통 유사도 측정에 사용되는 거리 측정 방법은 Euclidean, Mahalanobis, 시타블록, 코사인 거리 등이 있지만 본 논문에서는 가장 보편적으로 많이 사용되고 있는 Euclidean 거리를 계산하여 유사도 측정을 하였다. Euclidean 거리 함수는 다음의 수식 (1) 과 같이 계산할 수 있으며 이 값이 가장 작은 순으로 검색 결과를 응답한다. 수식 (1)에서  $x_i$ 는 질의 특징 벡터, 그리고  $x_j$ 는 트레인 DB내의 각 음악 특징 벡터를 나타낸다.

$$d_{ij} = [(x_i - x_j)^T (x_i - x_j)]^{1/2} \quad (1)$$

검색 시스템의 검색 성공률은 Hit Rate 와 MRR (Mean Reciprocal Rank)[10] 2가지를 사용하였다. Hit Rate는 검색 결과 상위 10곡 안에 질의 음악이 포함되어 있다면 검색이 성공한 것으로 간주하고 실험을 100번 반복하여 성공률을 100분율로 표시한다. MRR은 상대적 거리 측정을 통해 질의에 해당하는 음악이 몇 위로 검색, 제시되었는가를 측정한다. 첫 검색 시험에서 트레인 DB 내의 곡들과 질의 음악과의 Euclidean 거리를 계산하여 거리가 가장 가까운 10곡을 선정하고 질의에 해당하는 곡이 3위로 검색되었다면 검색 결과의 순위 값을 ( $r_i$ ) 3으로 한다. 다음 2번째 검색 시험에서 질의에 해당하는 음악이 2위로 검색 되었다면 순위 값은 2가 되고, 검색 결과가 상위 10위 안에 없다면 0의 값이 배정된다. 이러한 검색 시험을 100번 반복하여 각 순위 값을 수식 (2)와 같이 산술 평균하여 MRR값을 구하게 된다.

$$MRR = \frac{1}{N} \sum_i \frac{1}{r_i} \quad (2)$$

여기서  $r_i$ 는 각 검색 시험에서의 검색 순위 값, 그리고  $N$ 은 전체 검색 시험 횟수 (본 논문에서는 100)를 의미한다. 전체 질의 입력에 대하여 검색 결과가 전부 1순위로 응답 되었다면 MRR은 1이 되고, 최악의 경우 전부 10위 안에 제시되지 못하였다면 0의 값이 된다. 이러한 MRR 평가는 MRR값이 높을수록 검색 성공률이 높다는 것 외에 얼마나 1순위에 가깝게 검색되었는지를 나타내는 것이므로 검색 시스템의 질을 평가하는 좋은 기준으로 사용될 수 있다.

### IV. 실험 결과 및 분석

#### 4.1. 실험 결과

본 논문에서는 다음과 같이 크게 3가지의 실험을 수행하였다. 첫째는 SFS를 이용한 최적 특징 벡터의 선정 및 검색 성공률 비교, 둘째는 임의 질의 구간에 따른 검색 성능 평가, 마지막으로 임의 질의 길이에 따른 성능 평가를 수행하였다.

다음의 그림 1은 2.2절에서 언급한 SFS를 이용한 최적의 특징 벡터 선정 과정을 보이고 있다.

SFS는 총 60차 특징 벡터를 대상으로 먼저 각 특징 계수들을 개별적으로 사용하여 검색을 한 후, 가장 좋은 성공률을 나타내는 특징부터 순차적으로 나열하여 새로운 60차의 특징 벡터 열을 만든다. 다음으로는 첫 번째 특징 벡터의 성분부터 순차적으로 특징 계수의 수를 늘려가면서 최적의 검색 성공률을 나타내는 특징 벡터 열만을 찾아낸다. 그림에서 보듯이 특징 벡터의 수가 1에서 9까지는 Hit Rate가 지속적으로 증가하다가 9~14까지 구간에서는 거의 변하지 않으면서 약 90% 가까운 Hit Rate를 보이는 반면 그 이상에서는 오히려 Hit Rate가 떨어지는 결과를 보이고 있다. SFS 실험 결과 최적의 특징 벡터 열은 (Roll\_m, Cent\_m, Flux\_m, MFCC1\_m, MFCC2\_m, MFCC3\_m, LPC1\_m, LPC2\_m, LPC3m, LPC5\_m)의 10차 벡터이다. 여기서 Roll은 spectral rolloff 계수, Cent는 spectral centroid 계수, Flux는 spectral flux의 계수, MFCC1은 MFCC 1차 계수, LPC1은 LPC 1차 계수 등을 말하며 \_m 은 각 계수의 평균값을 의미한다.

60차 특징 벡터를 모두 사용하였을 때와 SFS 최적화를 거쳐 선정된 10차의 특징 벡터만을 사용하였을 때 각 장르별 Hit Rate와 MRR을 그림 2 에 비교하였다.

그림 2 (a)에서 보듯이 60차 특징 벡터를 모두 사용하여 검색하였을 경우 65%~74%의 범위 내에서 평균 66% Hit Rate 그리고 SFS기법을 이용해 최적화 된 10개의 특징 벡터를 이용한 실험에서는 77%~90%의 범위 내에서 평균 84% Hit Rate를 보여 평균 22% 가까운 성능 향상을 보이고 있다. 또한 비교 그림에서 보듯이 60차 특징 벡터를 사용하였을 경우 Hit Rate가 Classic, Hiphop, Jazz, Rock 순으로 순차적으로 감소하는 경향을 볼 수 있으나, SFS 기법을 사용한 경우 음악 장르에 구분 없이 전반적으로 안정된 Hit Rate 결과를 얻을 수 있는 장점이 있다.

그림 2 (b)는 검색성공률을 수식 (2)의 MRR 값으로

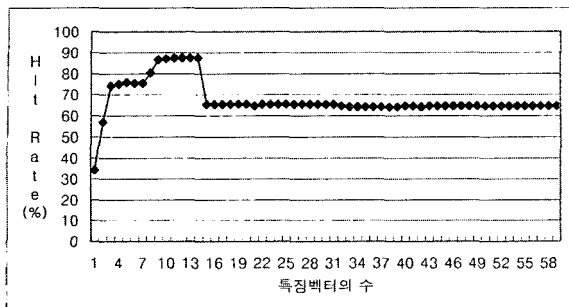
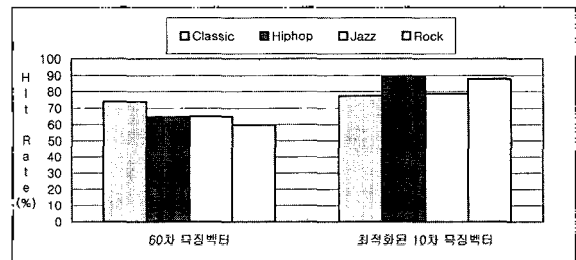


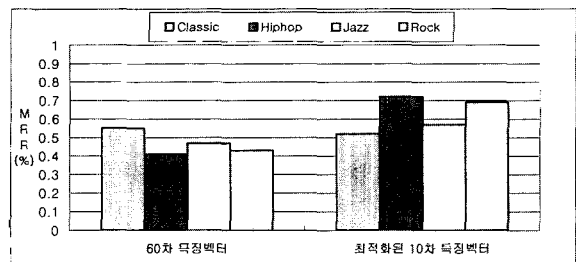
그림 1. SFS 최적 특징 벡터 선정 결과  
Fig. 1. SFS optimum feature vector selection.

계량화 한 결과이다. 실험결과 60차 특징 벡터를 모두 사용하여 검색하였을 경우 0.41 ~ 0.55의 범위 내에서 평균 0.46 MRR을 보이고 있으며, SFS를 적용한 10차 특징벡터의 경우 0.52 ~ 0.72 범위 내에서 평균 0.63 MRR을 나타내었다. MRR은 질의에 해당하는 음악이 몇 위로 검색, 제시되었는가를 측정할 수 있는 방법으로 검색시스템의 질을 평가할 수 있는 중요한 척도이다. 실험 결과 Rock 과 Hiphop 계열 음악의 검색에서는 약 0.7 정도의 높은 MRR로서 검색 결과 1순위 응답이 가장 많음을 알 수 있으며, 반면 Classic 과 Jazz 계열에서는 1 순위 응답 비율이 다소 떨어지는 결과를 보이고 있다. 그러나 Classic의 경우 1순위 검색에 실패하더라도 상위 10곡 중에 해당 장르의 곡들이 주로 검색되는 결과를 보였으며, Jazz의 경우는 Classic과 Rock으로의 오 검색이 두드러졌다. Jazz 내에서도 스윙계열의 검색률은 높았으나, Fusion 계열은 검색률이 20%대의 아주 저조한 결과를 나타내어 최근에 발표되고 있는 Jazz 음악들의 Cross-over 성향을 보여주고 있다. 다음의 표 2에 SFS 최적화를 거쳐 선정된 10차의 특징 벡터를 사용하여 각 장르별 질의에 대한 검색 결과의 예를 나타내었다.

앞의 실험에서는 트레이닝 DB와 질의 입력 음악 특징 벡터를 해당 음악의 특징이 비교적 잘 나타나고 있는 음악 도입 부분부터 약 40% 지점에서 20초를 추출하여 사용하였으나 실제 시스템 사용 환경에서는 사용자들이 음



(a) Hit Rate



(b) MRR

그림 2. 60차 특징 벡터와 최적화된 10차 특징벡터를 이용한 검색 결과 비교

Fig. 2. Retrieval result comparison with 60 dimension feature vector and with SFS 10 dimension feature vector.

표 2. 장르별 질의 예에 대한 검색 결과

Table 2. Retrieval result to the query examples in each genre.

		질 의			
		가브리엘의 오보에(Classic)	웨이래 (HipHop)	Loads of love (Jazz)	Satisfaction (Rock)
유사도 검색 순위	1	룬도	웨이래	In your own sweet way	Satisfaction
	2	가브리엘의 오보에	She Bangs (Spanish)	Autumn leaves	Hard days night
	3	세레나데 (하이든)	The sunshine underground	I left my heart in San Francisco	Rise the shine
	4	세레나데 (슈베르트)	She Bangs	Gunz for sale	Sound of silence
	5	다베르티멘토	웨이래(Remix)	Aguas de Marco	Private eyes
	6	엘리제를 위하여	Loaded	St. Thomas	My generation
	7	행복한 농부	That Thing you do!	Love ballad	You really got me
	8	세레나데 작품3번	Runaway	Caravan	Sign
	9	멜로디	Fuel	Hush a bye	Be bop a lula
	10	미뉴엣	Freak of the week	Spear head	Dreams

악의 어느 부분을 질의로 사용할지는 알 수 없다. 따라서 실제 적용 가능한 시스템 구축을 위해서는 음악 파일의 처음 도입부분부터 끝 구간까지 임의의 20초 분량의 오디오 클립 질의에 대하여 신뢰성 있게 동작할 수 있는 강인한 시스템 구축이 중요하다. 이를 위해 본 논문에서는 음악 파일의 처음 시작부분, 음악 파일의 10%, 20%, 30%, 45%, 50% 그리고 80%되는 지점에서 20초 분량의 질의 음악을 추출하여 실험을 수행하였다. 그림 3은 각 장르별 임의 질의에 대한 Hit Rate을 나타내고 있다. 실험 결과 음악 파일의 40%지점에서 특징을 추출한 트레이닝 DB와 가장 근접한 지점에 (45%지점) 해당하는 질의에 대해서는 90% 이상의 성공률을 보였으나, 음악의 시작부분이나 끝부분으로 갈수록 Hit Rate가 떨어지는 결과를 보였다. 이는 일반적으로 음악의 도입부분과 끝부분에 해당 음악적 특징이 잘 나타나지 않는 특성에 기인한다 할 수 있다. 하지만 실제 검색 시스템은 모든 임의 질의 구간에 대하여 안정되게 동작하여야 함으로 이러한 시스템 불안정성은 반드시 해결되어야 할 문제라고 할 수 있다.

전 실험에서는 20초 길이의 음악 클립을 질의로 사용

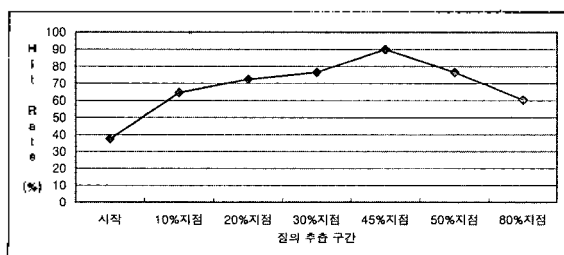


그림 3. 특징벡터 추출 구간 별 검색성공률  
Fig. 3. Retrieval success rate depending on query position.

하였으나 검색 시스템의 실용성을 고려했을 때 가능한 짧은 길이의 질의 사용이 바람직하다. 이를 위해 본 실험에서는 5초, 10초, 15초, 20초, 25초 그리고 30초의 6가지 길이로 변화를 준 임의 질의 클립을 사용하여 실험을 수행하였다. 그림 4의 실험 결과 트레이닝 DB 특징 벡터 추출 길이와 같은 20초 질의의 경우 80% 정도의 Hit Rate을 보였고, 20초 보다 긴 25초와 30초 질의의 경우는 더욱 좋은 성능을 보였다. 하지만 20초 이하의 질의 길이에서는 Hit Rate가 70% 이하로 검색 시스템의 성능이 급격히 떨어짐을 볼 수 있다. 그림 4의 결과로 보아서 20초 정도 길이의 질의가 처리속도와 검색 성공률 면에서는 적당하다고 볼 수 있으나 실제 시스템 환경을 고려 해볼 때 10초 미만의 질의를 사용하면서도 안정된 성능을 가져올 수 있는 시스템 구축이 요구된다.

#### 4.2. 기존 논문과의 비교 분석

본 논문에서 사용된 SFS 특징 벡터 최적화 기법은 10개 정도의 특징 벡터를 사용하면서도 평균 84% Hit Rate 와 0.63 MRR을 달성 할 수 있었다. Hit Rate 즉 검색성공률 측면에서 기존의 G. Tzanetakis[3]의 74%

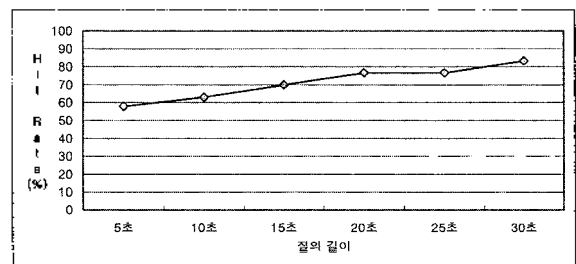


그림 4. 질의 길이에 따른 검색 성공률  
Fig. 4. Retrieval success rate depending on query length.

Hit Rate와 비교해 보면 약 10% 정도의 검색 성공률 향상이 있음을 알 수 있다. 한편 MRR은 질의에 해당하는 음악이 몇 위로 검색, 제시되었는가를 측정할 수 있는 방법으로 비교 대상으로 삼을 수 있는 QBE 방식의 검색 시스템은 없지만 논문 [10]의 QBH 방식의 검색 시스템에서 피치 정보를 이용하여 달성한 0.60 값과 비교하면 약 0.03 정도의 향상이 있어 검색 결과 1순위 응답 비율이 약간 높음을 알 수 있다.

본 논문에서 수행한 임의 질의 구간에 대한 실험은 현재까지 비교 대상의 논문은 없으나 서로 다른 질의 구간에 대한 검색 성능의 불안정성을 지적한 것으로 실용성 있는 검색 시스템 구축을 위해서는 반드시 선행 연구되어야 할 과제이다. 또한 임의 질의 길이에 대한 실험 결과에서도 실제 시스템 환경을 고려 해볼 때 가능한 짧은 길이의 질의를 사용하는 것이 바람직하나 질의 길이가 20초 이하인 경우 검색 성능이 급격히 떨어지는 문제를 해결 해야 하며 논문 [11]에서도 동일한 결론을 내리고 있음을 알 수 있다.

## V. 결론

본 논문에서는 Classic, Hiphop, Jazz, Rock 4개의 장르로 음악 데이터를 구분하여 각 장르별 60곡씩 총 240곡의 음악 DB를 대상으로 예제 질의 (QBE) 방식의 내용 기반 음악 정보 검색 시스템을 제안하였다. 본 시스템은 입력 음악 질의로부터 디지털 신호처리 기법을 이용하여 centroid, rolloff, flux 등의 STFT 기반의 특징들과 MFCC, LPC, Beat 정보 등의 총 60차 특징 벡터들을 추출한 후 트레인 DB와의 Euclidean 유사도를 측정하여 해당 음악을 검색한다. 검색에 사용되는 특징 벡터는 SFS 기법을 사용하여 최적화시킨 10차 특징 벡터를 사용하였으며 검색 결과는 평균 84% Hit Rate, 0.63 MRR의 성공률을 보이고 있어 기존의 연구 결과보다 평균 10% 이상의 성능 향상을 보이고 있다. 한편 본 논문에서는 실제 시스템 사용 환경을 고려하여 임의 질의 구간과 임의 질의 길이에 대한 시스템 성능 평가를 수행하였으며 실험 결과 이러한 임의성에 기인한 검색 성능의 불안정성을 지적하였다. 향후 연구과제로는 앞서 지적한 문제를 해결할 수 있는 강인한 내용 기반 음악검색 시스템에 대한 심도 있는 연구를 계획하고 있다.

## 감사의 글

본 연구는 한국과학재단 목적기초연구(R01-2004-000-10122-0) 지원으로 수행되었음.

## 참고 문헌

1. E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," IEEE Multimedia, 3(2), 1996.
2. T. Zhang and C. -C. Jay Kuo, "Hierarchical System for Content-based Audio Classification and Retrieval," Proceedings of SPIE's Conference on Multimedia Storage and Archiving Systems III, SPIE, 3527, pp. 398-409, Boston, Nov., 1998.
3. G. Tzanetakis, Manipulation, Analysis and Retrieval Systems for Audio, PhD Thesis in Computer Science from Princeton University, June, 2002.
4. P. G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machine," IEEE Trans. on neural networks, 14(1), 209-215, Jan. 2003.
5. A. Ghias, J. Logan, D. Chamberlin, and B. Smith, "Query by Humming: Musical Information Retrieval in an Audio Database," ACM Multimedia, 213-236, 1995.
6. Melucci, M. and Orio, N., "Musical Information Retrieval using Melodic Surface," Proceedings of the fourth ACM conference on Digital libraries, pp. 152-160, August 1999.
7. R.J. McNab, L. Smith, I.H. Witten, and C.L. Henderson, "Tune Retrieval in the Multimedia Library," Multimedia Tools and Applications, 10, 113-132, 2000.
8. Lutz Prechelt and Rainer Typke, "An Interface for Melody Input," ACM Transactions on Computer-Human Interaction, 8(2), pp. 133-149, June 2001.
9. J. J. Burred and A. Lerch, "A hierarchical approach to automatic musical genre classification," in Proc. DAFx03, 2003, 308-311.
10. S. Doraisamy and S. Ruger, "Robust polyphony music retrieval with N-grams," Journal of Intelligent Information Systems, 21(1), 53-70, 2003.
11. 조용준, 김광희, 이배호, "효율적인 웨이블릿 기반 오디오 데이터 검색 시스템의 구현," 한국음향학회지, 21(1), 82-88, 2002.

## 저자 이력

### • 윤원중 (Won-Jung Yoon)

1977년 3월 25일생  
2003년 2월: 상명대학교 정보통신학과 (이학사)  
2003년 3월~현재: 단국대학교 컴퓨터과학 및 통계학과 (석사과정)

### • 이강규 (Kang-Kyu Lee)

1978년 2월 17일생  
2003년 2월: 상명대학교 정보통신학과 (이학사)  
2003년 3월~현재: 단국대학교 컴퓨터과학 및 통계학과 (석사과정)

• 박 규 식 (Kyu-Sik Park)

1963년 1월 28일생

1986년 2월: Polytechnic University 전자공학과 (공학사)

1988년 2월: Polytechnic University 전자공학과 (공학석사)

1993년 2월: Polytechnic University 전자공학과 (공학박사)

1994년1월~1996년 2월: 삼성전자 마이크로사업부, 선임 연구원

1996년 3월~2001년 8월: 상명대학교 컴퓨터·정보통신공학부 조교수

2001년 9월~현재: 단국대학교 정보컴퓨터학부 부교수