

내용 기반 음원 검출 시스템 구현에 관한 연구

허태관[†], 조황원^{**}, 남기표^{***}, 이재현^{****}, 이석필^{*****}, 박성주^{*****}, 박강령^{*****}

요 약

현재 우리는 언제 어디에서나 다양한 음악을 들을 수 있다. 길거리나 카페 등에서 음악을 듣고 그 노래를 다시 듣고 싶어 찾으려 할 때 해당 음악의 가수나 노래 제목을 모른다면 그 음악을 찾을 수 없는 것이 현재 음악 검색 방식의 단점이다. 이러한 문제를 해결하기 위하여 본 연구에서는 음원의 음 정보를 추출하여 데이터베이스를 구축하고 찾고자 하는 음악의 허밍을 녹음하거나 듣고 있는 음악을 직접 녹음하여 검색하는 방법 및 전자 악기를 연주하여 검색하는 내용 기반의 검색 방법을 연구하였다. 본 연구에서는 기존의 많은 내용 기반 음원 검색 논문, 특히 딥 시스템을 분석하였으며, 이를 기반으로 내용 기반 검색 방법을 연구하였다. 즉, 허밍과 음악을 직접 녹음하여 검색하는 경우 음과 음원을 고속 푸리에 변환하여 주파수 정보를 추출하였으며, 악기를 연주하여 검색하는 경우 midi 파일을 사용하였다. 그리고 동적 정합 매칭 방식을 사용하여, 매칭 시 음원과 입력 음간의 길이차를 극복하였다.

A Study on the Implementation of the System of Content-based Retrieval of Music Data

Taikwan Hur[†], Hwangwon Cho^{**}, Gi Pyo Nam^{***}, Jaehyun Lee^{****},
Seok-Pil Lee^{*****}, Sung-Joo Park^{*****}, Kang Ryoung Park^{*****}

ABSTRACT

Recently, we can hear various kinds of music in everywhere and anytime. If a user wants to find the music which was heard before in a street or cafe, but he does not know the title of the music, it is difficult to find it. That is the limitation of previous retrieval system of music data. To overcome these problems, we research a method of content-based retrieval of music data based on the recorded humming, the part of recorded music and the played musical instrument. In this paper, we investigated previous content-based retrieval methods of papers, systems and patents. Based on that, we research a method of content-based retrieval of music data. That is, in case of using the recorded humming and music for query, we extract the frequency information from the recorded humming/music and the stored music data by using FFT. We use a MIDI file in case of query by the played musical instrument. And by using dynamic programming matching, the error caused by the disparity of length between the input source with the stored music data could be reduced.

Key words: Retrieval of Music Data(음원 검색), Content-based Retrieval(내용 기반 검색), FFT(FFT), DP Matching(동적 정합 매칭)

※ 교신저자(Corresponding Author) : 박강령, 주소 : 서울시 중구 필동 3가 26(100-715), 전화 : 010-3111-7022, FAX : 02)2277-8735, E-mail : parkgr@dongguk.edu
접수일 : 2009년 7월 2일, 수정일 : 2009년 8월 3일
완료일 : 2009년 9월 28일

[†] 동국대학교 전자공학과 공학사
(E-mail : 1982-12-15@hanmail.net)

^{**} 동국대학교 전자공학과 공학사
(E-mail : cchwchw@hanmail.net)

^{***} 동국대학교 전자공학과 석박사통합과정
(E-mail : oscar1201@dongguk.edu)

^{****} 동국대학교 전자공학과 공학사
(E-mail : isk8ers@nate.com)

^{*****} 전자부품연구원 디지털미디어연구센터 센터장
(E-mail : lspbio@keti.re.kr)

^{*****} 전자부품연구원 디지털미디어연구센터 선임연구원
(E-mail : bpark@keti.re.kr)

^{*****} 정희원, 동국대학교 전자공학과 부교수

※ 이 논문은 2009년도 정부(지식경제부)의 재원으로 정보통신 산업원천기술사업의 지원을 받아 수행된 연구임(No. 2009-S-001-01)

1. 서 론

인터넷과 정보 통신 기기들의 발달로 우리는 언제 어디서나 쉽고 간편하게 수많은 음악들을 접할 수 있게 되었고, 디지털 음원에 대한 대중들의 관심이 커짐에 따라 끊임없이 새로운 많은 곡들이 쏟아져 나오고 있다. 이처럼 새로운 음원들은 증가하고 있지만, 그것을 찾는 방법은 여전히 한정되어 있다. 사용자가 원하는 음악을 찾을 때 음원의 제목이나 가수, 작곡자 등의 정보를 모르는 경우 수많은 노래들 중에서 검색한다는 것이 불가능하다. 그래서 이런 경우에도 원하는 음원을 검색할 수 있게 하기 위해서는 음원의 내용을 기반으로 한 음원 검색 방식이 필요하며, 이러한 검색 방식을 내용 기반 검색(content-based retrieval)이라고 한다.

내용기반 검색을 위해서는 음원의 내용에 대한 특징을 추출하는 작업이 우선된다. 음원의 내용 특징을 추출하는 방법으로 본 연구에서는 두 가지 방법을 사용하였다. 첫 번째는 미디 (musical instrument digital interface (MIDI))파일에서 음악의 음 정보인 노트(note) 값을 추출하는 방법이며, 두 번째는 웨이브 파일에서 음악의 주파수 정보를 추출하는 방법이다. 또한 두 가지 방법을 이용하여 검색의 대상이 되는 음원의 데이터베이스를 구축하였다.

본 연구에서는 내용기반 검색을 위해 사용자가 입력할 수 있는 내용 샘플을 만드는 방법으로는 다음과 같은 세 가지 방법을 시도하였다. 사용자가 해당 음원의 일부분을 직접 허밍 하는 방법, 직접 녹음하는 방법 그리고 직접 연주하는 방법이다. 앞의 두 가지 방법으로부터 추출한 특성들은 웨이브 파일로 저장하여 사용하였고, 연주하는 음에 대한 특성은 추출하여 미디 파일로 저장함으로써 사용하였다. 추출된 내용 샘플의 특성을 이용하여 음원을 찾는 방법으로는 기존에 유사도 측정 방법으로 많이 사용하는 동적 정합 매칭 방법(dynamic programming matching)을 사용함으로써, 입력 샘플과 가장 유사한 음원을 결정하였다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 음원 검색 방법에 대한 설명 및 3장에서는 본 연구에서 제안하는 음원 검색 방법에 대한 설명을 포함하였다. 그리고 4장과 5장에서는 실험결과 및 결론을 포함하였다.

2. 기존 관련 연구

노승민 등은 다음과 같이 기존에 음원 정보 검색 연구 및 시스템들을 소개하였다[1]. Ghias등은 QBH (query by humming) 시스템을 개발하였는데[2], 이 시스템에서는 마이크로폰 (microphone)을 통해 사용자의 허밍이나 휘파람을 입력받고, 이로부터 음높이 변화(pitch contour)를 특징을 추출하여 이를 하기 그림 1과 같은 UDR 스트링으로 표현하고 음악 데이터베이스의 콘텐츠와 비교 검색하는 기능을 제공한다. McNab 등은 MELDEX 시스템을 개발하였다[3,4]. 이 시스템에서는 마이크로폰을 통해 입력된 허밍으로부터 추출된 멜로디의 음높이 변화나 음의 길이를 이용하여, 그림 1과 같은 UDR 스트링 매칭을 하고, 동적 정합 매칭 알고리즘을 이용한 검색을 수행하였다. Typke 등은 MelodyHound를 개발하였다[5]. 이 시스템 역시 사용자의 휘파람을 통한 입력으로부터 하기 그림 1과 같은 UDR 스트링을 얻어내고 이를 음원 검색에 이용하였다. Kornstadt 등은 Themefinder 시스템을 개발하였다[6,7]. 이 시스템은 기존의 시스템과는 다르게 16세기의 서양 클래식 음악, 포크송들을 대상으로 Humdrum이라고 하는 명령을 사용하여 사용자가 원하는 곡의 테마를 웹상에서 찾을 수 있는 기능을 제공하였다. Blackburn 등은 자신의 연구에서 음악 데이터가 구조적이고 분기하는 구조를 가지기 때문에 한 음악에서 다른 음악의 특정 부분으로 연결할 수 있다는 가정을 사용하였다[8]. 이를 기반으로 멜로디 음색의 변화를 이용한 내용 기반 향해 기능을 제공하였다. 하기 표 1은 이러한 기존 연구들을 정리한 것이다[1].

이외에도 기존에 내용 기반 음원 검색 관련 국내외 특허들이 출원 및 등록 되었다[14-22]. [14]의 연구에서는 기 저장되어 있는 곡으로부터 주 멜로디의 윤곽 특징을 추출하고, 추출한 특징에 바 형태의 분할을 수행한다. 이후 분할된 특징들을 프레임 단위의 음표 시퀀스로 변환하여 템플릿으로 저장한다. 매칭을 위해서는 노래/허밍으로부터 추출된 프레임 단위



그림 1. 동일한 UDR 스트링을 가지는 두 소절

표 1. 기존 음악 정보 검색 연구

가존 시스템	개발자/개발년도	음악포맷	추출되는 특징들
QBH System[2]	Cornell Univ.(Ghias) / 1995	미디 / 단성음악	음조 곡선(Contour)
MELDEX[3,4]	Univ. of Waikato (McNab, et al / 1997	미디 / 단성음악	음조곡선, 음정(Interval), 음길이(Duration)
MiDiLiB-PROMS [9]	Univ. of Bonn (Clausen) / 1998	미디 / 단성음악	음조 곡선
Themefinder [6,7]	Stanford & Ohio State University (Huron, et al) / 1998	미디, Humdrum/ 단성음악	음조곡선, 음정, 음계(Scale)
CBR and Navigation Music [8,10]	Univ. of Southampton (Steven G. Blackburn) / 1998	미디 / 단성, 다성음악	음조곡선
Melodic Matching Techniques [11,12]	Univ. of Australia (Uitdenbogerd & Zobel) / 1999	미디 / 단성음악	음조곡선
MelodyHound [5]	Univ. of Karlsruhe (R. Typke) / 2001	단성음악	음조곡선, 음길이
Fast Melody Finder [13]	Ajou University (S. Rho. et al) /2003	미디 / 단성음악	확장된 음조곡선, 음정, 음길이, 음계

의 질의 시퀀스, 바 형태로 분할된 템플릿 시퀀스를 입력하고, 상기 질의 시퀀스와 상기 템플릿 시퀀스 사이에 매칭을 수행하여 매칭 결과에 따라 질의 후보들을 출력한다. [15]의 연구에서 멜로디는 인접한 두 노트의 상대적인 값으로 표현되는데, 데이터베이스에 저장될 멜로디의 형태로써 음높이와 음길이의 정보가 사용되며, 음높이는 인접한 두 노트 간의 음높이의 차이로 변환하고, 음길이는 인접한 두 노트의 음길이(duration)의 비로 변환한다. 또한, 가이드 멜로디의 두 음 사이 발생 시간 간격이 일정 마디를 초과하는 경우, 사용자가 질의 입력을 연속적으로 할 가능성이 매우 낮아지는 것으로 판단하여, 인접한 두 음간의 시간을 고려하여 휴지(pause) 정보를 가이드 멜로디 내에 삽입함으로써 매칭의 정확성을 높이는 방법을 채택하였다. 그리고 이 연구에서는 유사 매칭의 개념을 적용하여 허밍과 음악 멜로디 간의 매칭이 가능하도록 설계하였으며, 유사매칭 방법으로는 편집 거리(edit distance)와 동적 정합매칭 방법을 사용하였다.

[16]의 연구에서는 서버-클라이언트 기반 음악 질의 및 음원 검색 시스템을 소개하였다. 이 연구에서는 입력된 심벌 멜로디 시퀀스와 메타데이터 데이터베이스에 있는 각 오디오의 특징 값과의 유사도를 계산하는 멜로디 유사도 계산기를 제안하였다. 이때, 유사도는 유클리디언 (euclidean) 거리나 절대 차이 값의 합(sum of absolute difference) 등을 기반으로

한 동적 정합 매칭 방법을 이용하여 구하였다. 이후, 거리 기반 분류기에서는 계산된 거리 정보들을 거리 값이 작은 순서대로 정렬하여 해당하는 메타데이터 정보를 메타데이터 데이터베이스에서 추출한 후, 검색된 결과 정보를 인터넷을 통해 클라이언트 측으로 전달하는 방식을 사용하였다.

[17]의 연구에서는 입력 음악 정보로서 허밍 멜로디를 사용하였다. 입력된 허밍으로부터 노트를 검출하고, 음 길이와 신뢰도 측정에 따른 다중 피치 특징을 추출하였다. 유사성(similarity) 측정 엔진에서는 추출된 특징을 이용하여 허밍과 데이터베이스 사이의 유사도 값을 측정하였다. 그리고 질의(query)엔진은 데이터베이스에 저장된 음원에 해당되는 허밍을 매칭 시키기 위하여 동적 정합을 수행하고, 그 결과로서 가장 작은 매칭 거리를 가지는 멜로디를 선택하는 방식을 사용하였다. 특히 이 연구에서는 매칭 시 3차원으로 확장된 동적 매칭 정합을 사용하여 모든 후보 점에서의 조합을 계산하고, 최적 점과 경로를 결정하였다.

[18]의 연구에서는 멜로디의 간격 및 멜로디 간격의 방향인 피치 윤곽(pitch contour)으로 노래를 표현하였다. 이 연구에서는 세 가지 상태의 허밍 기반 검색 시스템을 제안하였는데, 이러한 시스템을 UDS (Up-Down-Same) 시스템이라고 명칭 하였다. 일반적으로 사용자가 허밍할 때는 정확하게 노래를 표현할 수 없고 어느 정도 에러를 포함하게 된다. 따라서

정확하지 않은 허밍에 의한 멜로디를 표현할 수 있도록 제안된 것이 UDS 시스템이다. 이 시스템에서는 허밍한 음들을 U, D, S의 문자열들을 사용하여 멜로디 윤곽 (melody contour)으로 표현하였다. 'U(up)'는 상승 구간 (ascending intervals), 'D(down)'은 하강 구간 (descending intervals), 'S(same)'은 널 구간 (null interval) 을 나타낸다. 데이터베이스에 저장된 노래들 역시 U, D, S의 문자열들로 표현해서 입력된 허밍과 비교하였다. 비교를 위해서는 비터비 알고리즘 (viterbi algorithm)과 역방향추적 (back tracking) 방법을 이용하였으며, 허밍으로 입력된 데이터와 저장되어있는 데이터를 비교한 후 가장 거리값을 가지는 데이터를 선택하는 방식으로 매칭을 수행하였다.

[19]의 연구에서도 음악 질의를 사용하여 데이터베이스 내의 음원을 검색하였다. 즉, 사용자가 키보드의 키를 기반으로 음악 노트를 선택하고, 노트의 피치와 구간을 기록하였다. 그리고 노트의 피치와 구간의 유사도를 기반으로 매칭을 수행하였다. [20]의 연구에서 사용한 중요한 특징은 instrumentation과 beat이다. instrumentation 측정을 위해서 EMD (earth movers distance)라는 전역적 거리 측정방법을 사용하였고, beat 측정을 위해서는 rhythmic 기반 거리를 사용하였다. 최종적으로 이 두 거리 값에 가중치를 주어서 유사도를 측정하였다. [21]의 연구에서는 입력된 멜로디의 주파수 스펙트럼 에너지 분포 (spectrum energy distribution)의 변화와 차이점들을 이용하여 분기점을 설정하고 이 점을 기준으로 입력된 멜로디를 뚜렷한 음들로 변환하거나 또는 음 높이들의 변화들로 나타냈다. 매칭시 신뢰도 레벨은 피치의 변화, 스펙트럼 에너지 분포 지시값 및, 입력된 멜로디의 에너지 레벨에 따라서 결정하였다. 매칭 과정은 노트들을 삽입 (insertion), 제거 (deletion) 할 때의 에러를 최소화하는 방식으로 비용함수 (cost function)을 정의하여 수행하였다. [22]의 연구에서는 데이터베이스에 저장된 음악에서 상대적 피치 시퀀스와 상대적 구간 간의 비에 대한 히스토그램을 기반으로 음 높이와 길이의 상이도를 측정하였다.

이외에도 기존 관련 연구에서는 내용 기반 음원 검색 시 그림 1과 같은 음높이의 변화를 이용한 세 가지 타입의 UDR 스트링 비교 기법을 사용하였다 [1].

하지만, 그림 2와 같이 UDR 스트링은 전혀 다른



그림 2. 동일한 LSR 스트링을 가지는 두 소절

원음	62	60	70	62	67	67
앞뒤 음의 차	-2	+10	-8	+5	0	
첫 음과의 차	-2	+8	0	+5	+5	

그림 3. MIDI 노트 차이를 이용한 분석

음정을 가지는 음악 파일이나 질의를 동일하게 인식하는 문제점이 있어서, 기존의 UDR 스트링을 세분화하여 두 번째 소절과 같이 음정 (interval)이 급격하게 높아지거나 (Up a lot), 낮아지거나 (Down a lot) 하는 경우를 추가하여 u, U, d, D, r과 같이 세분화하여 사용하였다. 또한, 음의 길이 정보는 Longer, Shorter, Repeat의 세 가지 타입으로 구분하여 L, S, R의 스트링으로 표현하였다[1].

하지만, 이처럼 음정의 변화와 음의 길이를 이용하더라도 두 소절의 정확한 차이를 구분할 수 없는 경우가 발생한다. 특히, 음원과 동일한 입력 음사이의 길이차가 발생하는 경우, UDR 스트링 기반의 정확매칭 방식에서는 에러가 증대되는 문제점이 있다.

그러므로 본 논문에서는 위와 같은 문제점들을 해결하기 위하여, wave 파일에서 추출된 기본 주파수 (fundamental frequency) 값들을 특징으로 이용하였고, 음원과 동일한 입력 음사이의 길이 차 문제를 해결하기 위하여 동적 정합 매칭과 같은 유사 매칭 기법을 사용하였다. MIDI 파일을 이용한 검색 시에는 악기 연주 시 음 높이를 정확히 맞추기 힘들기 때문에 그림 3과 같은 앞 뒤 음의 차이 혹은 첫 음과의 차이를 이용하였다. 이에 대한 보다 자세한 설명은 3장에 나와 있다.

3. 본 론

3.1 전체 구성도

본 연구에서는 그림 4와 같은 방식으로 내용 기반 음원 검색 알고리즘을 구현하였다. 우선 음원으로부터 음의 정보를 추출하여 데이터베이스를 구축한다. 그리고 입력 데이터로부터 음 정보를 추출한 후 동적 정합 방식을 사용하여 검색한다.

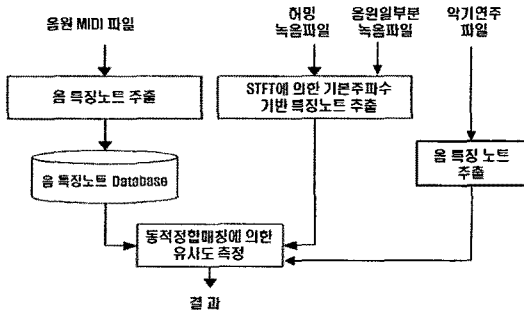


그림 4. 시스템 개략도

3.2 특징 추출

3.2.1 Wave 파일로부터 특징 추출

본 연구에서는 사용자의 허밍 입력이나 음원의 일부분을 녹음하여 미리 저장된 음원과 비교하는 연구를 수행한다. 이때 사용자의 허밍 입력이나 음원의 일부분 녹음은 wave 파일로 저장한다. Wave 파일은 음원의 정보를 PCM (pulse code modulation) data로 저장하고 있는 파일 포맷이다. 이 wave 파일로부터 주파수 정보를 추출하기 위해서 본 연구에서는 다음 수식 (1)과 같은 STFT (short time fourier transform)을 사용한다.

$$STFT(x[n]) = X(m, w) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (1)$$

식 (1)에서 $x[n]$ 은 입력 신호를 나타낸다. $w[n-m]$ 은 짧은 구간별 STFT을 수행하기 위한 윈도우 (window)를 나타낸다. STFT란, 짧은 시간 단위로 푸리에 변환을 반복 수행하는 과정으로, 본 연구에서는 125ms를 한 시간단위로 잡고 wave 형식의 음원에 대해 STFT를 수행한다. STFT 수행 후 한 시간 단위 내에 배음(harmonics) 구조가 여러 개 존재하는 경우, 최대 크기 (amplitude) 값을 갖는 기본 주파수(fundamental frequency)라 하여 이를 특징으로 사용하였다. 예를 들어 100Hz, 200Hz, 300Hz, 400Hz ... 와 70Hz, 140Hz, 210Hz, 280Hz... 와 같이 harmonics구조가 여러 개 존재할 때, 100Hz 와 70Hz의 amplitude를 비교하여 보다 큰 것을 기본주파수로 사용하였다. 그림 5는 추출된 기본 주파수의 예를 나타낸 것이다. 그림 5의 경우 440Hz, 880Hz, 1,320Hz... 와 670Hz, 1,340Hz, 2,010Hz... 와 같은 두 개의 배음 구

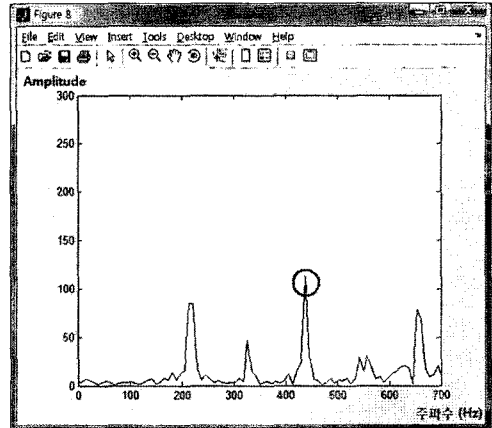


그림 5. 단위 시간 내 최대 크기를 갖는 기본 주파수의 예
조가 존재했으며, 이 중 amplitude가 보다 큰 440Hz가 기본 주파수 특징으로 사용되었다. Matlab program을 이용하여 구현하였다.

추출된 기본 주파수 값을 보고 주파수의 변화는 알 수 있었지만, 해당 기본 주파수 값이 어느 음에 해당하는지를 파악할 수 없어서, 하기 표 2와 같은 음계 주파수 테이블을 참고하였다[23].

그림 6은 입력 wave파일 및 저장된 음원에서 추출된 기본 주파수 값의 예시이다.

3.2.2 미디 파일로부터 특징 추출

미디 파일은 사운드 자체를 녹음한 다른 음악 파일들과 달리, 디지털 신호화된 연주 정보를 가지고 있고 그 정보를 분석하여 사운드 카드로 음악을 재생

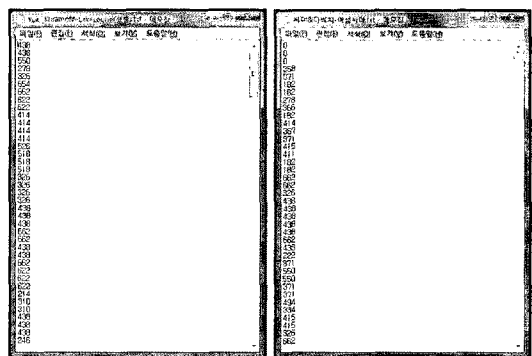


그림 6. 기본 주파수 값 추출 예. (a) 입력 wave 파일에서 추출된 기본 주파수, (b) 저장 음원에서 추출된 기본 주파수

표 2. 음계 주파수 테이블

A0	A0#	B1	C1	C1#	D1	D1#	E1	F1	F1#	G1	G1#
27.5	29.14	30.87	32.7	34.65	36.71	38.89	41.2	43.65	46.25	49	51.91
A1	A1#	B1	C2	C2#	D2	D2#	E2	F2	F2#	G2	G2#
55	58.27	61.74	65.41	69.3	73.42	77.78	82.41	87.31	92.5	98	103.83
A2	A2#	B2	C3	C3#	D3	D3#	E3	F3	F3#	G3	G3#
110	116.54	123.47	130.81	138.59	146.83	155.56	164.81	174.61	185	196	207.65
A3	A3#	B3	C4	C4#	D4	D4#	E4	F4	F4#	G4	G4#
220	233.08	246.94	261.63	277.18	293.66	311.13	329.63	349.23	369.99	392	415.3
A4	A4#	B4	C5	C5#	D5	D5#	E5	F5	F5#	G5	G5#
440	466.16	493.88	523.25	554.37	587.33	622.25	659.26	698.46	739.99	783.99	830.61
A5	A5#	B5	C6	C6#	D6	D6#	E6	F6	F6#	G6	G6#
880	932.33	987.77	1046.5	1108.7	1174.7	1244.5	1318.5	1396.9	1480	1568	1661.2
A6	A6#	B6	C7	C7#	D7	D7#	E7	F7	F7#	G7	G7#
1760	1864.7	1975.5	2093	2217.5	2349.3	2489	2637	2793.8	2960	3136	3322.4

하는 방식을 사용한다. 그러므로 미디 파일은 용량이 작고 정확한 음 정보를 얻을 수 있기 때문에 대부분의 오디오 검색 시스템이 미디 파일을 이용하여 오디오 특징 정보를 추출한다. 표 3은 각 음에 해당하는 노트번호를 나타낸다[24].

미디 파일은 여러 악기를 하나의 트랙에 기록하는 포맷 (format) 0 (연주 용), 여러 악기를 여러 트랙에 기록하는 포맷 1 (악보 편집 용), 포맷 1과 같으나 동시에 연주하지 않는 포맷 2, 이렇게 3가지 포맷이 있다. 크게 헤더 청크와 트랙 청크로 나눌 수 있다. 헤더 청크에는 미디 파일의 포맷 종류, 트랙의 수, 헤더의 크기, 4분음표의 길이와 같은 파일의 기본 정보를 담고 있다. 트랙 청크에는 앞쪽에 악보에 대한 정보를 가지고 있는 메타 이벤트가 있고, 그 뒤에 연주에 대한 정보를 가지고 있는 미디 이벤트가 있다. 메타 이벤트는 “<Variable Length Delta-Time> +

<FF> + <Meta-Event 명령> + <Parameter>”와 같은 형식으로 되어 있고, 미디 이벤트는 “<Variable Length Delta-Time> + <Event 명령> + <기타 파라미터>”와 같은 형식으로 되어 있다. 이 때 하나의 값은 1 바이트 (byte)로 되어 있고 메타 이벤트와 미디 이벤트를 구분하는 것은 <FF>이다. 델타 타임은 앞의 이벤트를 처리하고 얼마나 기다려야 하는지를 나타내고, 가변 길이로 델타 타임 자리의 값이 1xxxxxxx 이면 다음 정보도 델타 타임이고, 0xxxxxxx일 경우 델타 타임 값이 끝난다. 그러므로 1xxxxxxx 0xxxxxxx 의 경우 1과 0을 제외한 14개의 x값이 델타 타임 값을 나타낸다. 그러므로 어느 바이트가 델타 타임인지 잘 골라서 읽어야 한다. 미디 파일의 약점은 중간에 에러가 발생하면, 나머지 뒷부분을 읽기 어렵다는 점이다[25]. 본 연구에서 사용한 미디 파일은 모두 포맷 1이다. 트랙 중에서도 첫 번째 트랙에는 파일에 대한 정보만 가지고 있고, 두 번째 트랙부터 트랙마다 각각의 악기 연주 정보를 가지고 있다. 이 때 여러 트랙 중에 어느 트랙에 우리가 원하는 멜로디 정보를 가지고 있는지 알 수 있는 방법이 없다. 그러나 미디 파일로부터 악보를 보여주는 프로그램을 이용하여 분석 한 결과 많은 노래들이 두 번째 트랙(악기 정보를 가진 트랙 중 첫 번째 트랙)에 우리가 원하는 멜로디를 저장하였고, 그렇지 않은 노래들은 파일을 편집하여 두 번째 트랙으로 이동시켜 음 정보 추출 시간과 용량을 줄여주었다.

표 3. 각 음에 해당하는 노트 번호

Octave #	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

그리고 미디 파일 형태가 노트의 On/Off가 반복된 형태와 running status로 이루어진 두 가지 형태로 되어 있어 두 가지 형태 각각에 맞는 추출 방식을 사용하였다. 또한 일반적으로 음악 검색 시 음악 전곡이 필요하지 않고 1절 정보만 있어도 충분히 검색할 수 있으므로, 추출 값 수를 250개로 한정하여 용량, 속도를 줄여주었다. 음 노트 값을 250개로 추출하였을 시 가장 긴 노래도 1절 이상의 값을 얻을 수 있었고 짧은 노래는 노래 전체의 값을 얻은 것을 확인하였다.

3.3 매칭 (Matching)

일반적으로 입력 음악 정보($R = [r_0, r_1 \dots r_{NR-1}]$)와 저장된 음원($Q = [q_0, q_1 \dots q_{NQ-1}]$)사이에는 동일 음악이라고 할지라도, 길이차가 발생하므로, 본 연구에서는 하기 그림 8과 같이 동적 정합 매칭 방법을 사용하였다[26].

동적 정합 매칭은 그림 8과 같이 서로 길이가 다른 두 패턴간의 유사도를 측정하기 위한 방법으로, 삽입(insertion) 및 제거(delete)등의 방법으로 경로 별 패턴 유사도를 측정할 수 있는 방법이다[26]. 일반적으로 허밍과 같은 입력 음악 정보는 저장된 음원 데이터의 임의의 부분에 대한 정보를 포함할 수 있으므로, 아래 그림 7과 같이 입력 음악 정보 (R) 값이 이동

$$R=[3\ 4\ 5\ 6] \rightarrow R= [3\ 4\ 5\ 6] \rightarrow R= [3\ 4\ 5\ 6]$$

$$Q=[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8] \rightarrow Q=[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8] \rightarrow Q=[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$$

그림 7. Sliding DP 매칭

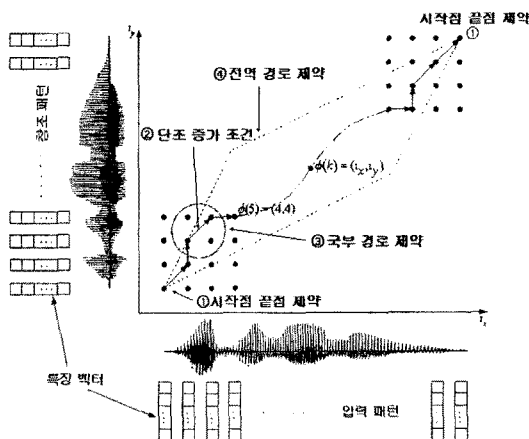


그림 8. 동적 정합 매칭 방법

(sliding)하며 비교를 하게 된다.

이를 수식으로 표현한 것은 아래와 같다[27].

$$d_{ps}(r_i, q_j) = \frac{\sqrt{\sum_{m=0}^{M-1} [r_i(m) - q_j(m-ps)]^2}}{\sqrt{\sum_{m=0}^{M-1} r_i^2(m) \sum_{m=0}^{M-1} q_j^2(m-ps)}}, \quad 0 \leq m, m-ps \leq M-1 \quad (2)$$

식 2에서 $r_i(m)$ 은 저장된 음원 정보이며, $q_j(m-ps)$ 는 입력 음악 정보이다. 이때 $q_j(m)$ 을 ps 만큼 sliding해 가면서 식 (2)에 의해 두 정보 ($r_i(m)$ 와 $q_j(m)$)간의 상이도를 측정하게 된다.

4. 실험 결과 및 토의

4.1 허밍 검색 실험

허밍 실험을 위해 하기 그림 9와 같이 22개의 wave 파일에서 3.2.(1)절에서 설명한 방법으로 기본 주파수 정보를 추출한 뒤, 텍스트 파일에 순서대로 저장하여 데이터베이스를 구성하였다. 그리고 검색할 내용 샘플을 생성하기 위해 22곡 중에서 고른 3곡의 일부를 허밍으로 녹음하였다.

일반적으로 음원의 멀티 트랙 혼합 신호에서 메인(보컬) 멜로디의 기본 주파수를 찾아내는 것은 매우 어려운 작업이다. 그래서 본 연구에서는 기존 일반적인 가요 미디 파일로부터 한 악기연주음만 추출하여 이를 wave 파일로 저장하고 (그림 9의 wave 파일), 이 파일과 입력된 허밍 음파의 검색을 수행하였다. 이는 미디 파일 같은 경우 내부에 각 악기별 음정보가 저장되어 있어서, 원하는 악기음만 추출 가능하기 때문이다.

실제적인 사용 환경을 고려하더라도, 기존 대부분

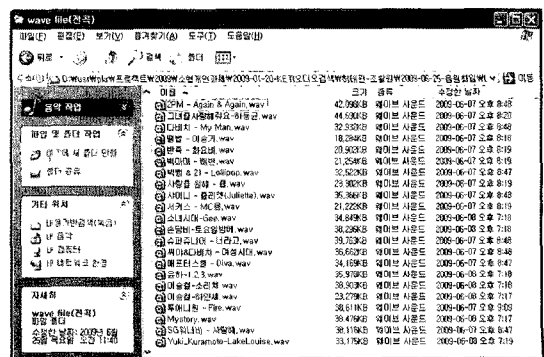


그림 9. 실험용 22개의 wave 파일

의 가요등과 같은 음원은 해당 미디어파일이 존재하므로 검색하고자하는 음원의 경우 사전에 미디어 파일 형태에서 검색에 용이한 악기음 정보만을 검출하여 미리 데이터베이스에 저장하여 사용할 수 있기 때문이다. 본 연구에서 가요 미디어 파일로부터 한 악기연주음을 추출한 기준으로는 일반적인 사람의 허밍음에서 크게 차이를 보이지 않는 악기연주음을 임의로 선택하였다. 그러므로 본 연구에서는 멀티 트랙의 혼합 신호에서 메인(보컬) 멜로디의 기본 주파수를 추출한 것이 아니라, 단음에서 기본 주파수를 추출하여 특징으로 사용하였다.

Wave 파일은 44 khz의 sampling rate 과 16 비트의 양자화 비트수를 이용하여 생성되었다.

그런데 허밍을 녹음하여 기본 주파수 정보를 추출해 본 결과 두 가지 문제점이 발생하였다. 첫 번째는 사람의 목소리로 녹음한 허밍과 음원의 주파수 대역 차이였고, 두 번째는 허밍을 녹음할 당시 발생한 잡음들이 기본 주파수 정보에 상당한 영향을 준다는 점이다.

그래서 음원과 녹음된 허밍의 주파수 대역 차를 좁히기 위해서 추출된 음원과 허밍의 주파수 정보값의 차를 구하고 그것들의 평균값을 구하였다. 그리고 그림 10과 같이 허밍의 주파수 정보를 그 평균값만큼 음원의 정보와 가까운 방향으로 보정(shift)해주었다.

그런데 허밍의 기본 주파수 정보를 보정시키는 과정에서, 허밍 녹음 시 발생한 잡음들도 같이 보정되어 결국, 잡음들은 그대로 남아 있게 되었다. 잡음들의 영향을 줄이기 위해서 5 point averager filter를 통과시킨 후 정합 매칭을 수행하였다. 하기 그림 11은 하기 4.4절 Roger Jang의 데이터베이스화일에서 동적정합매칭을 수행한 두 개의 샘플 비교 그림을 나타낸다.

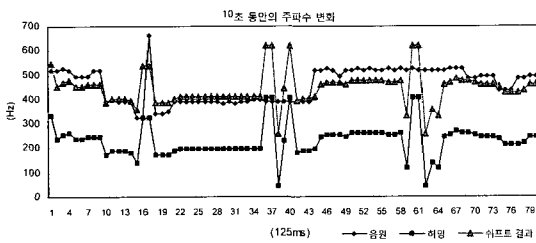


그림 10. 음원과 입력 허밍의 기본 주파수 차이 예시

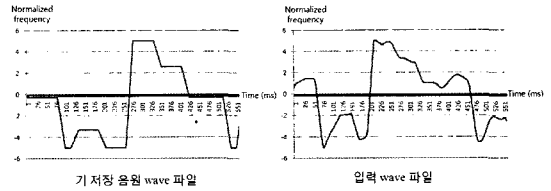


그림 11. 동적 정합 매칭을 한 두 개의 샘플 비교 그림 예

4.2 녹음 검색 실험

길거리나 카페 또는 여러 미디어들을 통해 음악을 듣고, 어떤 음악인지 찾고 싶지만, 음악의 정보를 알 수 없는 경우에 그 자리에서 바로 그 음악을 녹음하여, 녹음된 내용을 기반으로 음원을 검색하는 방식이다. 본 실험에서는 10초에서 40초 사이로 단일트랙으로 구성된 음원의 일부를 저장하여, 검색을 위한 입력 샘플로 이용하였다. 단일트랙으로 구성된 22곡의 wave파일에서 기본 주파수 정보를 추출하여 데이터베이스를 구성하였고, 5개의 입력 샘플을 저장한 뒤 기본 주파수 정보를 추출하였다. 그리고 입력 샘플과 22곡의 wave 파일에서 기본 주파수 특성을 추출하기 위해 3.2.(1)절에서 설명한 방법을 사용하였다. 그리고 입력 샘플과 데이터베이스에 저장된 곡간의 유사도 측정을 위해 3.3절에서 소개된 동적 정합 매칭 방법을 사용하였다.

실험결과 약 30초의 입력 샘플을 이용하여, 22곡으로 구성된 데이터베이스에서 검색하는데 약 1분의 처리시간(CPU : Intel(R) Core(TM)2 Duo T7250 @ 2.00GHz, RAM : 2 GByte, OS : Window XP 기준)이 소요되었다. 매칭 결과 음원들을 오류 없이 모두 정확하게 검색함을 알 수 있었다. 하기 그림 12는 이러한 실험 결과의 예를 나타낸 것이다.

4.3 악기 연주 검색 실험

이 방식은 음은 기억하고 있으나 노래 정보를 모르거나 기억 안날 때, 전자 악기로 음악의 일부를 연주하여 해당 음악을 검색하는 방식이다. 그림 13은 찾고자 하는 음악을 연주하여 녹음하면 미디어 파일을 생성시켜 주는 프로그램으로, 이 방식을 실용화한다면 음악 검색 서비스를 제공 할 사이트나 프로그램에서 악기 연주 프로그램을 만들고 미디어 파일을 생성시킬 필요 없이 연주 한 음 정보 데이터로 바로 데이터베이스의 음원과 검색을 수행할 수 있게 된다.

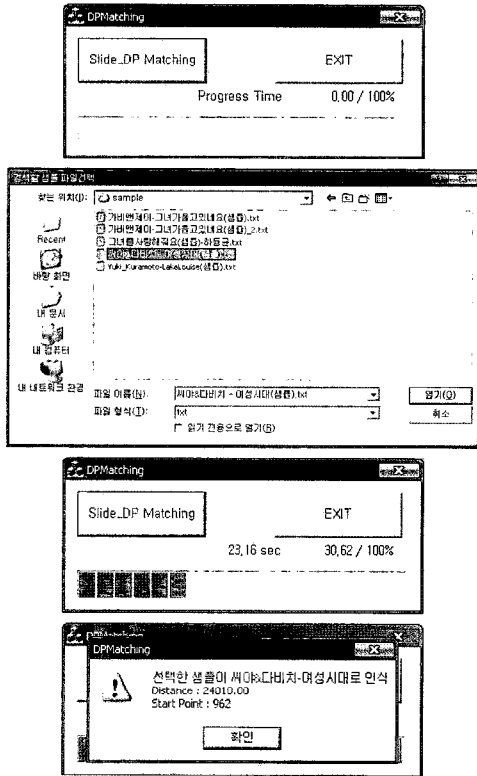


그림 12. 녹음 검색 실험 결과의 예

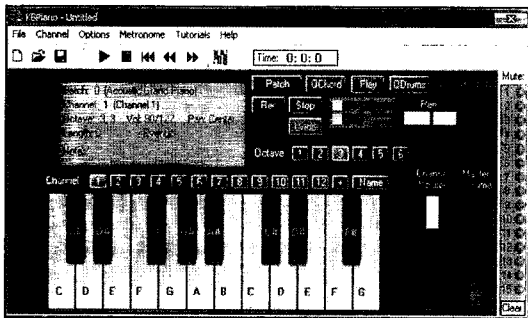


그림 13. KBPiano 프로그램

하지만 악기 연주를 할 때 실제 노래와 음높이를 똑같이 연주를 한다는 것은 매우 어렵다. X/Y 옥타브만큼 높을 수도 낮을 수도 있다. 하지만 본 실험에서 사용한 동적 정합 매칭알고리즘으로는 이러한 차이를 해결하지 못한다. 그러므로 본 연구에서는 앞뒤 음의 차, 첫 음과의 차를 특징으로 이용하였다.

그림 14에서는 원음보다 노트 10만컴씩 낮은 음높이로 연주를 하였다고 가정하였다. 그리고 색칠 되어 있는 2번 째 음을 실수나 부정확한 기억 등으로 노트

원음	62	60	70	62	67	67
앞뒤 음의 차	-2	+10	-8	+5	0	
첫 음과의 차	-2	+8	0	+5	+5	

오지발생	52	49	60	52	57	57
앞뒤 음의 차	-3	+11	-8	+5	0	
첫 음과의 차	-3	+8	0	+5	+5	

그림 14. 노트간의 차이를 특징으로 사용한 예

1만큼 낮게 연주 하였다고 가정하였다. 기존 음높이 변화를 이용한 방식들은 앞뒤 음의 차를 이용하였는데 지금과 같은 오차가 생겼을 경우 앞의 음과의 차, 뒤의 음과의 차 2군데서 오류가 발생한다. 하지만 첫 음과의 차를 이용한다면 잘못 연주한 부분 1곳에서만 오류가 발생하여 정확성을 높여준다. 그림 15는 실험 결과의 예를 나타낸다. 음원 미디어에서 추출한 음 정보를 텍스트 파일로 만들어 데이터베이스를 구축하고 악기로 연주한 샘플 파일의 음 정보 역시 텍스트 파일로 만들어 검색을 한 결과이다. 실험 결과 음원 미디어 파일을 오류 없이 정확하게 검출함을 알 수 있었다.

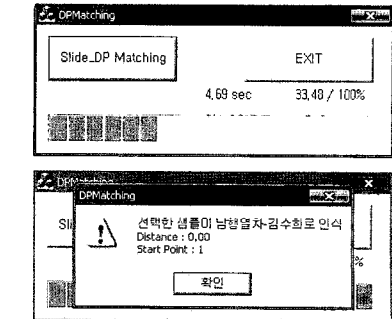
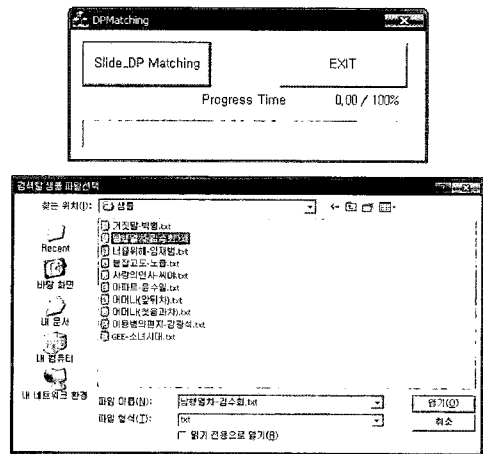


그림 15. 악기 연주 검색 실험 결과의 예

4.4 MIREX-Roger Jang 데이터베이스를 이용한 검색

마지막으로 Music Information Retrieval Evaluation eXchange (MIREX) 2006에서 사용하였던 Roger Jang의 데이터베이스를 이용한 검색실험을 수행하였다. 본 데이터베이스는 48개의 MIDI화일에 대해 118명의 사람들이 전화기, 마이크로폰 등을 이용하여 다양한 환경에서 노래 및 허밍을 수행한 2797개의 wave파일들로 구성되어 있다[28]. 일반적으로 MIREX등에서는 1후보의 정확도뿐 아니라 10후보 및 20후보 정확도를 시스템 성능으로 함께 고려하므로[29], 본 연구에서도 Roger Jang 데이터베이스에서 제공하는 PV화일을 이용하여 1후보, 10후보 및 20후보 정확도를 함께 측정하였다. 여기서 10후보란 검출된 상위 10개의 후보군내에 맞는 음원이 존재하는 정확도를 나타낸 것이다. 실험결과 1후보 검출 정확도는 32.2% (901/2797), 10후보 검출 정확도는 68.3%(1910/2797) 그리고 20후보 검출 정확도는 84% (2350/2797)의 결과를 나타냈다.

4.5 문제점 및 보완점

허밍 검색과 녹음 검색 실험 결과, 압력 잡음에 의해 검색 성능이 영향을 많이 받는 것을 알 수 있었다. 그래서 허밍을 녹음하는 다른 방법에 대해 생각해 보았다. 사람의 귀가 소리를 감지하는 방법은 세 가지 기준으로 분류할 수 있다. 소리의 크기, 소리의 높고 낮음(주파수), 그리고 음색이다. 음색이란 사람은 피아노와 바이올린이 같은 크기로 261.63Hz의 낮은 도를 연주한다면 그 두 소리가 다르다고 느낄 수 있는데, 이것은 바로 음색의 차이를 느낄 수 있기 때문이다. 그러므로 사람의 목소리 음색을 구별하여 녹음하는 경우, 원하는 소리 이외의 잡음에 대해 강한 허밍 검색이 가능할 것으로 예상된다.

음높이 변화를 이용하여 동적 정합 매칭을 하였을 경우 +2의 경우 50에서 52로 변한 것인지 60에서 62로 변한 것인지 알 수가 없고 대부분의 값들이 -3 ~ +3의 값을 갖기 때문에 오차가 생겼을 경우 원래의 노트로 검색할 경우보다 정확성이 떨어지는 것을 확인하였다. 그래서 고려한 방법이 그림 16과 같이 원음과 연주한 음 비교 시 첫 음의 차이만큼 뒤의 모든 음도 더하거나 빼 값으로 비교를 하는 것이다.

만약 첫 음에서 오차가 생겼을 경우에는 다시 연

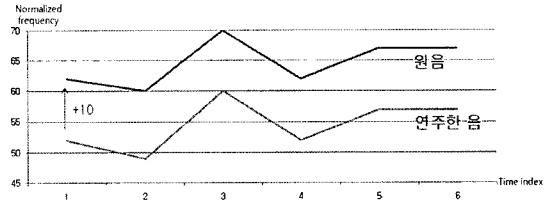


그림 16. 원음과 연주한 음의 노트 값 차이 보정

주하여 검색 할 수도 있고, 뒤의 음들이 같은 값의 차이가 계속 난다면 그 차이만큼 다시 이동시켜 비교하는 방식을 사용할 수도 있을 것이다. 이렇게 하면 음높이 변화를 이용한 방식은 보다 좀 더 정확한 결과를 얻을 수 있게 될 것으로 예상된다.

Roger Jang의 데이터베이스에 대한 실험 결과 1후보 검출 성능은 다소 떨어짐을 알 수 있었다. 이는 사용한 동적정합매칭 방법에서 사용하는 지역적(local) 매칭 정확도의 문제인 것으로 파악된다. 향후 earth mover's distance (EMD)와 같은 전역적(global) 매칭 방법들을 결합하여 사용함으로써 매칭 정확도를 향상할 수 있을 것으로 예상된다.

5. 결 론

본 논문에서는 현재의 음악 검색 방식에서 벗어나 새로운 내용 기반 검색 방식을 제안하였다. 기존 검색 방식 외에 내용 기반의 검색 방식도 사용 한다면 좀 더 많은 사람들이 자신이 원하는 음악을 검색할 수 있을 것으로 생각한다. 본 논문에서 제안했던 3가지 내용 기반 검색 방식 중 허밍과 노래의 녹음 방식에서는 좀 더 깨끗하고 우리가 원하는 소리만 녹음하고 잡음은 모두 제거해 줄 수 있는 기술이 필요하며, 미디를 사용한 연주 검색 방식은 악기 연주를 할 수 있는 사람만 이용할 수 있는 단점이 있지만 주파수 추출 방식보다 적은 저장 공간 및 빠른 속도로 이용 가능할 것으로 예상된다.

향후 음원 검색 시간을 보다 단축하기 위한 추가 연구가 필요하며, 대용량 음원 파일들을 대상으로 한 실험을 통해, 보다 다양한 환경에서의 성능 평가가 필요할 것으로 예상된다.

참 고 문 헌

[1] S. Rho, and E. Hwang, "Implementation of an

- Efficient Music Retrieval System based on the Analysis of User Query Pattern,” *Journal of KIPS*, 10-A, Vol. 6, pp. 735-748, December 2003.
- [2] A. Ghias, *et al.*, “Query by Humming Musical Information Retrieval in An Audio Database,” *Proceedings of ACM Multimedia*, pp. 231-236, 1995.
- [3] R.J. McNab, *et al.*, “Toward the Digital Music Library: Tune Retrieval from Acoustic Input,” *Proceedings of the First ACM Conference on Digital Libraries*, pp. 11-18, 1996.
- [4] R.J. McNab, *et al.*, “The New Zealand Digital Library Melody Index,” *Digital Libraries Magazine*, 1997.
- [5] R. Typke, and L. Prechelt, “An Interface for Melody Input,” *ACM Transactions on Computer-Human Interaction*, pp. 133-149, 2001.
- [6] A. Kornstadt, “Themefinder: A Web-Based Melodic Search Tool,” *Computing in Musicology*, Vol. 11, MIT Press, 1998.
- [7] Themefinder. <http://www.themefinder.org/> (accessed on September 14, 2009).
- [8] S. Blackburn, and D. DeRoure, “Tool for Content Based Navigation of Music,” *Proceedings of ACM Multimedia*, pp. 361-368. 1998.
- [9] MiDiLiB project. <http://www-mmdb.iai.uni-bonn.de/forschungprojekte/midilib/english/> (accessed on September 14, 2009)
- [10] S.R. Subramanya, *et al.*, “Transforms-based Indexing of Audio Data for Multimedia Databases,” *Proceedings of IEEE International Conference on Multimedia Systems*, 1997.
- [11] A. Uitdenbogerd, and J. Zobel, “Manipulation of Music for Melody Matching,” *Proceedings of ACM Multimedia Conference*, pp. 235-240, 1998.
- [12] A. Uitdenbogerd, and J. Zobel, “Melodic Matching Techniques for Large Music Databases,” *Proceedings of ACM Multimedia Conference*, pp. 57-66, 1999.
- [13] E. Hwang, S. Rho, “Fast Melody Finding Based on Memorable Tunes,” *Proceedings of the 1st International Symposium on Computer Music Modeling and Retrieval*, Montpellier, France, pp. 227-239, May 26-27, 2003.
- [14] “노래/허밍에 의한 질의 방법 및 장치,” 국내 특허 출원 (출원번호 : 2008-0084038, 출원일 : 2008.08.27, 발명자 : 엄기완, 덩정, 주쭈안, 이재원, 시원원, 출원인 : 삼성전자 주식회사).
- [15] “멜로디 질의 기반 음악 검색 방법,” 국내 특허 등록 (등록번호 : 2004-0101592, 등록일 : 2004.12.03, 발명자 : 배소영, 송정민, 윤경로, 출원인 : 엘지전자 주식회사).
- [16] “허밍 기반의 음원 질의 / 검색 시스템 및 방법,” 국내 특허 등록 (등록번호 : 2006-0019096, 등록일 : 2006.03.03, 발명자 : 허성필, 한평희, 원성기, 출원인 : 주식회사 케이티).
- [17] “허밍과 음성인식을 이용한 음악정보검색 방법,” 국내 특허 등록 (등록번호 : 2005-0053903, 등록일 : 2005.06.10, 발명자 : 정현열, 허성필, 석수영, 출원인 : 영남학원).
- [18] “Music Information Retrieval Using a 3D Search Algorithm,” (Patent Number : 7488886, Granted Date : 2009.02.10, Inventors : Kemp, Thomas Esslingen DE, Applicants : Sony Deutschland GmbH Berlin DE).
- [19] “Musical Theme Searching,” (Patent Number : 7518052, Granted Date : 20090414, Inventors : Kourbatov, Alexei Bellevue WA US, Applicants : Microsoft Corporation Redmond WA US).
- [20] “Music Similarity Function Based on Signal Analysis,” (Patent Number : 7031980, Granted Date : 20060418, Inventors : Logan, Beth T. Somerville MA US, Salomon, Ariel Cambridge MA US, Applicants : Hewlett-Packard Development Company, L.P. Houston TX US).
- [21] “Melody Retrieval System,” (Patent Number

: 20080148924, Granted Date : 2008.06.26,
Inventors : Tsui, Chi-Ying Hon kong CN Shi,
Bertaram Hong Kong CN Yung, Chi Wai
Hong Kong CN, Applicants : Perception
Digital Technology(BVI) Limited Road Town
VG).

- [22] "Melody Retrieval System and Method,"
(Patent Number : 0944033, Inventors : Sonoda,
Tomonari).
- [23] 음계주파수 테이블, <http://cid-818bd0d828b3f50e.spaces.live.com/Blog/cns%21818BD0D828B3F50E%21393.entry> (accessed on September 14, 2009).
- [24] MIDI File Format, <http://faydoc.tripod.com/formats/mid.htm> (accessed on September 14, 2009).
- [25] MIDI File Format, <http://blog.daum.net/jty71/15645096> (accessed on September 14, 2009).
- [26] 한학용 저, 패턴인식 개론 - Matlab 실습을 통한 입체적 학습, 한빛미디어, 2005.
- [27] Jungmin Song, So-Young Bae, and Kyoungro Yoon, "Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System," *Proceedings of Symposium on Music Information Retrieval*, 2002.
- [28] QBSH: A Corpus for Designing QBSH (Query by Singing/Humming) Systems, <http://neural.cs.nthu.edu.tw/jang2/dataSet/childSong4public/QBSH-corporus/index.htm> (accessed on Sep. 14, 2009).
- [29] MIREX, http://www.music-ir.org/mirex/2009/index.php/Main_Page (accessed on Sep. 14, 2009).

허 태 관

2002년 3월~현재 동국대학교 전자공학과 공학사
관심분야 : 음성신호처리

조 황 원

2002년 3월~현재 동국대학교 전자공학과 공학사
관심분야 : 음성신호처리

남 기 표

2006년 3월~2009년 2월 상명대학교 디지털미디어학부
이학사
2009년 3월~현재 동국대학교 전자공학과 석박사통합
과정
관심분야 : 음성신호처리, 영상신호처리, 바이오인식

이 재 현

2004년 3월~현재 동국대학교 전자공학과 공학사
관심분야 : 영상 및 음성신호처리

이 석 필

1992년 8월 연세대학교 전기공학과 석사
1997년 7월 연세대학교 전기공학과 박사
2002년 2월~현재 전자부품연구원 방송통신융합연구센
터장
관심분야 : Place Shift system, 초고화질 AV시스템, 휴
대영상시스템

박 성 주

1997년 2월 경북대학교 전자공학과 석사
2009년 현재 광운대 박사 재학중.
2004년 4월~현재 전자부품연구원 선임연구원.
관심분야 : 방통융합 솔루션, 신호처리



박 강 령

1994년 연세대학교 전자공학과 졸업
1996년 연세대학교 전자공학과 석사
2000년 연세대학교 전기·컴퓨터공학과 박사
2000년~2003년 LG 전자 기술원



Digital Vision Group 선임연구원
2003년~2008년 상명대학교 소프트웨어대학 디지털미
디어학부 조교수
2008년~현재 동국대학교 전자공학과 부교수
관심분야 : 영상/음성신호처리, 바이오인식, 패턴인식