

내용기반 음악검색 시스템의 비교 분석

A Comparative Analysis of Content-based Music Retrieval Systems

노정순 (Jung-Soon Ro)*

초 록

본 연구는 웹에서 접근 가능한 내용기반 음악검색(CBMR) 시스템들을 조사하여, 탐색질의의 종류, 접근점, 입출력, 탐색기능, 데이터베이스 성격과 크기 등의 관점에서 특성을 비교 분석하고자 하였다. 비교 분석에 사용된 특성을 추출하기 위해 내용기반 음악정보의 특성과 시스템 구축에 필요한 파일의 변환, 멜로디 추출 및 분할, 색인자질 추출과 색인, 매칭에 사용되는 기술들을 선행연구로 리뷰하였다. 15개의 시스템을 분석한 결과 다음과 같은 특성과 문제점이 분석되었다. 첫째, 도치색인, N-gram 색인, 불리언 탐색, 용어절단검색, 키워드 및 어구 탐색, 음길이 정규화, 필터링, 브라우징, 편집거리, 정렬과 같은 텍스트 정보 검색 기법이 CBMR에서도 검색성능을 향상시키는 도구로 사용되고 있었다. 둘째, 시스템들은 웹에서 크롤링하거나 탐색질의를 DB에 추가하는 등으로 DB의 성장과 실용성을 위한 노력을 하고 있었다. 셋째, 개선되어야 할 문제점으로 선율이나 주선율을 추출하는데 부정확성, 색인자질을 추출할 때 사용되는 불용음(stop notes)을 탐색질의에서도 자동 제거할 필요성, 옥타브를 무시한 solfege 검색의 문제점 등이 분석되었다.

ABSTRACT

This study compared and analyzed 15 CBMR (Content-based Music Retrieval) systems accessible on the web in terms of DB size and type, query type, access point, input and output type, and search functions, with reviewing features of music information and techniques used for transforming or transcribing of music sources, extracting and segmenting melodies, extracting and indexing features of music, and matching algorithms for CBMR systems. Application of text information retrieval techniques such as inverted indexing, N-gram indexing, Boolean search, truncation, keyword and phrase search, normalization, filtering, browsing, exact matching, similarity measure using edit distance, sorting, etc. to enhancing the CBMR; effort for increasing DB size and usability; and problems in extracting melodies, deleting stop notes in queries, and using solfege as pitch information were found as the results of analysis.

키워드: 내용기반 음악 검색시스템, 선율검색, 노래검색, 악보검색, 내용기반 음악정보

CBMR(content-based music retrieval), QBH(query by humming),

QBN(query by music notation), QBC(query by contour)

* 한남대학교 문과대학 문현정보학과 교수(jsr@hnu.kr)

■ 논문접수일자 : 2013년 8월 1일 ■ 최초심사일자 : 2013년 9월 5일 ■ 게재확정일자 : 2013년 9월 10일

■ 정보관리학회지, 30(3), 23-48, 2013. [<http://dx.doi.org/10.3743/KOSIM.2013.30.3.023>]

1. 서 론

내용기반 음악 검색(CBMR: Content-Based Music Retrieval)이란 악보나 오디오, 심볼문자 음악파일을 대상으로 마이크로폰을 사용하여 직접 노래를 부르거나 오디오 파일을 재생하여, 혹은 가상 건반이나 오선지에 선율을 음표로 나타내거나 텍스트 박스에 음계나 음정을 입력하여 음악을 검색하는 것으로, 노래가사를 텍스트로 탐색하는 가사(lyric) 검색이나, 곡명과 같은 메타데이터로 검색하는 기존의 음악 검색과 구별된다. CBMR은 일반 이용자가 선율로 곡명이나 악보, 오디오 파일을 찾는 데는 물론, 분위기가 비슷한 음악을 추천한다든지, 음원의 저작권 관리 영역에서 표절여부를 판정하는데 유용하게 사용될 수 있다(노정순, 2011).

CBMR은 탐색질의를 입력하는 방식에 따라 QBE(query by example), QBH(query by humming/sing), QBP(query by playing), QBN(query by music notation), QBC(query by contour) 검색 등으로 나뉜다.

Ghias, Logan, Chamberlin, Smith(1995)에 의해 허밍검색(QBH)이 소개된 이후 많은 연구와 시스템이 소개되었다. 2004년 이후 TREC(Text REtrieval Conference)처럼 표준 테스트 콜렉션을 사용한 연간 음악정보 검색 평가회 MIREX(Music Information Retrieval Evaluation eXchange)를 통해 오디오 음악의 자동 전사(transcription), 오디오 박자 추적(beat tracking), 다성 음악에서 멜로디 추출과 음표 인식, 코드나 키의 검출, 템포 추출, 유사도 매칭기법, 허밍질의 검색 등의 분야에서 활발한 연구가 이루어지고 있다(<http://www.music-ir.org/mirex/>

wiki/MIREX_HOME). 다양한 연구기반 실험 시스템을 비롯하여 선율로 검색된 음악을 구입 할 수 있도록 온라인 음반가계로 링크시키는 상업적인 시스템도 등장하였다. 그러나 CBMR 시스템들을 조사하여 하나의 분석틀로 비교 분석하여 소개한 연구는 부족한 현실이다. Typke, Wiering, Veltkamp(2005)는 실험시스템 17개를 비교하였으나 그 중 3개만 현재 웹으로 접근 가능하다.

본 연구는 2012년 말 웹에서 접근 가능한 CBMR 시스템들을 조사하여, 탐색의 접근점, 입력력, 탐색기능, 데이터베이스 성격과 크기 등 의 관점에서 특성을 비교 분석하고자 하였다. 선행연구를 통하여 CBMR 시스템 구축에 필요한 파일의 변환, 멜로디 추출 및 분할, 색인자질 추출과 색인, 매칭기법과 같은 구성요소에 사용된 제반 기술들을 조사하여 분석의 틀로 사용하였다.

CBMR에 사용된 기술과, 웹기반 시스템의 특성과 문제점, 활성화를 위해 향상되어야 할 발전 과제를 제시한 연구결과는 CBMR 시스템의 성능향상은 물론 전자도서관에서 다양한 음악자료의 내용기반 검색시스템을 개발 및 선정하는데 가이드라인으로도 활용될 수 있을 것이다.

2. 내용기반 음악정보의 속성 및 정의

Downie(1999)는 서지정보를 제외하고 음악 검색시스템에서 사용할 수 있는 내용(content) 정보로 6가지 속성 즉 피치, 시간, 화성, 음색, 편집, 텍스트 정보를 들고 있다.

피치(pitch) 정보는 피치, 음정(interval), 선율의 윤곽(melody contour)과 같은 특성으로 표현된다. 피치는 음높이 즉 사람이 느끼는 음의 고유 주파수로, Hz와 같은 초당 진동수나, A4, B3와 같은 기보법을 사용한다. A4는 보통 440 Hz를 나타내고 주파수가 2배가 되면 한 옥타브 높은 음(A5)이 된다. 두 개의 피치 사이의 차이는 음정이라 하며, 반음을 1도로 하여 8도 까지 증(+) 감(-)을 표현한다. 선율(melody)이란 여러 개의 음을 규칙에 따라 시간적으로 배치한 것으로, 듣는 사람에게 같은 느낌을 주는 C장조의 선율 EDCEDC와 G장조의 BAGBAG는 선율의 윤곽이 동일하다. 윤곽은 주로 파슨스(Parsons) 코드로 표현된다(3.2 참조).

시간(temporal) 정보는 템포(tempo), 박자(meter), 피치 길이(duration), 리듬을 포함한다. 템포는 분당 비트수로 표시하거나($\text{J} = 120$ 은 1분에 120박자 빠르기), 아다지오나 프레스토 등과 같은 지시어로 표현한다. 박자(meter)는 몇 분 음표 몇 개로 한 마디가 구성되는지를 나타내는 것으로, 3/4박자는 4분음표 3개가 한 마디를 이루는 것을 의미한다. 피치 길이는 4분음표(J), 8분음표(J) 등 음의 길이이다. 쉼(표)를 포함하여 음의 길이나 강약, 빠르기 등에 따라 되풀이되는 음의 흐름을 리듬이라 한다.

화성(harmonic)이란 둘 이상의 피치를 동시에 울려 합성된 음으로 다성(polyphony)이라고도 하며, 화성이 발생하지 않은 단성(monophony)과 구별된다. 화성은 코드(chord)로 표현되며 3화음(triads), 4화음 등이 있다.

음색(timbral)은 사람이 음을 들었을 때 느끼는 음의 이미지를 말하며, 그 음이 어떤 악기의 음이라고 판단하는 것은 악기에 따라 피치의

음색이 다르기 때문이다. 과학적으로는 기본파의 몇 배나 되는 고주파가 어떤 비율(spector)로 존재하는가에 따라 순간의 음색이 결정되지만, 음이 나기 시작할 때와 뒤의 여운에 따라 음색은 변화한다. 이 시간의 변화는 악기에 따라 고유의 패턴이 있기 때문에 스펙터의 시간변화로 음색을 알 수 있다. 또한 운궁법(bowings), 운지법(fingerings), 피아노 페달링(pedalings)과 같은 편집 정보로도 음색을 알 수 있다.

편집(editorial) 정보는 운궁법, 운지법, 이음줄, 음의 세기(p, ff, crescendo 등) 등 연주 지시사항을 나타내는 정보이고, 텍스트(textual) 정보는 노래의 가사(lyric)이다.

6가지 내용기반 음악정보 중 CBMR에 주로 사용되는 속성은 피치나 시간, 화성 정보이다. 음색 정보는 주로 장르 분류시스템에 사용되며, 음색을 접근점으로 사용한 예로는 비슷한 소리(sound)를 검색하는 Soundfisher(<http://www.soundfisher.com>)가 있다.

3. CBMR 시스템 구성 요소 및 기술

CBMR 시스템은 소스 DB와 함께 파일의 변환, 멜로디 추출 및 분할, 색인자질 추출과 색인, 매칭기법 등과 같은 처리 기술로 구축된다. <표 1>은 Ghias 등(1995)의 QBH 이후 소개된 대표적인 CBMR 시스템을 소스 파일의 변환과, 멜로디 추출 및 분할, 색인자질 추출, 질의처리, 매칭방법 등 시스템을 구축하는데 필수적인 구성요소로 나누어 요약한 것이다(<표 1>의 색인자질은 약어 c(ontour), p(itch), i(nterval),

〈표 1〉 대표적인 CBMR 시스템의 특성 요약

시스템명(개발자)	문헌	DB	색인자질	질의	질의 분석 및 멜로디(파치) 추출	매칭
QBH (Cornell U)	Ghias et al., (1995)	183 Midi	c(uds) n-gram 색인	해킹	Autocorrelation Maximum Likelihood, Cepstrum 분석	부분매칭(LCS)
Meldex (U. Waikato)	McNab et al., (1996; 1997) Bainbridge (2004)	포크 툰 9400	i, c, d	해킹	최대 20ms 단위로 분할 pitch tracker 사용하여 p와 d 추출	완전일치, 부분일치
Musipedia (Tuneserver)	Precelt & Typke (2001) Typke, Veltkamp, & Wiering (2004)	Midi, Musicxml 10370 주제	c p, d, r	해킹	보컬추출(skyline 알고리즘) Wav → FFT 사용하여 udr 추출	편집거리(c) EMD (p+d, r)
MIT	Chai & Vercoe (2002)	8000 Midi	c+d+비트	해킹	Autocorrelation	편집거리(p < p+비트)
U. Mich	Shiffin, Pardo, & Birmingham (2002)	277 Midi에서 2633 테마 추출	p+d	해킹	절의 10ms 단위로 분할 Tolonen의 Pitch tracker 사용	HMM forward, 편집거리
NYU	Zhu & Shasha (2003)	50 비틀즈 노래에서 1000개 신율(15-30음)	C, R* tree 색인	해킹	10ms 프레임 분할, Tolonen	시계열 검색 (UTW, LDTW, DTW)
MUSEMBLE (이주태)	노승민 등 (2003) Rho et al., (2008)	Midi (12000)	jMusic으로 특성추출 c, ior파울(LSR)	해킹 오선지	히밍을 Wav로 저장 후 WAE, ADF, K-최근 클러스터링	완전매칭 Boyer-Moore 유사매칭 (동적프로그래밍, LCS)
Shazam	Wang (2003; 2006)	3백만 트랙/다양	평가프린팅	QBE 10초 이상 해扰타이틀로 쳐장: 15초	경과시간 - 주파수에서 peak-intensity의 주파수를 해扰타이틀로 쳐장: 10ms 단위로 분할	완전일치
VocalSearch	Pardo, Shiffin, & Birmingham (2004)	260 Midi	i, r(ioi 비율:LSE)	해킹	히밍을 Wave로 저장 후 10ms 단위로 분할 Autocorrelation	편집거리, HMM
Tunepal	Duggan et al. (2009; 2010) Hug, Cartwright, & Pardo (2010)	악보-> Midi 변환 이용자 질의 (11000 example)	p i, ioi 비율	QBP	연주된 질의 abc 코드로 전사, 정규화	편집거리
ThemeFinder (Stanford & OSU)	Kornstädter (1998)	Midi Hundrund	c, i, p	텍스트		완전일치
C-Brahms (U. Helsinki)	Lemströms & Tariño (2003)	7667 Midi/다양	p+d, r, i	간반(88) 2-30음	9개 탐색방법(단성 다성, 완전일치(리듬, 다형) Splitting 정규화, 부분일치)	
SMILE (U. Padova)	Melucci & Orio (2004)	419 Midi	p, d	간반	색인: LBDM으로 멜로디를 분할 후 파치와 음길이 정규화	tf x idf와 코사인 유사도
Peachnote	Viro (2011)	65,000 악보 1백만 page(pdf)	p(Smartscore OMR 사용)	간반:	색인: 15-gram(p)	15 gram 일치탐색

d(uration), r(hytm) 사용). <표 1>의 시스템을 포함하여 선행연구에서 연구된 CBMR 시스템 구축에 필요한 제반 기술들을 구성요소별로 나누어 요약하면 다음과 같다.

3.1 소스파일의 변환

음악작품은 주로 악보(score)나, Midi와 ABC 파일과 같은 심볼문자(symbolic), 오디오(WAV, PCM, MP3 등) 파일로 표현된다.

악보는 대체로 이미지 파일로 스캔된 후, 광학악보인식(Optical Music Recognition) 장치

를 사용하여 MusicXML 문서나 Midi, ABC 파일로 변환하여 심볼문자로 전사된다(<표 1>의 Peachnote, Musipedia, Tunepal).

MusicXML은 퍼치, 박자, 조성, 리듬, 화음 등의 모든 음악정보를 표현한 마크업 언어로, 단성은 물론 다성까지 Midi 파일로 변환하는데 필요한 요소를 포함하고 있어 Midi나 ABC 파일로 변환이 가능하다. <그림 1>은 Gabriel Fauré의 “Après un rêve”의 첫 4마디 악보를 MusicXML로 코딩한 예이다. 단성의 성악파트와, 피아노 반주 파트, Andantino, pp, dolce 와 같은 템포, 성악의 가사까지 코딩된다.

```

<part id="P1">
    <measure number="1">
        <attributes>
            <divisions>24</divisions>
            <key>
                <fifths>-3</fifths>
                <mode>minor</mode>
            </key>
            <time>
                <beats>3</beats>
                <beat-type>4</beat-type>
            </time>
            <clef>
                <sign>G</sign>
            </clef>
            <line>2</line>
        </attributes>
        <direction directive="yes" placement="above">
            <direction-type>
                <words default-y="15" font-weight="bold">Andantino</words>
            </direction-type>
            <sound tempo="60"/>
        </direction>
        <note>
            <pitch>A4</pitch>
            <duration>1/4</duration>
            <voice>1</voice>
        </note>
    </measure>
    <measure number="2">
        <note>
            <pitch>A4</pitch>
            <duration>1/4</duration>
            <voice>1</voice>
        </note>
    </measure>
    <measure number="3">
        <note>
            <pitch>A4</pitch>
            <duration>1/4</duration>
            <voice>1</voice>
        </note>
    </measure>
    <measure number="4">
        <note>
            <pitch>A4</pitch>
            <duration>1/4</duration>
            <voice>1</voice>
        </note>
    </measure>
</part>
<part id="P2">
    <measure number="1">
        <note>
            <pitch>C4</pitch>
            <duration>1/4</duration>
            <voice>2</voice>
            <type>eighth</type>
            <stem default-y="3">up</stem>
            <staff>1</staff>
            <beam number="1">begin</beam>
        </note>
    </measure>
    <measure number="2">
        <note>
            <pitch>C4</pitch>
            <duration>1/4</duration>
            <voice>2</voice>
            <type>eighth</type>
            <stem default-y="3">up</stem>
            <staff>1</staff>
            <beam number="1">begin</beam>
        </note>
    </measure>
    <measure number="3">
        <note>
            <pitch>C4</pitch>
            <duration>1/4</duration>
            <voice>2</voice>
            <type>eighth</type>
            <stem default-y="3">up</stem>
            <staff>1</staff>
            <beam number="1">begin</beam>
        </note>
    </measure>
    <measure number="4">
        <note>
            <pitch>C4</pitch>
            <duration>1/4</duration>
            <voice>2</voice>
            <type>eighth</type>
            <stem default-y="3">up</stem>
            <staff>1</staff>
            <beam number="1">begin</beam>
        </note>
    </measure>
</part>

```

-- 이하 생략 --

<그림 1> Midi파일로 변환을 위한 MusicXML 코딩 예
<http://www.makemusic.com/musicxml/tutorial/how-music-looks-vs-how-music-sounds>

Midi(Music Instrument Digital Interface)는 전자악기나 컴퓨터 간의 디지털 신호를 교환하기 위한 프로토콜 등을 기술한 기술표준이다. Midi 파일은 Midi 메시지를 컴퓨터 파일로 저장한 파일 형태로, 퍼치를 0~127 숫자로 나타내며, 음표가 연주되는 순서와 톤, 길이, 볼륨 등의 지시사항을 포함하여, 곡의 타이틀, 작곡가, 트랙명 등의 설명 정보를 가지고 있기 때문에 음악 교환 포맷으로 가장 많이 사용되고 있다. MP3나 WAV와는 달리 Midi2score와 같은 s/w를 사용하여 악보로 전사하여 출력할 수 있기 때문에 음악검색 콜렉션 구축에 많이 사용되고 있다. Skyline 알고리즘이나 jMusic Java와 같은 라이브러리를 이용하여 Midi 파일에서 심볼문자 정보를 자동으로 추출한다(〈표 1〉의 Musipedia: Rho, Han, Hwang, & Kim, 2008).

ABC 파일은 음악을 ABC 기보(notation)로 표현한 형식이다. 〈그림 2〉는 JC's ABC Tune Finder에서 멜로디 CCGGAAG로 탐색하여 검색한 Alphabet song(Twinkle twinkle little stars로도 알려진 노래)을 악보와 ABC 기호로 출력한 예이다. 제목에서부터 음표에 해당하는 노래가사까지 표현이 가능하기 때문에 오디오와 악보, 가사를 동시에 재생하는 다중형태(multi-modal) 시스템에 쉽게 사용된다(Arifi, Clausen, Kurth, & Muller, 2003; Turetsky & Ellis, 2003).

ABC2MIDI, MIDI2ABC, Score2midi, Abc2xml 등과 같은 s/w를 사용하여 악보와 Midi, ABC 파일 사이에는 상호 변환이 가능하기 때문에, 〈표 1〉에서 Shazam을 제외한 대부분의 시스템은 임출력 데이터의 형식과 상관없이 색인과 탐색 자질로 심볼문자 코드를 사용한다.

The image shows a musical score for 'Alphabet Song' (Twinkle, Twinkle Little Star). It consists of two staves. The top staff is labeled 'Traditional Kid's Song' and the bottom staff is labeled 'Alphabet Song'. Both staves have a treble clef and a common time signature. The notes are represented by letters corresponding to the ABC notation. The lyrics are written below the notes. The score includes a tempo marking of 100 BPM and various performance instructions like 'dou - ble u' and 'and'. Below the score, there is a legend and some musical symbols.

Alphabet Song
(Twinkle, Twinkle Little Star)

Traditional Kid's Song

X: 1
T:Alphabet Song
T:(Twinkle, Twinkle Little Star)
C:Traditional Kid's Song
M:4/4
L:1/4
Q:1/4=100K:D
| "D" D D A A | "G" B B "D" A2 | "G" G G "D" F F | "A" E / 2E / 2E / 2E / 2 "D" D2 |
w: A B C D E F G H I J K L M N O P
A A "G" G F | "D" F F "A" E2 | "D" A A "G" G G | "D" F F "A" E2 |
w: Q R S and T U V W~(dou - ble u) and X Y Z
"D" D D A A | "G" B B "D" A2 | "G" G G "D" F F | "A" E E "D" D2 |
w: Now I've said my A B C's. Tell me what you think of me.

〈그림 2〉 ABC 포맷 예

3.2 색인자질로서의 심볼문자

음을 심볼문자로 표현하는 데는 주로 피치나 음길이(pitch or duration)가 사용된다. <그림 3>은 선행연구와 분석대상 시스템에서 사용된 심볼문자를 요약한 것이다. 피치만 표현하는 방법과, 음길이로 표현하는 방법, 피치에 음길이까지 포함한 방법이 있다.

피치는 도레미…로 표현하는 solfege, ABC, Midi 파일에서 사용되는 절대(absolute) 피치, 절대 피치를 12개의 값으로 축소한 directed modulo-12 value 등으로 나타낸다. Hundrum **kern 음악포맷은 중간 옥타브 do는 C4, 한 옥타브 높은 do는 C5, 한 옥타브 낮은 do는 C3 와 같이 옥타브를 표현한다. ABC 코드는 중간 옥타브 do는 대문자 C로, 한 옥타브 높은 do는

소문자 c, 두 옥타브 높은 do은 c', 한 옥타브 낮은 do는 컴마를 붙여 C,로 표시한다. Sharp 과 flat은 문자열 뒤에 “#”과 “-”를 추가하여 나타낸다.

음정(interval)은 인접한 두 피치의 차이를 반 음을 1도로, 도수에 증(+) 감(-)을 표현하는 exact interval과, 선율의 키(key)와 피치와의 차이를 나타내는 key relative, 앞 음에 비해 뒤에 오는 음의 피치가 높고 낮고 동일한지만을 U(p) D(own) R(epet)로 표현하는 파슨스 (Parsons) 코드, 높고 낮음의 차이가 임계치보다 크고 작음까지 표현하는 uUdDr(Themefinder) 등으로 표현한다. Absolute pitch보다 interval은 키가 다른 음악까지 검색하는 장점을 갖고, 파슨스 코드는 부정확한 허밍질의를 표현하는데 유용하다.

	solfege	ti	ti	do	so	mi	la	so	DoDoSoSo
피 치	cdefgab	b	b	c	g	e	a	g	SID, Parfait Ole
	ABC 기보	B	B	r4	C	G	e	A	G ABC
		B2	B2		C3	G2	E3	A2	G2 Melodycatcher
	Humdrum	B4	B4		C5	G4	E5	A4	G4 Parfait Ole
	absolute	71	71	72	67	76	69	67	
	dir modulo-12	11	11	+0	-7	+4	-9	-7	
음	exact interval	59	0	+1	-5	+9	-7	-2	peachnote
정	key relative	11	11	0	7	4	9	7	
율	파슨스		R	U	D	U	D	D	myMoozie
곡			s	u	D	U	D	d	Themefinder/RC
음	absolute	4	4	4	4	2	2	8	
길	SLE		E	E	E	S	E	L	
이	음길이 차이	0	0	0	0	2	0	6	
	음길이 비율	1	1	1	1	1/2	1	4	
피 치 + 길 이	ABC 기보	B/4	B/4	r4	c/4	G/4	e/8	A/8	G/2 ABC
		B	B		c	G	e/2	A/2	G2 J 가 디폴트일 때
		b'4	b'4	r4	c"4	g'4	e"8	a"8	g"2 Musipedia
	C-Brahms	4B4	4B4		5C4	4G4	5E8	4A8	4G2

<그림 3> 피치와 음길이를 사용한 음악정보 표현

음길이는 2(♪) 4(♩) 8(▢)와 같이 절대값으로, 혹은 앞 음의 길이와 비교하여 S(horter) L(onger) E(qual) 혹은 R(peat)으로 표시하거나, 음길이의 차이 IOI(inter-onset-interval) 또는 IOI 비율 등으로 나타낸다.

피치와 음길이를 함께 표현하는 ABC 기호는 표준 4분음표 솔(so)은 G/4로 표시하나 <그림 3>에서처럼 2/4박자 곡은 4분음표가 기본이므로 표준 4분음표 so는 G, 8분음표 la는 A/2, 2분음표 so는 G2로 표현한다.

QBH에서 색인자질에 대한 평가는 대체로 피치와 시간(temporal) 정보를 함께 사용하는 것이 좋은 것으로 보고되었다. 윤곽(contour)보다는 음정(interval)이, 음길이를 함께 사용하는 것이(McNab, Smith, Bainbridge, & Witten, 1997), 두 음표의 피치 변화량과 IOI 비율을 함께 사용하는 것이(Kline & Glinert, 2003), 피치와 IOI 비율을 함께 사용하는 것이 우수하였다(Dannenberg, Birmingham, Hu, Meek, Pardo, & Tzanetakis, 2007). 피치 정보를 나타내는 Absolute pitch와, absolute interval, key relative, 파스스 코드 중에서는 key relative가 가장 좋고, 파스스 코드가 가장 나쁜 결과를 보였으며, 음길이를 함께 사용하는 것이 피치만 사용하는 것보다 우수하였다(Hanna, Ferraro, & Robine, 2007).

3.3 오디오 파일에서 색인자질 추출

WAV, PCM, MP3 등과 같은 오디오 파일은 심볼문자로 변환이 쉽지 않기 때문에, Shazam과 같은 시스템을 제외한 대부분의 시스템은 Midi 파일을 대상으로 심볼문자로 색인하고, QBH나 QBE로 들어온 오디오 질의를 대상으로 오디

오 파일의 변화과 멜로디 분리 및 추출, 심볼문자로의 변환 등을 연구해왔다(<표 1> 참조).

WAV, PCM, MP3와 같은 오디오 파일은 먼저 샘플링(sampling)을 통해 연속적인 소리 신호를 이산적인 데이터 값으로 변환한 후, 색인자질로 음향의 신호 자질이나 오디오 핑거프린팅 정보, 혹은 심볼문자 정보 등을 추출한다.

신호 자질로는 Loudness, Pitch, Tone(brilliance and bandwidth), Mel-filtered Cepstral Coefficients(MFCCs), Derivatives 등이 사용되었다(Wold, Keislar, & Wheaton, 1996).

오디오 핑거프린팅은 다양한 방법들이 소개되었다(박만수, 김희린, 2006; Cano, Batlle, Kalker, & Haitsma, 2005). Shazam의 핑거프린트는 시간-주파수 스펙트럼에서 강도(intensity)가 높은 peak intensity의 주파수를 나타낸다. 10 초 동안 몇 지점에서의 peak의 주파수(Hz)로 나타낸다(Wang, 2003). 세 가지 핑거프린팅을 평가한 연구에서 Baluja의 방법이 가장 우수하였으며, 방법과 관계없이 질의가 길수록(15초) 결과가 우수하였다(Chandrasekhar, Sharifi, & Ross, 2011). 음악에서 보컬만 추출하여 핑거프린팅으로 색인한 DB에서 허밍질의를 핑거프린팅으로 유사도(유클리디안 거리)로 검색하여도 높은 정확률을 보였다(Bandera, Barbancho, Tardón, Sammartino, & Barbancho, 2011; Chew, Georgiou, & Narayanan, 2008).

오디오 파일에서 심볼문자 자질을 추출하는 과정은 다성 파일에서 멜로디를 추출하는 것으로 시작된다. 노래를 반주와 분리한다든지, 주 악기의 선율을 협주 악기의 선율과 분리하거나, 다중 트랙의 Midi 파일에서 멜로디 트랙을 분리하는 것이다.

오디오에서 주파수 영역으로 멜로디(중/고 주파수)와 베이스(낮은 주파수)를 구별할 수 있다(Goto, 2004). Midi에서 피치 값(0-127)을 기반으로 평균값이 낮고 편차가 큰 트랙을 반주 트랙으로 간주하여 제거할 수 있다(김무정, 낭종호, 2011). 기본적으로 오른손은 멜로디를 왼손은 반주를 담당하는 피아노곡에서도 원칙을 벗어난 예외 때문에 멜로디 검출은 쉽지 않다(최윤재, 박종철, 2009).

오디오에서 피치 트래킹은 멜로디 트래킹과 비슷하나, 다성 음악에서는 여러 개의 피치가 동시에 나타날 수 있으므로 피치 트래킹이 더 복잡하다. 마이크를 통해 입력된 허밍질의나 음악 파일에서 피치나 음정, 음길이와 같은 심볼문자 정보를 추출하기 위한 방법으로는 ZCR(zero crossing rate)이나 ACF(autocorrelation function)를 주로 사용하는 시간 기반 방법과 (David, 2003; Wan & Liu, 2006), FFT(fast Fourier transformation)와 같은 주파수 기반 방법, HMM(hidden Markov model)과 같은 확률모델 등이 사용된다(<표 1> 참조).

HMM은 Wav 파일을 Midi 파일로 변환하거나(Tripathy, Chhaatre, Surendranath, & Kalsi, 2009), 오디오음을 음표(note)로 전사(Sheh & Ellis, 2003), 코드 인지(Chen, Shen, Srinivasamurthy, & Chordia, 2012; Cheng, Yang, Lin, Liao, & Chen, 2008)와 키 추출(Lee & Slaney, 2008; Papadopoulos & Tzanetakis, 2012) 등에도 좋은 성능을 보였다.

질의의 키(key)를 추출하는 방법으로 PLCA(probabilistic latent component analysis)가 REPET(REpeating Pattern Extraction Technology)보다 우수하나, 사람이 부여하는 정보

다는 좋지 못하였다(Cartwright, Rafii, Han, & Pardo, 2011).

이밖에도 오디오에서 멜로디 추출, 평거프린팅, 코드 인지, 키 추출, 리듬검색을 위한 비트트래킹 등에 대한 다양한 연구는 ISMIR의 연간 proceedings에서 볼 수 있다.

3.4 색인단위 분할 및 정규화

저작권 검사의 단위이자 음악에서 의미를 갖는 최소단위는 동기(motif)이다. 하나의 동기는 2개의 마디로 구성되며, 한 마디의 음표의 수는 박자에 의해 결정된다. 음악을 색인할 단위로 나누는 데는 첫 번째 동기만으로(McNab et al., 1997), 혹은 전체 동기를 대상으로, 또는 주제(theme) 선율을 대상으로 분할한다.

주제 선율이 되는 부분은 보통 곡의 시작위치의 2마디, 4마디, 혹은 8마디가 될 수 있으나, 곡에 따라 음악의 시작 부분이 아닌 곳에서 주제선율이 생성될 수 있다. 보통은 반복되는 선율을 주제선율로 추출하나, 일정 길이 이상의 쉼표 후에 시작되는 선율을(유진희, 박상현, 2007), 동기간의 유사도를 기반으로 동기들을 클러스터링하여 각 클러스터를 대표하는 선율을 주제선율로 추출하기도 한다(구경이, 임상혁, 이재현, 김유성, 2003; Cilibarsi, Vitanyi, & Wolf, 2004). 주제선율 검색은 중요한 혹은 반복적인 선율만 색인함으로써 색인파일의 크기를 축소시켜 검색의 효율을 높일 수 있다.

음악에서 N-gram 색인은 주제선율을 대상으로 한 음씩 오른쪽으로 이동시키며 N개의 음들을 하나의 색인단위로 분할하는 것으로, 주로 음정 정보를 사용한다(Ghias et al., 1995;

Downie, 1999; Doraisamy & Ruger, 2002; Roger, Dannenberg, & Hu, 2004; Viro, 2011; Wang, Li, & Shi, 2006).

오디오 질의를 분할하는 방법으로는 시간을 기반으로 예를 들면 10ms 프레임이나(MIT와 U.Mich 시스템) 20ms 프레임(Meldex, Tunepal)과 같은 단위로 분할하거나, N-gram(Peachnote), HMM(Chew, Georgiou, & Narayanan, 2008) 등을 사용하여 분할한다(〈표 1〉 참조).

탐색의 성능을 높이기 위해 색인 문자열과 질의를 정규화 할 수 있다. 정규화란 동기의 길이를 일치시키는 것으로, 박자(meter)와 발생 시간(ontime) 대한 정규화가 있다. 박자 기반은 예를 들어 3/4박자의 곡에는 모든 음의 길이에 4를 곱하고 4/4박자 곡에는 3을 곱하여 두곡의 동기를 같게 조정하는 것이다. 발생시간 기반은 2분음표를 8분음표 4개로 분리(split)하거나, 반대로 8분음표 4개를 2분음표 하나로 합치는(union) 것이다. C-Brahms와 Meldex, Tunepal은 길이를 split로, SMILE은 피치와 길이를 정규화하였다(〈표 1〉 참조).

3.5 매칭방법

도치 색인파일로 구축된 심볼문자나 평거프린팅 정보는 완전일치 매칭에 유용하다. 파슨스 코드와 피치 검색에(Themefinder), 피치와 리듬에 대한 N-gram 검색에(Peachnote), 평거프린트 검색(Shazam) 등에 완전일치 검색이 사용되었다(〈표 1〉 참조).

그리나 허밍질의의 경우 불일치한 키(key)나 부정확한 음높이와 음길이 등으로 완전일치는 적합 음악을 검색하지 못하는 것으로 알려졌

다(McNab, Smith, Witten, & Cunningham, 1996). 따라서 QBH에서는 부정확한 질의 선율을 고려하여 완전일치보다는 부분일치(approximate matching) 기법이 사용된다. 부분일치 기법으로는 LCS(Longest common subsequence), 편집 거리(Edit distance), EMD(Earth Mover's Distance), 기하학적(Geometric) 기법, LS(linear scaling), DTW(dynamic time warping) 등이 있다.

LCS는 질의와 색인 두 문자열에서 일치하는 문자수 순으로 정렬하는 것으로, Ghias 등(1995)은 파슨스코드에 대해 사용하였다.

편집거리는 질의 문자열을 색인파일의 문자열로 변환하는데 필요한 최소한의 편집연산(삭제, 추가, 교체)의 수로 유사도를 계산한다. 파슨스 코드열에 대해(〈표 1〉의 Musipedia, MIT, U.Mich 시스템, Tripathy et al., 2009), 음정에 대해(Lemström & Pienimäki, 2007) 편집거리가 사용되었고, 피치와 음길이를 조합한 다양한 심볼문자 정보를 대상으로 N-gram, Geometric과 비교 평가되었다(Hanna, Ferraro, & Robine, 2007).

Musipedia에서 사용한 EMD는 색인단위로 분할된 멜로디나 질의를 피치와 time(rhythm)을 나타내는 2차원의 공간에서 점(point)들로 표현하여 질의와 색인된 선율의 거리를 계산하는 매칭기법으로, 리듬(tapping) 질의는 2차원 공간에서 모든 음의 피치는 동일한 것으로 간주한다(Typke, 2007). C-Brahms는 다성음악 검색에 사용할 기하학적 패턴매칭 알고리즘을 개발하였고(Lemström & Tarhio, 2003), 편집거리와 비교 평가하였다(Lemström & Pienimaki, 2007). 유사도 기법이 사용된 예로는 벡터공간

모델 기반의 코사인 유사도를 사용한 SMILE에서 볼 수 있다(〈표 1〉 참조).

다섯 개의 탐색 알고리즘을 비교한 연구에서 N-gram과 편집거리 매칭은 HMM이나 선율 윤곽 매칭보다 성능이 좋지 못하였다(Dannenberg et al., 2007). LS와 DTW를 함께 사용하여 성능을 향상시킨 연구도 수행되었다(Nam, Park, Park, Lee, & Kim, 2012).

4. 시스템 선정과 분석 방법

본 연구에서 분석 대상이 된 시스템은 2012년 12월 Google에서 content-based music이나, melody, song, tune, humming, note에 대한 search(혹은 retrieval)에 관한 주제로 탐색하여 웹에서 탐색 가능한 엔진이나 시스템으로 제한되었다. 〈표 1〉에 소개된 시스템 중 다수의 testbed 시스템은 일정 기간이 지난 후 웹에서 사라졌고, C-Brahms는 인터페이스의 접속은 가능하나 실행이 되지 않아 제외되었다. ABC Tune finder처럼 동일 탐색엔진이 서로 다른 사이트에서 운영되는 시스템은 하나만 포함시켰고, Song Tapper(<http://www.bored.com/songtapper>)처럼 리듬만 키보드로 태핑하는 단순 시스템은 제외되었다. 스마트폰 어플은 포함시키지 않고

PC기반 시스템으로 웹에서 접근 가능한 시스템으로 제한하였다. 조사결과 총 15개의 탐색 엔진이 분석되었다.

분석에 사용된 분석틀은 선행연구를 통하여 연구된 파일의 변환, 멜로디 추출 및 분할, 색인자질 추출과 색인, 매칭기법과 같은 CBMR 시스템 구성요소로 언급된 기술 중 탐색시스템을 통하여 확인할 수 있는 요소로 제한되었다. 따라서 확인할 수 없는 색인자질대신에 접근점을 사용하였다. 분석틀은 소스데이터, 탐색 도구, 접근점, 탐색 기능, 출력물, 데이터베이스 성격과 크기 등으로 구성되었다(〈표 2〉 참조). 시스템은 연구보고서나 홈페이지를 참조하고 탐색을 통해 분석되었다.

탐색질의로는 DB 음악장르에 따라 Alphabet song, Yesterday, Three blind mice, Auld lang syne, 잊혀진 계절 등이 사용되었지만, Alphabet song의 첫 3마디는 Mozart의 Variations in C on theme: Ah, Vous Dirai-Je, Maman, K 265의 선율과 동일하므로 장르와 관계없이 클래식 악보검색에서도 우선적으로 사용되었다. 〈그림 4〉에서와 같이 이 두곡은 음 하나의 길이(duration)가 다르기 때문에 음길이 정규화나 완전일치 여부를 판정하는데 매우 유용하였다. 뿐만 아니라 Alphabet song은 키(key) 변환 검색의 여부를 확인하는 데도 유용하였다.

The image shows two musical staves side-by-side. The top staff is labeled "Alphabet Song (Twinkle, Twinkle Little Star)" and the bottom staff is labeled "Mozart. Wolfgang Amadeus: Variationen für Klavier in C KV 265". Both staves begin with a treble clef. The top staff has a key signature of one sharp (F#) and a common time signature. The bottom staff has a common time signature. The notation consists of vertical stems with small horizontal dashes indicating pitch and duration. In both cases, the first three notes are identical: a quarter note followed by a eighth note, and another quarter note. The fourth note is where they differ: the top staff has a eighth note, while the bottom staff has a sixteenth note. This visual comparison highlights the difference in note duration between the two pieces, which is a key factor in the search algorithm described in the text.

〈그림 4〉 Alphabet song과 Mozart 피아노 변주곡

〈표 2〉 15개 CBMR 시스템에 대한 분석표

		ABC tune finder	DoDo SoSo	Folkture finder	Kooplet	Meldex	Melody Catcher	Midomi	Musipedia
입력	악보		U		pdf	xml			xml
	오디오파일							U	
물	심볼레이터	abc		abc		midi	midi		abc
다성음악									*
단위	beginning		*	*		옵션	*		*
위	anywhere	16음			*	*	*	*	옵션
탐색도구	마이크로폰				*	*		10초	*
	가상 건반		오선지	5음<	*	*	5-7음		5음<
	텍스트 코드	ud	3음<	↑↓→					파슨스
접근점	피치	*	*	*			*	U	
	피치+길이				*				*
	interval					옵션(+리듬)			
	contour	ud		↑↓→		옵션(+리듬)			udr
	rhythm				*	옵션			태평
	기타								
제한점									
음길이 정규화						*			
어구/용어절단 탐색					“어구”				
키(key) 변환 검색	*	C 만	*	*	*	*	*	*	*
제한/필터링/refine				키/박자/ 악기					
브라우즈									
매칭	완전일치	*	*				*		
	부분일치	*		*	*	*		*	*
결과 출력물	메타데이터만		*곡명						
	오디오 파일	midi		midi	*/p	wav,midi,aiff	midi/p	*video	YouTube링크
	악보(p부분)	pdf,ps,gif		*	*xml				pdf/p
	코드 파일	abc,txt		abc					파슨스
	외부소스로 링크	*	*	*	*		*	*	*
하이라이트				색	색				
기타 출력물									
정렬	곡명순	U		적합성, 길이, 최신곡, 속도			U		
웹탐색									*
DB	웹크롤링			abc	악보		Midi		
		200 abc콜렉션	이용자 질의	201. 467 민요 틈	787,036악보	9354민요, 악보 Midi	60,000선율	200만 곡, 질의	클래식, 민요 팝
기타	메타데이터검색 (브)라우즈	곡명		곡명	곡명, 작곡가, 가사/어구	곡명 (+브라우즈)	곡명, 작곡가		곡명, 가사/어구

		myMoodle	Parfait ole	Peachnote	SID	Themefinder	Tunebot	Tunepal
입 력 물	악보		*	pdf		xml		
	오디오파일				sid			
	심볼레이터	U					midi	abc
다성음악				*		*		*
단 위	beginning		*			*		
	anywhere	*		15그램	*	음션	*	*
탐 색 도 구	마이크로폰						10초	12초
	가상 건반/			*				
	텍스트 코드	udr	4-32음		4음⟨	*		15음⟨
접 근 점	피치		*	*	*	*		*
	피치+길이						*	
	interval					*	*	
	contour	udr				*		
	rhythm			*			*	*
	기타			코드		기, 박자		
제한점				정규화				키
음길이 정규화								*
어구/용어절단 탐색		절단(?#)			절단(.*)	절단(.?*)		
키(key) 변환 검색		*	*	*	*	음션	*	
필터링/refine				년도				
브라우즈								
매 칭	완전일치	*	*	*	*	*		
	부분일치						*	*
결 과 출 려 물	메타데이터only				*			
	오디오 파일	midi	midi/p			midi/p	midi	midi
	악보(p:부분)	p	p	page		gif/p		*
	코드 파일							abc
	외부소스로링크			*		*	*	*
하이라이트		색						
기타 출력물				빈도/년도				
정렬		작곡가	U	파일명	작곡가(클래식)			
웹탐색								
DB	웹크롤링							
		116,632곡, 464,820선율	6000선율	65000클래식악 보, 160면page	SID콜렉션	motet19000 클래식theme 민요10000	3000노래, 11000질의	20-30콜렉션, 23137선율
기 타	메타데이터탐색					motet(곡명, 작곡가, 출판사) 민요(곡명, 나라)	곡명, 연주자	곡명

〈표 2〉에서 제공된 특성은 *로, 상업적인 이유 등으로 시스템에 대한 설명자료가 충분치 않아 특성을 분명하게 알 수 없는 것은 U(unknown)으로 표시하였다. 시스템의 탐색방법과 탐색결과물에 따른 시스템의 유형은 다차원척도로 분석되었다. ALSCAL(M)을 사용하여 이분형 유클리디언 거리 측도로 분석되었다(제공된 특성(*)은 1, 나머지는 0).

5. 결과 분석

5.1 데이터베이스

조사된 CBMR 시스템은 자체 구축한 음악 DB를 대상으로 색인하여 탐색을 제공하거나, 웹 탐색엔진처럼 외부 콜렉션이나 사이트를 대상으로 색인파일만 유지하고 음악은 외부 소스로 링크시키는 시스템으로 존재하였다.

악보를 대상으로 DB를 구축하는 Meldex나 Musicpedia, Peachnote, Themefinder는 악보를 XML이나 Midi 형식으로 변환하나, Tunepal은 ABC 파일로 변환하여 색인자질을 추출한다.

ABC Tune finder는 외부에서 제작된 ABC 콜렉션에 대해 색인파일만 작성하고 소스는 ABC 콜렉션으로 링크시킨다. Folktunefinder는 ABC 파일을 대상으로, MelodyCatcher는 Midi 파일을 대상으로, Kooplet은 포맷은 관계 없이 모든 악보를 대상으로, 웹에서 크롤링하여 색인파일을 구축하고 탐색 결과 음원은 웹사이트로 링크시킨다. Musipedia는 메타엔진처럼 다른 탐색엔진(Alexa Internet)으로 질의를 보내 검색된 결과를 소스로 링크시키는

서비스도 수행한다.

DoDoSoSo와 Midomi는 탐색자가 마이크로 노래한 질의 선율을 데이터베이스에 추가하여, Tunebot는 Karaoke Callout이라는 게임을 통해 키(key)가 다른 노래들을 수집하는 등의 기능을 통해 데이터베이스 구축에 이용자를 참여시킨다.

본 연구에서 분석된 시스템들의 DB 규모는 비교적 대규모이다. Folktunefinder는 20만 정도의 선율을, Midomi는 200만 음악을, myMoozle은 46만 선율을, Peachnote는 160만 페이지의 악보를, Kooplet은 78만 악보를 색인하고 있으며, 계속 증가하고 있다.

Parfait Olé와 Peachnote는 클래식 음악을 대상으로, Folktunefinder와 MelodyCatcher는 민요를 대상으로 서비스를 제공하나, ABC Tune finder나 Midomi, Musipedia, Themefinder 등은 장르에 관계없이 다양한 음악에 대한 탐색을 제공한다. Midomi는 한국가요까지 색인한다.

5.2 색인 단위

대부분의 CBMR 시스템은 멜로디나 노래를 추출하여 단성 음악을 색인하나, Musipedia나 Peachnote, Themefinder, Tunepal은 다성 음악도 색인하고, Peachnote는 코드(chord) 검색을 제공한다.

시작부분(첫째 동기)으로만 색인하는 DoDoSoSo와 Folktunefinder, Parfait Ole를 제외하고 다른 시스템들은 출현위치와 관계없이 모든 동기나 주제 선율로 색인한다. Meldex, MelodyCatcher, Musipedia, SID, Themefinder는 첫 동기 탐색과 모든 동기 탐색(search anywhere)을 옵션

으로 제공한다. Peachnote는 출현위치와 관계 없이 음표를 15-gram 색인한다.

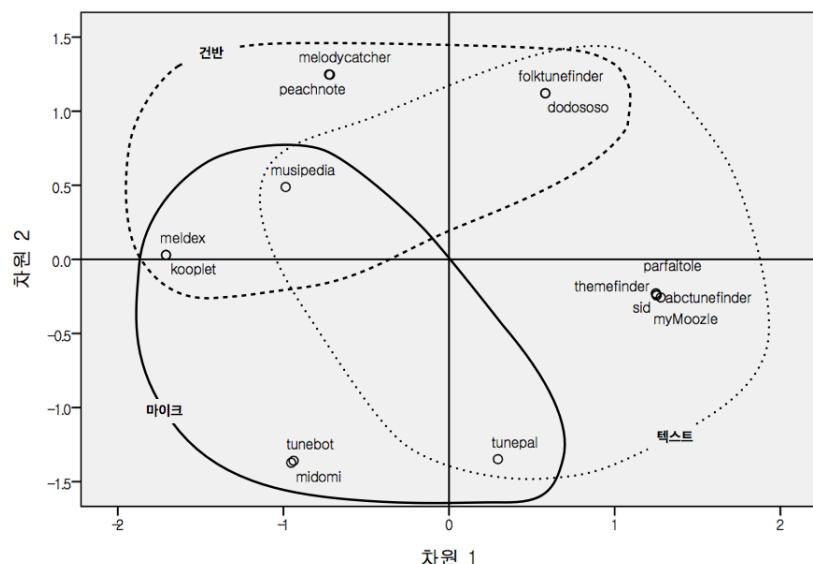
5.3 질의 종류와 탐색 접근점

〈그림 5〉는 탐색질의를 입력하는 방법으로 시스템을 다차원척도로 분석한 것이다. ABC Tune finder와 myMoozle, Parfait Ole, SID, Themefinder는 텍스트 박스를 통한 심볼문자 탐색만 제공하고, MelodyCatcher와 Peachnote는 가상건반 탐색만, Midomi와 Tunebot는 마이크로 폰을 사용한 QBH 탐색만 제공하나, 다른 시스템들은 두 개 이상의 탐색방법을 제공한다. DoDoSoSo은 텍스트 박스 입력과 함께 오선지에 음표를 그리는 고급탐색을 제공한다.

マイ크를 사용하는 시스템 중 Meldex와 Midomi, Musipedia, Tunebot은 QBH이고, Tunepal은 QBP 시스템이다. Tunepal은 연주된 선율을

ABC로 변환하여 혹은 직접 텍스트 박스에 입력한 ABC 코드를 탐색자질로 사용한다. 연주는 12초 이상, 텍스트로 입력한 코드는 최소 15개 이상 사용해야만 검색된다. Midomi와 Tunebot는 10초 이상 노래하라고 권하고 있다.

텍스트 박스에 음표를 심볼로 직접 입력하거나 가상건반으로 입력하는 시스템에서 〈그림 3〉에서 보는 바와 같이 다양한 기호들이 사용된다. myMoozle은 파슨스 코드로, DoDoSoSo는 do re mi …로, Musipedia는 ABC 기호, Peachnote는 exact interval을 사용한다. 심볼문자 검색만 가능한 Themefinder는 다양한 심볼 문자 탐색을 제공한다. a부터 g까지 7개 solfege 기호를 사용하는 Pitch 탐색, 반음정을 1도로 하여 8도까지에 다시 장(major), 단(minor), 증(augmented), 감(diminished), 완전(perfect) 등을 결합해서 사용하는 Interval 탐색, “1234567”을 사용하는 Scale Degree 탐색, 파슨스 코드대신



〈그림 5〉 질의 입력방법에 따른 다차원척도

/ \ -를 사용하는 Gross Contour 탐색, 음정이 기준치보다 크고 작음을 uUdDs로 나타내는 Refined Contour 탐색과, 키(Key) 탐색과 함께 박자(meter) 탐색까지 제공한다. Folktunefinder 는 가상 건반 탐색과 함께 ↑, ↓, →를 사용한 Contour 탐색도 제공한다.

가상 건반으로 질의를 입력하는 시스템은 5개의 옥타브까지 사용하는 Kooplet을 제외하고 대부분 두(2) 옥타브 정도의 건반(14~17건반)만 제공하고 있다. C-Brahms가 88개 건반(11옥타브)를, Hundrum이나 ABC 기호로 텍스트 탐색 하는 시스템은 4~5옥타브의 음표까지 표현할 수 있음에 비하면 비교적 간단하다. QBH 탐색과 함께 건반탐색을 제공하는 Musipedia는 건반에 음길이까지 입력하나, Folktunefinder와 MelodyCatcher, Peachnote는 피치만 사용한다. Meldex도 건반에 음길이까지 입력하나 음길이의 사용여부와 interval과 contour 중 어느 것을 피치 검색에 사용할지를 Preferences에서 지정할 수 있다.

이 밖에도 Kooplet과 Peachnote는 음표길이를 사용한 리듬 탐색을, Musipedia는 컴퓨터자판의 space 키로 리듬을 태핑(tapping)하는 리듬 탐색을 제공하며, Peachnote는 코드(chord) 탐색을 제공한다.

5.4 탐색 및 매칭 방법

이용자에게 익숙하지 않은 absolute pitch나 relative pitch, key relative 등을 색인자질로 사용한 시스템은 이용자의 편의를 위해 건반이나 solfege와 같은 텍스트코드를 접근점으로 제공하고 입력된 질의를 색인에 사용된 자질로

바꾸어 탐색한다. Peachnote는 건반에 “도도솔 솔라리솔”을 중간음으로 치면 exact interval “48 0 7 0 2 0 -2”로, 높은 음으로 치면 “60 0 7 0 2 0 -2”로 질의를 변환하고, ABC Tune finder 도 “ccggaag”로 입력하면 relative pitch “040107”로 변환하여 탐색한다. Relative pitch나 exact interval로 색인하면 키가 다른 음악까지 검색 할 수 있기 때문에 ABC Tune finder와 Parfait Ole(relative pitch), Peachnote(interval)는 피치 탐색에서 이 방법으로 키 변환 탐색을 제공한다. 그러나 SID는 입력된 질의를 12개의 키(Key)로 자동 변환하여 탐색한다. Themefinder와 Tunepal 은 키 탐색을 접근점으로 제공하며, DoDoSoSo는 C 키의 음악만 검색한다.

ABC Tune finder와 DoDoSoSo, MelodyCatcher, myMoozle, Parfait Ole, Peachnote, SID, Themefinder는 질의와 완전일치한 곡만 검색 한다. SID와 Themefinder는 텍스트 검색에서 처럼 와일드 카드(.과 ? *)를 사용하여 탐색선율의 중간에 있는 음표를 절단 검색할 수 있다. myMoozle은 ?(음표 하나)와 #(음표 5개까지)로 절단할 음표의 길이까지 구별한다.

음길이에 대한 정규화는 Peachnote는 탐색 옵션으로 제공하나, Meldex와 Tunepal은 자동으로 split 정규화한다.

Musipedia은 파슨스 코드열에 대해서는 편집 거리를, 피치와 음길이 검색이나 리듬 검색에는 EMD를 사용한다. Meldex와 Tunebot, Tunepal 은 편집거리를 사용한다. 유사도 탐색을 제공하는 Kooplet은 완전일치한 멜로디 검색을 위해 따옴표를 사용한 어구(phrase) 탐색도 제공한다.

5.5 탐색 결과 출력물

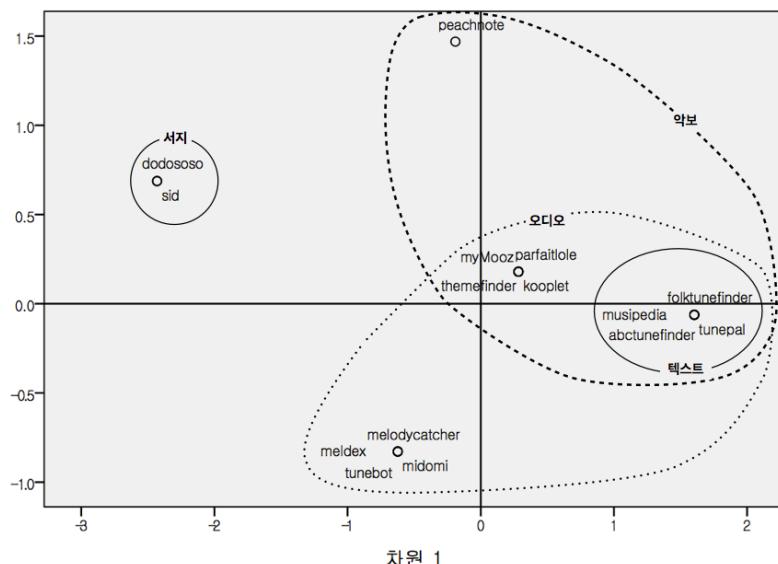
〈그림 6〉은 탐색결과로 제공되는 출력물에 따라 시스템을 다차원척도로 분석한 것이다. ABC2MIDI, MIDI2ABC, Score2midi, Abc2xml 등과 같은 s/w를 사용하여 악보와 Midi, ABC 파일 사이에는 상호 변환이 가능하기 때문에, 대부분의 시스템은 DB 소스 파일의 종류와 관계 없이 메타데이터와 함께 다양한 출력물을 제공한다.

ABC 콜렉션을 탐색하는 ABC Tune finder와 Folktunefinder, 악보를 ABC 파일로 변환하여 색인하는 Tunepal은 검색결과를 Midi 파일과 악보로 변환하여 제공하며, 스트링 정보도 txt나 abc 파일로 제공한다. Musipedia도 오디오와 악보와 함께 스트링정보를 파슨스 코드로 보여준다(View details에서). 그러나 Meldex와 MelodyCatcher, Midomi, Tunebot은 오디오만,

Peachnote는 악보만 제공하며, DoDoSoSo나 SID는 피치로 검색된 선율에 대해 곡명과 같은 메타데이터만 제공한다.

MelodyCatcher나 Parfrait Ole, Themefinder를 제외하고 오디오 파일의 경우는 대체로 곡 전체를 들을 수 있게 제공되지만, Kooplet은 일치한 부분만, Midomi는 일치부분이나 전곡을 선택하게 한다. 악보의 경우는 ABC Tune finder와 Folktunefinder, Kooplet, Tunepal을 제외하고 탐색 선율과 일치하는 선율의 전후 부분만을 제공한다.

웹에 있는 음악을 크롤링하여 색인한 시스템은 외부 소스로 링크를 제공하나, Peachnote는 검색된 악보 외에 음악파일을 YouTube로, Tunebot은 iTune이나 Amazon에서 구매하도록 링크를 제공하고 있다. Musipedia는 자체 구축한 DB에서 탐색하는 외에 Alexa Internet을 통한 웹 탐색도 제공한다.



〈그림 6〉 출력물에 따른 차원척도

이밖에도 Peachnote는 악보와 함께 질의와 일치한 N-gram의 빈도수를 연도별 그래프로 제공하며, 연도별 필터링도 제공한다. Folktunefinder와 Kooplet, myMoozle은 검색된 악보에서 탐색질의의 음표를 붉은색으로 강조함으로써 적합성 판정을 돋는다.

Parfait Ole(<http://www.parfaitole.com>)는 탐색질의에 검색된 선율을 재생할 악기(클라리넷이나, 플룻, 오르간, 피아노)를 지정하여 원하는 음색으로 재생할 수 있다.

Folktunefinder는 검색된 결과에 대하여 키나, 박자, 악기 등으로 필터링하는 제한(refine) 탐색과 함께 브라우즈 탐색도 제공한다.

탐색결과는 완전일치 검색에서는 대체로 곡명순으로, 부분일치 시스템에서는 적합성순으로 정렬되지만, Kooplet은 적합성과 함께 최신곡, 빠른/느린 곡, 파일의 크기 순서로 정렬이 가능하다.

5.6 기타 메타데이터 탐색

메타데이터 검색도 함께 제공하는 CBMR 시스템에서 메타데이터 탐색은 접근점 확대는 물론 내용기반으로 검색된 곡을 곡명으로 검색하여 재현율을 확인하는데 유용하였다.

ABC Tune finder와 Folktunefinder, Meldex, Tunepal은 곡명 탐색을, MelodyCatcher와 Tunebot는 곡명과 작곡가/연주가 탐색을, Meldex는 곡명 브라우즈도 제공한다. Musipedia는 가사를 대상으로 키워드 탐색과 어구(phrase) 탐색을, Kooplet는 불리언 연산자를 사용하는 작곡가, 곡명, 가사 키워드 탐색과 어구(phrase) 탐색을 제공한다. Themefinder는 라틴 motet

탐색에서 작곡가, 곡명, 출판사 등을, 민요탐색에서는 나라와 곡명, 장르와 같은 서지 검색을 제공한다.

6. 결 론

본 연구에서 검토된 15개의 웹기반 CBMR 시스템은 DB나 탐색방법 측면에서 매우 다양하였다. Lee와 Moon(2006)에서와 같이 가상 전반이나 텍스트 박스에 편집이나 음길이를 정확하게 입력하여야하는 QBN보다는 마이크로 노래하거나 음악을 재생하여 탐색하는 QBE나 QBH가 비음악전공 연구자가 사용하기에 편리하였다. 연구결과 다음과 같은 특성과 문제점이 검토되었다.

첫째, CBMR 시스템에서 상징문자로 표현되는 음악정보를 색인하고 탐색하는 데는 서지DB에서 사용되는 텍스트 정보 처리 및 검색 기법이 적용되고 있음을 알 수 있었다. 도치색인, N-gram 색인, 불리언 탐색, 용어절단검색(myMoozle, SID, Themefinder), 키워드 및 어구 탐색(Kooplet), 음길이 정규화(Meldex, Peachnote, Tunepal), 년도별 필터링(Peachnote), 키, 박자, 악기별 필터링과 브라우징(Folktunefinder), 적합성과 발표년도, 길이, 빠르기순 정렬(Kooplet), 검색된 악보에서 질의 음의 하이라이트(Folktunefinder, Kooplet, myMoozle) 등은 CBMR의 검색성능을 향상시키기 위한 좋은 도구로 보인다.

둘째, DB의 성장 측면에서 조사된 CBMR 시스템은 실험시스템과 실용시스템으로 구별되었다. Meldex나 Themefinder와 같은 연구 기반의 실험시스템은 DB의 확장에 소극적으로

대응하여 일반이용자가 사용하기에 유용해 보이지 않지만, Tunepal은 작으나마 색인DB를 확대하는 노력(2013년 5월 22,751선율에서 7월 23,137선율로 증가)과 함께 iPhone 기반의 모바일 서비스도 제공하고 있다.

실용시스템들은 대규모의 자체 DB를 구축하거나(Midomi, myMooze, Parfait Ole, Peachnote), ABC 파일(Folk tune finder), 악보(Kooplet), Midi 파일(MelodyCatcher)을 대상으로 웹에서 크롤링하여, 혹은 탐색자가 노래한 질의 선율을 DB에 추가하는(DoDoSoSo, Midomi, Tunebot) 등의 노력으로 DB를 확장하고 있다.

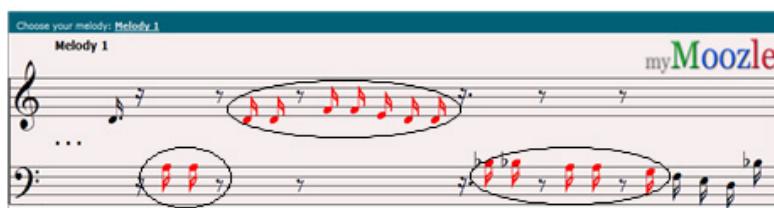
이밖에도 아마존이나 iTune과 같은 음반 판매 DB와 연계하여 내용기반 검색을 제공하는 Midomi는 스마트폰 기반의 SoundHound도 제공하여 실용성을 높이고 있다.

셋째, 완전일치 검색에서 입력한 질의의 길이와 색인문자열의 길이가 일치하지 않을 때의 문제이다. ABC Tune finder나 DoDoSoSo, Melody Catcher, Musipedia, Parfait Ole 등 대부분의 시스템에서 탐색질의는 자동 우측절단 탐색을 수행한다. 그러므로 질의음의 길이가 짧을수록 탐색 결과는 많아지고 정확률이 낮아진다. DoDoSoSo에서 2개의 음으로 탐색하면 검색결과를 보여주지 않고 탐색결과가 너무 많으니 질의를 길게 입력하여 결과를 축소하라는 메시지만 보여

준다. MelodyCatcher도 결과가 너무 많으면 검색질의를 길게 입력하라는 탐색힌트를 보여준다. 그러나 Tunepal은 입력한 길이가 15음표나 10초보다 짧으면 검색하지 않고 15음 이상을 입력하라고 안내한다. 색인단위보다 질의가 더 길 경우 ABC Tune finder는 16개 이상의 질의 음표(즉 15개의 음정으로 색인되기 때문에)에 대해 질의를 16개까지 자동으로 우측 절단하여 탐색한다. Peachnote는 15그램 질의로 탐색하여 일치한 결과가 없으면 15보다 작은 그램의 질의로 탐색하여 설명과 함께 사용된 질의와 결과물을 보여준다.

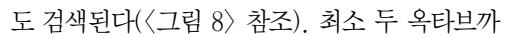
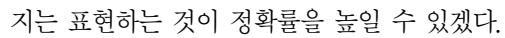
넷째, 선율이나 주선율을 탐색하는 시스템에서 선율을 추출하는데 문제가 있어 보였다. <그림 7>은 myMooze에서 Alphabet song의 선율 “도도솔솔라리솔파미미레레도”를 *RURURD DRDRDRD로 탐색하여 검색된 곡(Bulgarian-sasho Roman Moi Angele)의 악보이다. 반주 음과 멜로디가 잘 분리되지 않아 질의 앞 두(2) 음은 반주에, 그 다음 7개의 음은 멜로디에, 나머지는 다시 반주에 출연한 곡이 검색되었다.

다섯째, ABC Tune finder는 Contour 탐색에서 UDS 중 S는 생략하고 U와 D만 사용한다. 따라서 Alphabet song의 선율 “도도솔솔라리솔파미미레레도”는 “UUDDDDD”로 탐색하여야 한다. 이 사실을 모르고 파운드코드



<그림 7> myMooze의 검색결과

탐색에 익숙한 이용자의 탐색을 돋기 위해서는 탐색질의에 사용된 S도 자동으로 제외시켜 탐색질의를 수정하여 매칭할 필요가 있다. 색인 시에 색인어 추출을 위한 언어학적 처리는 탐색질의를 처리하는데도 똑같이 적용되어야 탐색성능을 높일 수 있다.

여섯째, 피치를 표현하는데 옥타브를 표현하지 않는 것보다는 ABC 기보처럼 옥타브까지 표현하는 것이 정확한 음악을 검색하였다. Themefinder에서 사용된 Hundrum **kern은 피치를 옥타브 까지 표현하나 Themefinder의 Pitch 탐색은 옥타브를 무시하기 때문에, Alphabet song의 선율 “도도솔솔라리솔”을 “ccggaag”로 탐색하면 과 함께 과 도 검색된다(<그림 8> 참조). 최소 두 옥타브까지는 표현하는 것이 정확률을 높일 수 있겠다.

이밖에도 인터페이스의 사용성이 본 연구의 분석 요소는 아니지만 DB에 따라 3종류의(Folkfull, Folktest, Miditest) 탐색 인터페이스를 선택해야하는 Meldex 시스템은 하나의 인터페이스에서 DB를 선택하게 하는 것이 사용성 측면에서 더 편리해 보였다.

노래 오디오 파일에서 자동으로 가사를 추출하여 색인/탐색하거나(Mesaros & Virtanen, 2010), 노래에 맞춰 가사와 악보를 동시에 다중 형태(multi-modal)로 제공하는(Kan, Wang,

Iskandar, Nwe, & Shenoy, 2008) 등의 최신 기술은 분석 대상 시스템에서 아직 적용되지 않고 있다.

CBMR에 사용된 기술의 현황과, 위에서 제기된 시스템의 장점과 문제점, 발전방향을 제시한 연구결과는 CBMR 시스템의 성능 향상은 물론 전자도서관에서 다양한 음악자료를 수집하여 내용기반 검색을 제공하는 시스템을 개발 또는 선정하는데 시스템 명세(specification)로도 활용될 수 있을 것이다.

본 연구는 웹 기반의 CBMR 시스템간의 특성을 비교했을 뿐 시스템간의 성능을 평가하지 않았다. CBMR 시스템은 이제 이용자들에게 소개되기 시작하여 향후 꾸준히 이용이 증가할 것으로 예측된다. DB가 증가할수록 이용자는 보다 정확한 검색을 원할 것이다. 동일 시스템에서 마이크나 건반과 같이 탐색도구에 따른 성능의 차이나, 피치나 interval, contour 등과 같은 탐색 접근점에 따라 정확률이나 재현율 측면으로 성능을 비교 평가하는 후속연구가 필요할 것이다. 본 연구는 CBMR 시스템이나 엔진을 대상으로 특성을 연구하였다. 내용기반 음악정보 기반의 무드(mood)나 장르 분류 혹은 추천 시스템 등은 포함되지 않았다. 탐색 시스템 외에 이들 시스템의 특성에 대한 연구도 필요할 것이다.



<그림 8> Themefinder의 검색결과

참 고 문 헌

- 구경이, 임상혁, 이재현, 김유성 (2003). 주제 선율 색인을 이용한 내용 기반 음악정보 검색 시스템. 데이터베이스 연구, 19(3), 34-45.
- 김무정, 낭종호 (2011). Query By Humming 응용을 위한 Midi 파일에서의 자동 멜로디 트랙 선택 방법. 한국정보과학회 한국컴퓨터종합학술대회 논문집, 38(1B), 405-408.
- 노정순 (2011). 정보검색: 이론과 실제. 대전: 글누리.
- 박만수, 김희린 (2006). 실제 잡음 환경에 강인한 오디오 핑거프린팅 기법. Telecommunications Review (SK Telecom), 16(3), 435-446.
- 유진희, 박상현 (2007). 허밍 질의 처리 시스템의 성능 향상을 위한 효율적인 빈번 멜로디 인덱싱 방법. 정보과학회논문지: 데이터베이스, 34(4), 283-303.
- 최윤재, 박종철 (2009). 음악의 특성에 따른 피아노 솔로 음악으로부터의 멜로디 추출. 정보과학회논문지: 컴퓨팅의 실제 및 테터, 15(12), 923-927.
- Arifi, V., Clausen, M., Kurth, F., & Muller, M. (2003). Automatic synchronization of music data in score-, MIDI-, and PCM-format. Proceedings of ISMIR 2003. Retrieved from <http://ismir2003.ismir.net/papers/Arifi.pdf>
- Bainbridge, D. (2004). Music information retrieval research and its context at the University of Waikato. Journal of the American Society for Information Science and Technology, 55(12), 1092-1099. <http://dx.doi.org/10.1002/asi.20062>
- Bandera, C. de la, Barbancho, A. M., Tardón, L. J., Sammartino, S., & Barbancho, I. (2011). Humming method for content-based music information retrieval. Proceedings of ISMIR 2011, 49-54.
- Cano, P., Batlle, E., Kalker, T., & Haitsma, J. (2005). A review of audio fingerprinting. Journal of VLSI Signal Processing, 41, 271-284. <http://dx.doi.org/10.1007/s11265-005-4151-3>
- Cartwright, M. B., Rafii, Z., Han, J. Y., & Pardo, B. (2011). Making searchable melodies: Human versus machine. Proceedings of Human Computation, 2011. Retrieved from http://www.cs.northwestern.edu/~jha222/paper/2011_humancomp_cartwright_etal.pdf
- Chai, W., & Vercoe, B. (2002). Melody retrieval on the web. Proceedings of ACM/SPIE Conference on Multimedia Computing and Networking, 226. <http://dx.doi.org/10.1117/12.449982>
- Chandrasekhar, V., Sharifi, M., & Ross, D. A. (2011). Survey and evaluation of audio fingerprinting schemes for mobile query-by-example applications. Proceedings of ISMIR 2011, 801-806.
- Chen, R., Shen, W., Srinivasamurthy, A., & Chordia, P. (2012). Chord recognition using duration-explicit hidden Markov models. Proceedings of ISMIR 2012, 445-450.

- Cheng, H. T., Yang, Y. H., Lin, Y. C., Liao, I. B., & Chen, H. H. (2008). Automatic chord recognition for music classification and retrieval. Proceedings of the IEEE International Conference on Multimedia and Expo 2008, 1505-1508.
<http://dx.doi.org/10.1109/ICME.2008.4607732>
- Chew, E., Georgiou, P., & Narayanan, S. (2008). Challenging uncertainty in query by humming systems: A fingerprinting approach. IEEE Transactions on Audio, Speech, and Language Processing, 16(2), 359-371. <http://dx.doi.org/10.1109/TASL.2007.912373>
- Cilibrasi, R., Vitanyi, P., & Wolf, R. (2004). Algorithmic clustering of music based on string compression. Computer Music Journal, 28(4), 49-67.
- Dannenberg, R., Birmingham, W. P., Hu, N., Meek, C., Pardo, B., & Tzanetakis, G. (2007). A Comparative evaluation of search techniques for query by humming using the MUSART testbed. JASIST, 58(5), 587-701.
- David, G. (2003). Pitch extraction and fundamental frequency: history and current techniques. Technical report TR-CS/2003-06. Retrieved from
<http://audio-fingerprint.googlecode.com/svn-history/r62/trunk/referencias/2003-06.pdf>
- Duggan, B., O'Shea, B., Gainza, M., & Cunningham, P. (2009). Compensating for expressiveness in queries to a content based music information retrieval system. Proceedings of the International Computer Music Conference (ICMC 2009), 33-36.
- Doraisamy, S., & Ruger, S. (2002). Robust polyphonic music retrieval with n-grams. Journal of Intelligent Information Systems, 21(1), 53-70. <http://dx.doi.org/10.1023/A:1023553801115>
- Downie, S. (1999). Evaluating a simple approach to music information retrieval: Conceiving melodic N-grams as text. Unpublished doctoral dissertation. Univ. of Western Ontario. USA.
- Ghias, A., Logan, J., Chamberlin, D., & Smith, B. (1995). Query by humming: Musical information retrieval in an audio database. Proceedings of the 3rd Annual ACM International Conference on Multimedia, 231-236.
- Goto, M. (2004). A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. Speech Communication, 43(4), 311-329. <http://dx.doi.org/10.1016/j.specom.2004.07.001>
- Hanna, P., Ferraro, P., & Robine, M. (2007). On optimizing the editing algorithms for evaluating similarity between monophonic musical sequences. Journal of New Music Research, 36(4), 267-279. <http://dx.doi.org/10.1080/09298210801927861>
- Hug, A., Cartwright, M., & Pardo, B. (2010). Crowdsourcing a real-world on-line query by humming system. Proceedings of the 7th Sound and Music Computing Conference, 2010, Barcelona,

- Spain, Retrieved from
<http://music.eecs.northwestern.edu/publications/smc2010-huq-cartwright-pardo.pdf>
- Kan, M., Wang, Y., Iskandar, D., Nwe, T. L., & Shenoy, A. (2008). LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transaction on Audio, Speech, and Language Processing*, 16(2), 338-349.
<http://dx.doi.org/10.1109/TASL.2007.911559>
- Kline, R. L., & Glinert, E. P. (2003). Approximate matching algorithms for music information retrieval using vocal input. *Proceedings of the Eleventh ACM International Conference on Multimedia 2003*, 130-139. <http://dx.doi.org/10.1145/957013.957042>
- Kornstädt, A. (1998). Themefinder: A web-based melodic search tool. *Computing in Musicology*, 11, 231-236.
- Lee, K., & Slaney, M. (2008). Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 26(2), 291-301. <http://dx.doi.org/10.1109/TASL.2007.914399>
- Lee, Y. J., & Moon, S. B. (2006). A user study on information searching behaviors for designing user-centered query interface of content-based music information retrieval system. *Journal of the Korean Society for Information Management*, 23(2), 5-19.
<http://dx.doi.org/10.3743/KOSIM.2006.23.2.005>
- Lemström, K., & Pienimäki, A. (2007). On comparing edit distance and geometric frameworks in content-based retrieval of symbolically encoded polyphonic music. *Musicae Scientiae, Discussion Forum* 4a, 135-152.
- Lemström, K., & Tarhio, J. (2003). Transposition invariant pattern matching for multi-track strings. *Nordic Journal of Computing*, 10, 185-205.
- McNab, R. J., Smith, L A., Witten I. H., & Cunningham, S. J. (1996). Towards the digital music library: tune retrieval from acoustic input. *Proceedings of the 1st ACM International Conference on Digital Libraries*, 11-18.
- McNab, R. J., Smith, L A., Bainbridge, D., & Witten, I. H. (1997). The New Zealand Digital Library MELOdy inDEX. *D-Lib Magazine*, 3(5), 4-15.
- Melucci, M., & Orio, N. (2004). Combining melody processing and information retrieval techniques: Methodology, evaluation and system implementation. *Journal of the American Society for Information Science and Technology*, 55(12), 1058-1066.
<http://dx.doi.org/10.1002/asi.20058>
- Mesaros, A., & Virtanen, T. (2010). Automatic recognition of lyrics in singing. *EURASIP Journal*

- on Audio, Speech, and Music Processing, 2010:546047.
<http://dx.doi.org/10.1155/2010/546047>
- Nam, G. P., Park, K. R., Park, S., Lee, S., & Kim, M. (2012). A new query-by-humming system based on the score level fusion of two classifiers. International Journal of Communication Systems, 25(6), 717-733. <http://dx.doi.org/10.1002/dac.1187>
- Papadopoulos, H., & Tzanetakis, G. (2012). Modeling chord and key structure with Markov logic. Proceedings of ISMIR 2012, 127-132.
- Pardo, B., Shiffrin, J., & Birmingham, W. (2004). Name that tune: A pilot study in finding a melody from a sung query. Journal of the American Society for Information Science and Technology, 55(4), 283-300. <http://dx.doi.org/10.1002/asi.10373>
- Prechelt, M., & Typke, R. (2001). An interface for melody input. ACM Transactions on Computer-Human Interaction, 6(2), 133-149. <http://dx.doi.org/10.1145/376929.376978>
- Rho, S., Han, B., Hwang, E., & Kim, M. (2008). MUSEMBLE: A novel music retrieval system with automatic voice query transcription and reformulation. The Journal of Systems and Software, 81(7), 1065-1080. <http://dx.doi.org/10.1016/j.jss.2007.05.038>
- Roger, B., Dannenberg, R., & Hu, N. (2004). Understanding search performance in query-by humming systems. Proceedings of ISMIR 2004, 85-89.
- Sheh, A., & Ellis, D. P. (2003). Chord segmentation and recognition using EM-trained hidden Markov models. Proceedings of ISMIR 2003. Retrieved from <http://ismir2003.ismir.net/papers/Sheh.PDF>
- Shiffrin, J., Pardo, B., & Birmingham, W. (2002). HMM-based musical query retrieval. Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, 295-300.
- Tripathy, A., Chhaatre, N., Surendranath, N., & Kalsi, M. (2009). Query by humming system. International Journal of Recent Trends in Engineering, 2(5), 373-379.
- Turetsky, R. J., & Ellis, D. P. W. (2003). Ground truth transcriptions of real music from force-aligned MIDI syntheses. Proceeding of ISMIR 2003, 445-448.
- Typke, R. (2007). Music retrieval based on melodic similarity. Unpublished doctoral dissertation. Universiteit Utrecht. Nederlands.
- Typke, R., Veltkamp, R. C., & Wiering, F. (2004). Searching notated polyphonic music using transportation distances. Proceedings of the 12th Annual ACM International Conference on Multimedia, 128-135.
- Typke, R., Wiering, F., & Veltkamp, R. C. (2005). A survey of music information retrieval systems. Proceedings of ISMIR 2005, 153-160.

- Viro, V. (2011). Peachnote: Music score search and analysis platform. Proceedings of ISMIR 2011, 359-362.
- Wan, C., & Liu, M. (2006). Content-based audio retrieval with relevance feedback. Pattern Recognition Letters, 27(2), 85-92. <http://dx.doi.org/10.1016/j.patrec.2005.07.005>
- Wang, A. (2003). An industrial-strength audio search algorithm. Proceedings of the 4th International Conference on Music Information Retrieval.
<http://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf>
- Wang, A. (2006). The Shazam music recognition service. Communications of the ACM, 49(8), 44-48. <http://dx.doi.org/10.1145/1145287.1145312>
- Wang, C., Li, J., & Shi, S. (2006). N-gram inverted index structures on music data for theme mining and content-based information retrieval. Pattern Recognition Letters, 27(5), 492-503. <http://dx.doi.org/10.1016/j.patrec.2005.09.012>
- Wold, E., Keislar, B. D., & Wheaton, J. (1996). Content-based classification, search, and retrieval of audio. IEEE Multimedia, 3(3), 27-36.
- Zhu, Y., & Shasha, D. (2003). Warping indexes with envelops transforms for query by humming. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD '03), 181-192. <http://dx.doi.org/10.1145/872757.872780>

• 국문 참고문헌에 대한 영문 표기
(English translation of references written in Korean)

- Choi, Yoonjae, & Park, Jong C. (2009). Extracting melodies from piano solo music based on its characteristics. Journal of KISS(C): Computing Practices and Letters, 15(12), 923-927.
- Kim, Moojung, & Nang, Jongho (2011). An automatic melody track selection in MIDI files for query by humming(QBH) application, Proceedings of Conference of the Korea Information Science Society, 38(1B), 405-408.
- Ku, Kyong-I, Lim, Sang-Hyuk, Lee, Jae-Heon, & Kim, Yoo-Sung (2003). A content-based music information retrieval system using theme melody index. Database Research, 19(3), 34-45.
- Park, Mansoo, & Kim, Hoirin (2006). An audio fingerprinting scheme robust to real-noise environments. Telecommunications Review (SK Telecom), 16(3), 435-446.
- Ro, Jung-Soon (2011). Information retrieval: Theory and practice. Daejeon: Geulnuri.
- You, Jinhee, & Park, Sanghyun (2007). An efficient frequent melody indexing method to improve performance of query-by-humming system. Journal of KISS: Database, 34(4), 283-303.

[부록] 분석 대상 시스템의 사이트

(JC's) ABC Tune finder <<http://trillian.mit.edu/~jc/cgi/abc/tunefind>>
DoDoSoSo <<http://www.DoDoSoSo.com/>>
Folktunefinder <<http://www.folktunefinder.com/>>
Kooplet <<http://www.kooplet.com>>
Meldex(New Zealand Digital Library) <<http://www.nzdl.org/musiclib>>
MelodyCatcher <<http://www.melodycatcher.com>>
Midomi <<http://www.midomi.com>>
Musipedia <<http://www.musipedia.org>>
myMoozle <<http://www.mymoozle.com>>
Parfait Olé <<http://www.parfaitole.com>>
Peachnote <<http://www.peachnote.com>>
SID theme search engine <<http://www.linusakesson.net/themes>>
Themefinder <<http://www.themefinder.org>>
Tunebot music search <<http://tunebot.cs.northwestern.edu>>
Tunepal <<http://tunepal.org/tunepal>>