Journal of Korea Multimedia Society Vol. 16, No. 4, April 2013(pp. 453-463)
http://dx.doi.org/10.9717/kmms.2013.16.4.453

오디오 핑거프린트의 비트에러율을 이용한 자동 음악 요약 기법 및 시스템

김민성[†], 박만수[††], 김회린[†††]

요 약

본 논문은 음악의 코러스(chorus) 구간을 자동으로 추출하는 기법 및 시스템에 대하여 다루었다. 코러스 구간을 자동으로 추출하는 음악 요약 기술은 방대한 음악 데이터베이스에서 특정 음악 검색을 용이하게 할 수 있으며, 온라인 스트리밍 서비스에서 샘플 음악을 생성할 때 사용될 수 있다. 이를 구현하기 위해, 기존의 알고리즘들은 2차원 유사도 행렬, 확률모델, 신경망모델, 템포 특징 벡터, 클러스터링 기법 등을 적절히 활용하여 개발되었다. 본 논문에서는 음악의 오디오 핑거프린트를 추출한 후 곡 내의 오디오 핑거프린트 구간 쌍의 비트에러율을 통해 음악 요약을 추출한다. 다만, 음악 검색 솔루션에서 사용된 오디오 핑거프린트가 데이터베이스에 이미 존재할 경우에는 이를 바로 로딩한 후 비트에러율을 계산하여 음악 요약을 추출할 수 있다. 이런 방법은 이미 만들어진 데이터베이스를 변형 없이 그대로 사용할 수 있음으로써 음악 데이터베이스를 활용한 다양한 알고리즘과 솔루션의 가능성을 보여주었다. 또한, 음악의 코러스를 추출하는데 있어서 기존 방식보다 매우 뛰어난 성능을 보임을 알 수 있었다.

Automatic Music Summarization Method by using the Bit Error Rate of the Audio Fingerprint and a System thereof

Minseong Kim[†], Mansoo Park[††], Hoirin Kim[†††]

ABSTRACT

In this paper, we present an effective method and a system for the music summarization which automatically extract the chorus portion of a piece of music. A music summary technology is very useful for browsing a song or generating a sample music for an online music service. To develop the solution, conventional automatic music summarization methods use a 2-dimensional similarity matrix, statistical models, or clustering techniques. But our proposed method extracts the music summary by calculating BER(Bit Error Rate) between audio fingerprint blocks which are extracted from a song. But we could directly use an enormous audio fingerprint database which was already saved for a music retrieval solution. This shows the possibility of developing a various of new algorithms and solutions using the audio fingerprint database. In addition, experiments show that the proposed method captures the chorus of a song more effectively than a conventional method.

Key words: Music Summarization(음악 요약), Music Retrieval(음악 검색), Bit Error rate(비트에러율), Audio Fingerprint(오디오 핑거프린트), Hash Code(해시코드)

※ 교신저자(Corresponding Author) : 김민성, 주소 : 서울특별시 동대문구 회기로 37(청량리동) 국방기술품질원 서울 3팀(130-871), 전화 : 02) 961-1436, FAX : 02) 964-0198, E-mail : minseong.k@dtaq.re.kr
접수일 : 2013년 3월 10일, 수정일 : 2013년 4월 6일
완료일 : 2013년 4월 9일

[†] 정회원, 국방기술품질원
[††] 정회원, 코난테크놀로지
(E-mail : himansoo@gmail.com)
[†††] 정회원, 한국과학기술원
(E-mail : hrkim@ee.kaist.ac.kr)

1. 서 론

현재 디지털 음악 서비스 시장은 계속적인 성장을 거듭하고 있다. 2011년 음반회사의 수익 비중 중 디지털 음원이 차지하는 비율은 중국이 71%, 한국이 53%, 미국이 53%를 차지하였다. 또한 전 세계적으로 디지털 음악 서비스 가입자는 2010년 820만 명에서 2011년 1340만 명으로 65퍼센트나 증가하였다[1]. 이런 현상은 앞으로도 가속화되어 디지털 음악 서비스 가입자 수는 계속 증가할 것으로 예상되고 디지털 음악 데이터베이스 역시 대폭 증가할 것으로 기대된다. 이에 따라, 최근 빅데이터가 이슈가 되는 시점에서 음악 신호처리 분야도 기존에 연구되던 기술 또는 연구 주제를 시대 흐름에 발맞추어 색다른 시각으로 접근할 필요성이 있을 것이다. 특히, 데이터베이스를 활용한 기술은 새로운 서비스를 가능하게 하고 비즈니스 모델을 다양화할 수 있을 것[2]이라는 점에서 텍스트 기반의 기술뿐만 아니라 디지털 음악 분야도 이런 맥락에서 연구될 수 있을 것이다. 이에 본 논문은 음악 신호처리 기술 중 음악의 코러스(chorus)나 사비(sabi)같은 구간을 자동 추출하는 음악 요약 기술을 새로운 방법으로 제안했다.

먼저, 음악 요약 기술이란 코러스나 사비같은 구간이나 그 외의 여러 다양한 구간을 오디오 신호처리 기법을 사용하여 자동으로 추출하는 것으로서 상당히 다양한 용도로 활용될 수 있다. 첫 번째로, 온라인 음악 서비스에서 샘플 음악을 지정할 때 사용될 수 있다. 보통 온라인 음악 서비스에서 제공하는 샘플 음악은 음악의 시작 부분부터 1분 정도 내외를 무료로 들려주는 경우가 많다. 그러나 음악의 첫 부분은 대체로 인트로(intro)로서 곡의 하이라이트 부분인 코러스를 포함하지 않는 경우가 많다. 그러나 청자의 음악 구매 욕구를 자극하기 위해서는 단순히 음악의 시작부터 일정 시간까지를 샘플 음악으로 제공하는 것보다 코러스 같은 음악의 대표적인 부분을 샘플 음악으로 제공하는 것이 효과적이라고 판단할 수 있다. 이 때 음악 요약 기술을 활용하여 방대한 음악 데이터베이스를 처리하여 자동으로 음악의 대표적인 부분을 추출하여 샘플 음악으로 제공할 수 있는 것이다.

두 번째 사용 용도로서 음악 데이터베이스에서 브라우징(browsing)을 손쉽게 하는데도 활용될 수 있다. 개인 사용자가 보유한 방대한 음악 중에서 특정 곡을 검색할 때, 그 곡의 일부분만을 듣고 싶은 경우가 있다. 이 때, 수동으로 위치를 지정하기보다는 음악의 코러스 부분을 자동으로 들을 수 있다면 더욱 효과적인 브라우징이 가능할 수 있다.

이 외에도 자동 추출되어 생성된 음악 요약은 다른 어플리케이션에 사용될 수 있다. 예를 들면, 서로 다른 음악 간의 유사도를 신호처리 기법을 통해 계산하여 자동으로 음악을 추천하는 솔루션을 구현할 때 쓰일 수 있다. 음악 내에서도 다양한 분위기(mood), 템포(tempo)가 존재하므로 음악 내에서 하나의 구간만을 지정하여 음악 간의 유사도를 구하는 것이 정확할 수 있기 때문이다.

이런 다양한 응용성이 있는 자동 음악 요약 기법을 위해 기존에 수행된 방법들은 이차원 유사도 행렬[3], 유사도 행렬과 템포 특징벡터[4], 확률 모델[5], 클러스터링 기법[6]을 적절히 사용하거나 핑거프린트와 에너지 특징(energy feature)등을 조합[7]하여 사용하기도 하였다. 하지만 본 논문에서는 음악의 오디오 핑거프린트를 추출한 후 핑거프린트 블록간의 비트에러율을 계산하여 음악 요약을 추출한다. 이런 방식은 기존의 방식과 달리 기존 음악 검색 솔루션에서 사용된 이미 저장된 방대한 오디오 핑거프린트 데이터베이스를 변형 없이 그대로 활용할 수 있음을 의미한다. 만약 핑거프린트 데이터베이스에 요약하고 싶은 곡이 존재할 경우, 해당 핑거프린트를 로딩한 후 오디오 핑거프린트 블록간의 비트에러율(BER: Bit Error Rate)을 계산하여 음악 요약을 추출하게 된다. 실험결과 제안된 기법은 코러스 부분을 기존 방법보다 매우 잘 찾아냄을 알 수 있었다. 또한 핑거프린트가 데이터베이스에 존재할 경우에는 기존 데이터베이스를 그대로 변형 없이 사용할 수 있다. 이는 곧, 방대한 오디오 핑거프린트 데이터베이스를 사용하는 방법론의 시작으로 다양한 알고리즘과 어플리케이션으로의 적용 가능성을 보여주었다고 생각한다.

2. 음악 검색 및 제안된 요약 시스템

2.1 필립스 음악 검색 시스템

음악 검색 솔루션이란 사용자가 길거리나 집에서 어떤 음악을 들었을 때, 그 곡의 제목이나 관련 정보

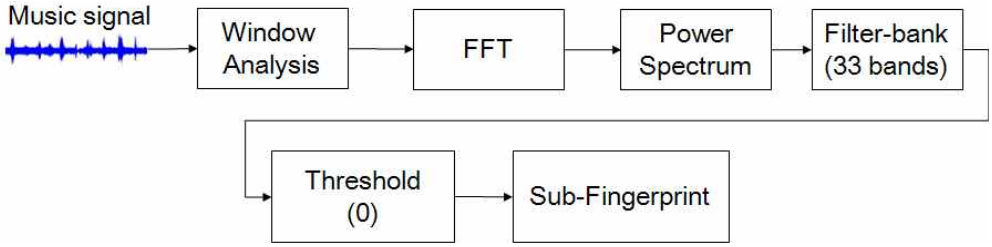

그림 1. 필립스 오디오 핑거프린팅 방식

를 알고자 할 때 사용된다. 예를 들면, 길거리에서 어떤 음악을 접했을 때, 그 음악을 자신의 휴대폰을 통해 일정 시간 녹음하여 서비스 제공 업체에 질의하면 해당 곡의 제목, 작곡가, 관련 부가서비스 등을 사용자에게 제공하게 되는 것이다. 이렇게 음원의 내용을 기반으로 한 검색 방식을 내용 기반 검색(content-based retrieval)이라고 한다[8]. 이런 음악 검색 서비스는 예전에는 통신사를 통해 제공되었으나 지금은 스마트폰의 대중화로 인해 스마트폰 어플리케이션으로도 개발되어 현재 서비스되고 있는 실정이다. 이런 음악 검색 솔루션을 위한 알고리즘에는 여러 종류가 있으나 본 논문에서 사용한 핑거프린트 데이터베이스는 필립스 社가 고안한 방식[9]으로 생성되었던 것을 사용하였다.

필립스 방식은 그림 1에서 잘 나타난 것처럼 시간 영역(time domain)에서의 디지털 음악 신호들을 적절한 윈도우 크기(window size)와 스텝 크기(step size)로 윈도우 분석(window analysis)을 하게 된다. 이렇게 프레임(frame)으로 나눈 후 프레임마다 FFT (Fast Fourier Transform)를 수행하여 파워 스펙트럼(power spectrum)을 구한 후에 주파수 영역(frequency domain)에서 33개의 멜 필터뱅크 에너지(Mel-scale filter-bank energy)값을 구하게 된다. 그리고 이 결과를 식 (1)을 이용하여 해싱(hashing)하여 프레임 당 하나의 32bit 서브-핑거프린트(sub-fingerprint)를 추출한다.

$$E(n,m) = EB(n,m) - EB(n,m+1) - (EB(n-1,m) - EB(n-1,m+1))$$
$$H(n,m) = \begin{cases} 1, & \text{if } E(n,m) > 0 \\ 0, & \text{if } E(n,m) \le 0 \end{cases} \quad (1)$$

식 (1)에서 n은 프레임 인덱스(frame index), m은 주파수 대역의 인덱스, $EB(n,m)$은 n번째 프레임의 m번째 멜 필터뱅크 에너지 값을 의미한다. 이렇게 각 프레임마다 인접 주파수 대역과 인접 프레임 사이의 에너지 차이를 고유한 서브-핑거프린트, 즉 해시코드(hash code)로 추출하게 되는 것이다.[1] 추출된 해시코드 $H(n,m)$은 n번째 프레임의 m번째 이진코드를 의미한다. 식 (1)에서 나타나듯이 $H(n,m)$은 $E(n,m)$의 값이 0보다 크면 1, 0보다 작거나 같으면 0이 된다. 이런 방식으로 전체 음악에 대해서 오디오 핑거프린트가 추출되면 이는 데이터베이스로 저장된다.

핑거프린트 데이터베이스가 저장이 되면 쿼리(query)가 입력되었을 때, 쿼리에 대응하는 음악을 찾아주는 검색 솔루션에 쓰일 수 있게 된다. 검색 솔루션의 검색 구조와 방법은 그림 2에 잘 나타난다. 먼저, 핑거프린트 데이터베이스는 해시 테이블(hash table)을 통해 룩업 테이블(LUT)로서 메모리에 로딩(loading)이 된다. 그리고 쿼리가 입력이 되면 데이터베이스의 핑거프린트를 추출했던 방식과 동일하게 프레임 별로 쿼리의 서브-핑거프린트들을 추출하게 된다. 추출된 핑거프린트, 즉 해시코드들은 검색 후보군(Candidate hash code)이 되어 룩업 테이블을 통해 검색을 하게 된다. 쿼리의 해시코드와 동일한 코드들을 룩업테이블에서 찾아 그 룩업테이블이 가리키는 데이터베이스의 곡의 특정 위치에서 쿼리와 데이터베이스의 핑거프린트 블록 간의 비트에러율을 구하게 된다. 이 때, 최솟값을 가지는 곡을 쿼리에 해당하는 음악으로 간주하게 된다. 단, 비트에러율이 일정 값 이하일 경우에는 바로 정답으로 간주하고 검색을 중지할 수 있다. 이와 같은 필립스 방식의 오디오 핑거프린팅 및 검색방법은 비교적 간단한 구조를 가지고 있으면서도 강력한 성능을 보여 매우 다양

1) 핑거프린트는 어떤 신호로부터 변환되어 기존 신호의 특징 정보를 담고 있다는 뉘앙스가 강하고 해시코드는 변환된 이진코드 그 자체로서의 의미가 강하다. 본 논문에서는 핑거프린트, 해시코드 또는 해시키(hash key)는 결과적으로 모두 동일한 의미로 쓰인다.

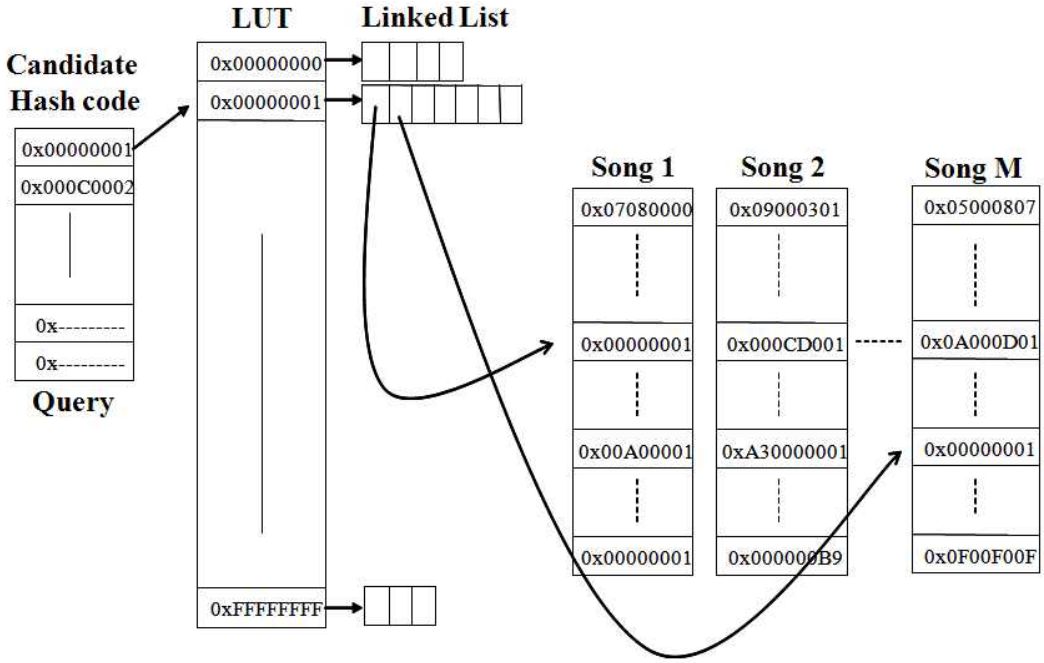

그림 2. 데이터베이스 레이아웃(layout) 구조

한 연구가 활발하게 진행되었다[10].

2.2 제안된 음악 요약 시스템

본 논문에서는 2장 1절에서 설명된 검색 구조를 이용하여 음악 요약을 추출하게 된다. 다만 필립스 검색 시스템에서는 룩업테이블로 로딩되는 검색 데이터베이스가 보유하고 있는 수십만에서 수백만 곡의 전체 오디오 핑거프린트 데이터베이스이지만 요약 시스템에서는 요약 대상이 되는 한 곡의 오디오 핑거프린트들만 로딩이 된다. 그림 3에서 제안된 음악 요약 시스템의 흐름도를 보여준다.

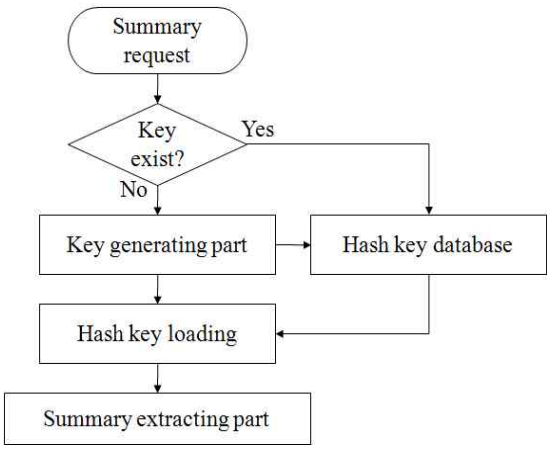

그림 3. 제안된 음악 요약 시스템

먼저, 음악 요약 대상이 되는 곡의 오디오 핑거프린트가 데이터베이스에 존재하면 기 저장된 데이터베이스에서 해당 곡의 해시코드들을 룩업 테이블로 메모리에 로딩하게 된다. 만약, 데이터베이스에 요약할 음악의 핑거프린트가 존재하지 않는다면 요약 대상이 되는 곡의 오디오 핑거프린트를 추출한 후 데이터베이스로 저장하고 이를 다시 룩업 테이블로 로딩하게 된다. 그리고 요약할 대상이 되는 곡 내에서 쿼리들을 지정한 후 검색을 하게 된다.

필립스 검색 시스템에서는 쿼리에서 추출된 해시코드만을 이용하여 검색하지 않고 쿼리의 각 해시코드의 각 비트를 반전하여 검색 대상을 확장할 수 있다. 즉, 해시코드 하나가 32개 bit를 가지고 있으므로 원래 해시코드의 각 비트 한 개씩만을 반전시킨 32개의 해시코드를 추가로 생성하는 것이다. 따라서 쿼리에서 추출된 각각의 해시코드마다 32개의 해시코드들을 추가하여 검색을 하게 된다. 이렇게 원래 해시코드와의 해밍 거리(hamming distance)가 1인 코드들을 검색 후보군으로 추가하여 검색하는 이유는 외부로부터 녹음된 쿼리에는 검색 성능에 악영향을 미칠 수준의 잡음이 혼입될 수 있기 때문이다. 따라서 쿼리의 해시코드들의 각 비트를 한 개씩만 반전하여 생성된 해시코드들을 추가적으로 검색함으로써 검색 정확도를 높일 수 있다. 그러나 제안된 요약 시스

템에서는 쿼리를 외부로부터 녹음하지 않고 원곡에서 지정하여 사용하기 때문에 잡음이 혼입될 여지가 없다. 따라서 잡음요인 때문에 검색 범위를 확장할 필요는 없다. 하지만 음악의 특성에 따라 검색 범위를 확장할 필요가 있을 수 있다. 예를 들면, 라이브 음악 같이 동일 후렴구일지라도 관객들의 함성 소리, 가수의 의도적인 멜로디 변조 등 다양한 요인으로 음향학적인 특성이 크게 달라질 수 있기 때문이다.

3. 제안된 음악 요약 기법

음악 요약은 추출된 음악 요약의 길이 또는 음악 요약이 포함하는 음악 구간의 수에 따라 분류할 수 있다. 음악 요약 길이는 짧게는 10초 내외, 길게는 1분 정도의 길이를 가질 수 있는데, 음악 요약을 어떤 목적으로 사용하는지에 따라 최적 길이는 달라질 수 있다. 또한, 음악 자체의 특성에 따라서도 최적 요약 길이는 달라질 수 있다. 예를 들면, 비교적 복잡한 구조를 지니며 음악의 재생시간이 길고 다양한 템포와 다양한 멜로디를 포함하는 장르나 음악의 경우, 주 멜로디뿐만 아니라 다양한 구간(intro, verse, bridge, sabi, chorus 등)의 일부분씩을 취합하여 음악 요약으로 추출할 수 있을 것이다.

반면에, 구조가 명확하고 청자(listener)의 뇌리에 각인시키기 위한 멜로디가 극명하게 드러나는 팝, 가요, 댄스 등의 음악의 경우는 가장 반복적이고 대표적인 코러스나 사비같은 구간만을 추출하는 것이 효과적일 수 있다. 이 외에도, 음악 요약을 사용자 구매를 촉진하기 위한 샘플 음악으로 사용할 때와 개인 사용자가 검색이나 음악을 인지하고 판별하기 위한 목적으로 사용할 때의 음악 요약의 최적 길이나 특성은 달라질 수 있다.

이렇듯 사용 목적과 음악 특성에 따라 음악 요약의 길이나 형태는 달라질 수 있는데 본 논문에서는 코러스나 사비같은 단일 구간만을 포함하는 단시간 음악 요약(short-time music summary)을 추출하는 것을 목적으로 한다. 그림 4에서 나타나듯이 코러스 구간은 곡 내에서 최소 2번 이상 반복되며 동일 가사, 동일 반주 및 동일 멜로디로 구성되는 경향이 매우 강하다. 이런 특성을 이용하여 곡 내의 오디오 핑거프린트 블록 간의 비트에러율을 계산하여 최고 유사 구간, 즉 최소 비트에러율을 갖는 구간 쌍(pair)을 찾게 된다.

이를 위해 요약 대상이 되는 곡 내에서 여러 개의 쿼리를 차례로 지정한다. 그림 5에서 나타나듯이 고정 길이의 쿼리를 일정 간격으로 지정한 후 지정된 각 쿼리를 통해 메모리에 로딩 된 룩업 테이블을 대상으로 검색을 하게 된다. 그리고 매칭되는 위치에서 쿼리의 핑거프린트 블록과 원곡 해당 위치에서의 오디오 핑거프린트 블록 간의 비트에러율을 계산하게 된다. 그리고 그 중 최솟값을 가지는 구간 쌍을 찾아내어 음악 요약으로 추출하게 되는 것이다. 이를 식으로 정리하면 아래와 같다.

$$N_Q = \frac{L - W}{S} \tag{2}$$

$$H_Q(X, m) = H_S(X+n, m) \tag{3}$$

$$BER(S_Q, F_Q, L_{list}) = \frac{\sum_{n=1}^{N_Q}\sum_{m=1}^{32} |H_Q(S_Q, m) - H_S(L_{list} - F_Q + n, m)|}{32 \times N_Q}, \tag{4}$$

여기서, $0 \le S_Q \le \frac{F_S}{2}$, $0 \le F_Q \le N_Q$,

$\{L_{list} | L_{list} \in H_S(S_Q + F_Q)$의 연결리스트값$\}$,

단, $L_{list} < S_Q - 100$ 또는 $L_{list} > S_Q + N_Q + 100$

$$S_Q^* = ArgMin\, BER(S_Q, F_Q, L_{list}) \tag{5}$$

식 (2)에서, L은 쿼리의 고정 길이(추출된 요약 길이), W는 윈도우 크기, S는 스텝 크기를 의미한다.

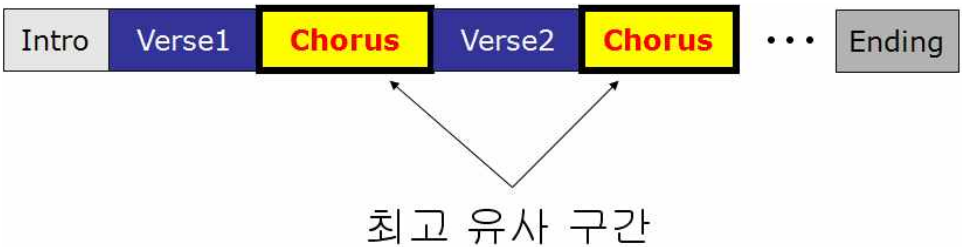

그림 4. 일반적인 대중음악 구조

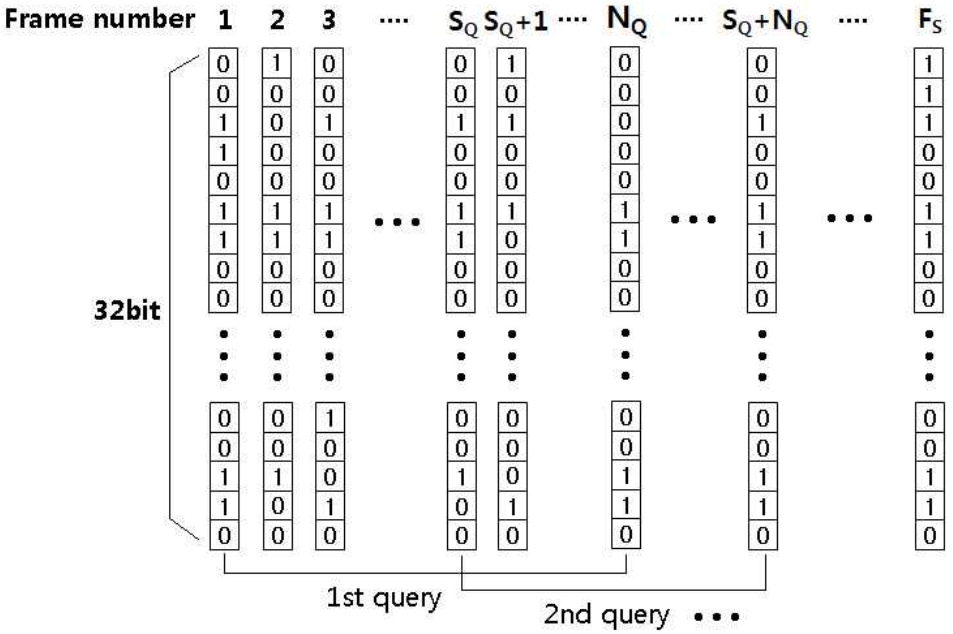

그림 5. 원곡에서 지정되는 쿼리의 위치

N_Q는 쿼리의 총 프레임 수를 의미한다. W와 S는 오디오 핑거프린트 데이터베이스 추출을 위해 윈도우 분석에서 사용했던 값들을 그대로 사용한다. 본 논문에서는 W는 370msec, S는 11.6msec를 사용하였다.

식 (3)은 원곡에서 지정된 쿼리의 해시코드를 정의한다. $H_Q(X,m)$은 원곡의 첫 프레임에서부터 X만큼 떨어진 쿼리의 해시코드를 의미하고, 이는 원곡의 X+n번째 해시코드인 $H_S(X+n, m)$과 동일하다.

식 (4)는 쿼리와 원곡의 해시코드 블록 간의 비트 에러율을 계산하는 식이다. 여기서 F_Q는 쿼리의 프레임 번호를 의미하고 F_S는 원곡의 총 프레임 수를 의미한다. S_Q는 쿼리의 첫 번째 프레임 위치를 원곡에서 지정할 때, 원곡 첫 프레임부터 쉬프트(shift)시킨 프레임 수를 의미한다. S_Q는 0에서부터 원 곡의 총 프레임 수의 반 정도까지 증가하도록 설정하였다. 이는 후렴구는 대체로 음악 내에서 최소한 재생시간의 반 이전에는 출현한다는 음악 배경 지식을 활용하여 쿼리 지정 범위를 제한하였다.

그리고 동일 구간의 비교를 방지하기 위해 쿼리의 원곡에서의 실제 구간에서 100프레임 전후로 안전마진(safety margin)을 두어 검색 대상에서 제외하였다. 이렇게 안전마진을 두지 않게 되면 쿼리로 지정된 원곡의 구간과 근소하게 떨어진 부분을 요약 결과로 추출하게 된다. 그 이유는 쿼리로 지정된 부분에서 근소하게 떨어진 부분은 프레임 당 11.6msec밖에 차이가 나지 않아 거의 동일한 해시코드들로 이루어지기 때문이다. 우리가 찾고자 하는 것은 쿼리로 지정된 구간에서 어느 정도 멀리 떨어져서 반복되는 소절을 찾고자 함이기 때문에 100프레임 정도의 여유 있는 안전마진을 두고 후렴구를 찾고자 하는 것이다. 스텝 사이즈가 11.6msec이므로 100프레임으로 안전마진을 두게 되면, 원곡에서 쿼리로 지정된 첫 프레임부터 약 1초 이상 전(before)이 되는 시점부터, 쿼리가 지정된 첫 프레임에 쿼리의 길이를 더하고 다시 안전마진 1초 이상을 더한 시점까지를 검색 대상에서 제외하는 것이다. 실험적으로 100프레임 정도의 안전마진을 두었을 때 쿼리와 매우 근접한 구간을 음악 요약으로 추출하는 결과는 발생하지 않았다. 이런 개념을 식으로 표현한 것이 식 (4)에서 L_{list}의 범위를 제한한 것이다. L_{list}는 원곡의 (S_Q+F_Q)번째 해시코드의 룩업테이블의 연결리스트 값들로서 이 해시코드와 동일한 해시코드를 가지는 원곡의 프레임 번호들을 의미한다. 이를 예를 들어 설명하면 다음과 같다.

먼저, S_Q가 200, F_Q가 3, N_Q가 3000이라고 하고, 이때의 해시코드가 0101(편의 상 4비트로 설명함)이고 이와 동일한 해시코드가 원곡에서 3번째 프레임, 205번째 프레임, 3280번째 프레임, 4553번째 프레임에서 출현하게 되면 S_Q가 200, F_Q가 3일 때의 L_{list}값들은 3, 205, 3280, 4553이 된다. 그런데 구간의 제한

을 안전마진과 함께 두게 됨으로써, 원곡에서 205번째와 3280번째 프레임의 해시코드가 쿼리의 해시코드와 일치하더라도 이 구간과는 비트에러율을 계산하지 않게 된다. 결과적으로 원곡의 3번째 프레임과 4553번째 프레임에 해당하는 블록 간의 비트에러율만 계산하게 된다. 식 (4)에서 나타나듯이, 비트에러율을 계산할 때는 쿼리의 위치가 F_Q가 3이고 매칭된 원곡의 프레임 번호가 3일 경우, 원곡의 첫 번째 프레임의 해시코드와 쿼리의 첫 번째 해시코드와 해밍 거리를 구하게 된다. 그리고 원곡과 쿼리 모두 프레임 번호를 1씩 늘려가면서 서로간의 해밍 거리를 구하여 더해가고 해밍 거리의 총합을 32와 쿼리의 프레임 수 N_Q로 나눔으로써 비트에러율이 계산된다.

이렇게 원곡에서 지정된 쿼리의 오디오 핑거프린트 블록과 원곡의 핑거프린트 블록 간의 비트에러율을 지정된 S_Q의 범위에 따라 구하고, 이 비트에러율이 최솟값이 될 때의 S_Q값을 식 (5)와 같이 찾는다. 그리고 이 S_Q에 해당하는 시점부터 요약길이 L초를 더한 시점까지를 음악 요약으로 추출하게 된다.

지금까지 설명한 제안된 음악 요약 기법 및 시스템을 기존 필립스 검색 시스템과 비교하면 표 1과 같이 정리할 수 있다. 필립스 음악 검색 시스템은 기본적으로 음악 곡명을 찾는 알고리즘이다. 따라서 쿼리는 주로 길거리나 집 등에서 녹음 장치를 통해 녹음이 되는 경우가 많다. 반면 제안된 요약시스템에서는 쿼리가 원곡의 디지털 파일 내부에서 바로 추출되거나 오디오 핑거프린트 데이터베이스에서 바로 지정된다. 따라서 잡음이 혼입될 여지가 없다. 다만 반복성이 떨어지는 곡의 경우 검색범위를 해밍거리 1인 해시코드까지 넓혀 사용할 수 있다. 또한, 필립스 검색 시스템과 달리 제안된 음악 요약 시스템에서는 안전마진이라는 개념을 도입하여 원곡에서 지정된 쿼리의 원래 위치를 정답으로 찾아내는 오류를 없앨 수 있었다. 그리고 필립스 검색 시스템의 경우, 특정 임계값을 설정하여 모든 검색이 완료되지 않더라도 임계값 이하의 비트에러율에 도달했을 경우 바로 검색을 중지할 수 있다. 그러나 제안된 요약 검색 시스템에서는 주어진 검색 범위 내에서 곡 내의 핑거프린트 블록 쌍의 비트에러율을 모두 구하고 그 중 최솟값을 찾는다. 마지막으로, 제안된 요약 시스템의 핑거프린팅 방식은 필립스 방식에 국한되지 않고 기 저장된 핑거프린트 데이터베이스에 적용된 핑거프린팅 방식을 따르게 된다. 본 논문에서는 필립스 방식으로 해싱된 핑거프린트를 대상으로 요약기법을 사용하였지만 핑거프린트 추출 공식, 핑거프린트 비트 수 등이 차이가 나더라도 제안된 요약기법을 적용할 수 있다.

본 논문에서는 기존에 널리 사용되는 음악 검색 시스템과 데이터베이스를 활용하여 음악 요약 기법으로 사용하는 방법에 대하여 다루었다. 검색 시스템은 이미 오래전부터 존재하였고 이를 위한 수십만에서 수백만곡의 핑거프린트 데이터베이스 역시 존재하고 있다는 점에 착안하여, 기존 알고리즘을 색다른 목적으로 구현한 것이다. 이는 핑거프린트 데이터베이스를 활용하는 다양한 알고리즘의 가능성을 보여주었을 뿐만 아니라 요약 성능 또한 매우 좋음을 실험을 통해 알 수 있었다.

표 1. 기존 검색 시스템과 제안된 요약 시스템의 비교

시스템 / 비교항목	필립스 음악 검색 시스템	제안된 음악 요약 시스템
사용 목적	음악 검색 (쿼리에 해당하는 곡명)	특정 곡의 코러스 부분 추출
쿼리 획득 방법	녹음장치 등에 의한 녹음	원곡 디지털 파일 또는 핑거프린트 DB
검색 대상	전체 핑거프린트 데이터베이스	안전마진을 제외한 요약할 곡의 핑거프린트
검색 중지	비트에러율의 임계값 지정 가능	임계값 없음
핑거프린팅 방식	지정된 공식 사용	기 존재하는 핑거프린트 DB의 핑거프린팅 방식에 따라 적용함

4. 실험 및 분석

4.1 실험 방법

자동 음악 요약 성능을 평가하기 위해서 다양한 평가 방법이 존재하였다. 먼저, 사전에 음악의 대표적인 부분을 핸드마킹(hand-marking)하여 구간을 미리 지정한 후 알고리즘을 통해 도출된 요약 구간을 비교하는 방법이 있다. 또 다른 방법으로는 자동 추출된 음악 요약 구간을 음악 전문가 또는 일반 청자들이 듣고 점수를 주는 형태이다. 점수의 형태는 일정한 범위를 갖는 임의의 수치가 될 수도 있지만 Good/Average/Poor[11] 또는 Voiced Title(노래 제목이 들어 간 코러스 부분)/Repeating Voiced Words (반복되는 가사 부분)/Repeating(그 외 반복되는 부분)/Not Repeating and Other(반복되지 않는 부분) 등[7]으로 구분하여 평가자들이 그 중에서 선택할 수 있도록 하는 방법이 존재하였다. 또한, 음악 요약 결과의 정확성 측면뿐만 아니라 알고리즘의 처리시간(processing time)도 성능 평가 기준으로 사용되기도 하였으며 다양한 분위기의 구간을 얼마나 많이 포함하는지도 평가 기준으로 사용되기도 하였다[4].

본 논문에서는 사전에 음악의 코러스와 사비 부분 등을 파악하여 지정한 후, 자동 추출된 음악 요약이 Voiced Title/코러스/사비일 경우 5점, 기타 반복되는 구간(verse 등)이면 3점, 그 외에 반복되지 않는 구간(intro, bridge등)이면 0점을 각각의 음악 요약 결과마다 점수를 주었다. 그리고 전체 점수의 산술평균을 평가지표로 사용하였다. 여기서 Voiced Title, 코러스, 사비 등은 대체로 동일한 구간을 의미하나 곡에 따라서 약간 씩 달라질 수가 있다. 먼저, Voiced Title은 노래 제목이 가사로 나타나는 소절을 말한다. 예를 들면, f(x)의 'Hot summer', 카라의 'Jumping', 싸이의 '강남스타일' 등의 곡들은 모두 노래 제목이 가사로서 후렴구에 나타나게 된다. 코러스나 사비는 음악 내에서 대체적으로 제일 많이 반복되는 구간으로 대중음악에서 가장 매력적인 구간으로 볼 수 있다. 둘의 차이는 거의 없다고 볼 수 있으나 전자는 음악의 구조적인 측면에서 반복성이 강한 동일 부분들을 의미하는 뉘앙스가 강하고 후자의 경우는 청자(listener)의 기억에 오래 남는 대표적 부분으로써의 뉘앙스가 강하다고 볼 수 있다. 요약하자면, 노래 제목이 가사로서 곡의 하이라이트 부분에 나타나면 Voiced Title이라고 보고, 노래 제목이 가사에 나오지 않을 때는 그 곡의 코러스나 사비부분을 음악 요약의 최적 구간으로 보는 것이다.

4.2 실험 결과 및 분석

실험에 쓰인 음악은 가요, 팝, 록 등이 포함된 1000여곡 중에서 랜덤하게 100곡을 선별하였다. 모든 곡들은 MP3파일로서 16bit, 16khz, mono format 웨이브(wav) 파일로 변환되어 처리되었다. 또한, 성능 비교를 위해 기존에 개발된 VQ를 이용한 요약 기법(VQ)[6]과 비교 평가를 하였다. 제안된 기법과 비교된 VQ 기법은 유사도 행렬과 템포 특징을 조합하여 이용한 기법[4], HMM을 이용한 기법[5]과 비교평가에서 비슷하거나 우수한 성능을 보이는 것으로 나타났다[6]. 단, 필립스 검색 방식은 변형 없이 사용할 경우 요약 추출 기능을 전혀 발휘할 수 없기 때문에 비교 대상에서 제외하였다. 실험 결과는 표 2에 나타나듯이 비트에러율을 이용한 요약 기법(BER)이 기존 방식보다 뛰어난 성능을 보임을 알 수 있었고 절대적 성능 또한 상당히 우수함을 알 수 있었다.

제안된 기법의 경우, 요약 길이를 20초로 설정했을 때 100곡 중 87곡의 코러스 파트를 정확히 추출하였다. 특히, 가요나 팝 장르에서 성능이 매우 뛰어남을 알 수 있었다. 이렇게 가요나 팝 같은 장르에서 성능이 뛰어난 이유는 대중들에게 기억시킬 자극적인 동일한 후렴구를 여러 번 반복하여 작곡하는 상당수의 대중가요나 팝 등의 작법과 직접적인 연관이 있을 것이다. 하지만 그런 반복적인 구조를 가진 음악과는 달리 반복성이 떨어지고 상대적으로 복잡한 구조를 지닌 음악 등은 가요나 팝 같은 장르보다 요약 성능이 떨어짐을 알 수 있었다. 그런 장르의 음악은 단일 코러스 또는 단일 구간을 추출하기보다는 다양한 구간들을 추출하여 취합하는 것이 더 적합할 수 있을 것이다. 이를 위해 비트에러율을 여러 단계(multi-stage)로 계산하여 추출하는 단계를 추가로

표 2. 성능 평가(가요, 팝, 록), 해밍 거리는 0

Summary Length (sec)	Average Score	
	VQ	BER
10	3.78	4.22
15	3.92	4.43
20	4.08	4.68

고안해 볼 수도 있을 것이다.

또 다른 실험을 위해 공연 실황을 녹음한 라이브(live) 음악 50곡을 선별하여 동일한 방식으로 비교 평가를 하였다. 라이브 음악의 경우 동일 후렴구일지라도 관객들의 함성, 반주, 음색, 음량, 멜로디, 전조(modulation) 등의 다양한 변화요인들로 인해 코러스나 사비 등의 구간이 곡 내에서 음향학적으로 동일하지 않을 가능성이 높음을 쉽게 예상할 수 있다. 이런 연유로 2장 2절에서 설명한대로 원곡에서 추출된 쿼리의 해시코드와의 해밍 거리가 1인 해시코드도 추가로 검색하여 실험한 결과 표 3과 같은 결과를 얻을 수 있었다. 그러나 이렇게 검색 영역을 확장하였음에도 불구하고 스튜디오에서 녹음된 음악들에 비해서 요약 결과 성능이 떨어짐을 알 수 있었다. 이런 문제를 해결하기 위해서는 단순히 검색영역을 무한정 계속해서 확장하기보다는 핑거프린팅 방식의 변화를 주는 것도 고려해 볼 수 있을 것이다. 이를 위해서는 청각 인지 특성을 고려한 음향 레벨의 특징 추출 패턴에서 벗어나 인간이 인지하는 음악 레벨의 정보를 담고 있는 핑거프린팅 추출 기법도 연구주제가 될 수 있을 것이다. 여기서 말하는 음향 레벨이란 주파수 영역에서의 에너지의 크기에 치중하는 차원을 말하고, 음악 레벨은 파워 스펙트럼을 통해 음악의 조성(tonality), 멜로디, 템포(tempo), 코드(chord) 등을 추출하는 것을 의미한다. 실제로 이런 음악 레벨의 정보를 디지털 음악 신호를 통해 추출하려는 연구는 오래 지속되어왔다. 다만, 이런 특징 추출 기법을 이용한 핑거프린팅 방식은 검색 성능과 요약 성능을 모두 만족시킬 수 있어야 할 것이다.

표 3. 성능 평가(라이브 음악), 해밍 거리는 0,1 모두 사용

Summary Length (sec)	Average Score	
	VQ	BER
10	2.76	3.11
15	2.95	3.53
20	3.36	3.93

5. 결 론

본 논문에서는 음악의 코러스 부분을 자동 추출하는 기법 및 시스템을 제안하였다. 기존에 이차원 유사도 행렬, 확률모델, 오디오 핑거프린트, 템포(tempo) 특징, 머신러닝, 클러스터링 기법 등을 이용하는 방법과 달리 본 논문에서는 비트에러율을 이용함으로써 간단한 스킴(scheme)을 가지면서도 우수한 성능을 보여줌을 알 수 있었다.

특히, 제안된 기법은 가요나 팝 등 반복성이 명료한 구조의 음악에서 기존 기법보다 매우 뛰어난 성능을 보임을 알 수 있었다. 다만, 추후에는 더욱 장르를 세분화하여 각 장르마다 1000여곡 이상의 음원을 확보하여 더 많은 곡들을 실험할 필요가 있을 것이다. 또한, 단일 구간 추출에서 그치지 않고 단계별 비트 에러율 계산을 통해 여러 구간을 추출하는 방식 등도 추후 연구 대상에 포함될 수 있을 것이다.

제안된 자동 음악 요약 기법과 시스템은 핑거프린트 데이터베이스가 존재할 경우에는, 특징추출(feature extraction) 단계나 클러스터링, 머신 러닝 등의 단계 없이도 검색 솔루션에 널리 이용되는 핑거프린트 데이터베이스를 변형 없이 그대로 사용함으로써 다양한 어플리케이션으로의 적용 가능성을 보여주었다. 연구 초기에는 특징추출단계를 제거할 수 있음으로써 처리시간을 줄일 수 있다는 생각이 주된 동기가 되었으나 연구를 진행하면서 처리시간보다는 이미 존재하는 데이터베이스를 변형 없이 사용하여 다양한 솔루션을 기획해 볼 수 있겠다는 생각이 강해졌다. 물론, 제안된 기법은 처리시간 면에서도 강점이 있지만 점점 더 빠른 속도로 컴퓨팅 파워와 스토리지(storage)가 증대되는 시점에서 처리시간의 감소나 제거 측면보다는 음악 데이터베이스만을 이용한 기술과 어플리케이션을 통해 cold data[12]의 활용가능성을 보여주었다는 점에서 연구의 의미를 주고 싶다. cold data란 특정기간동안 거의 질의되지 않고 저장되어 있는 데이터를 의미하는데, 음악 데이터베이스나 오디오 핑거프린트 데이터베이스에서도 이런 개념이 적용될 수 있을 것이다. 사용자의 쿼리나 요청이 없을 때에도 이런 데이터들을 이용해 시스템의 백그라운드(back ground)에서 분석 기술 솔루션을 통해 사용자에게 능동적인 서비스를 제공하는 것을 고려해 볼 수 있을 것이다.

본 논문에서는 단지 한 곡의 핑거프린트 데이터베이스만을 이용하였지만 추후 방대한 핑거프린트 데이터베이스를 모두 또는 일부 이용하는 다른 어플리케이션을 고려할 수 있을 것이다. 이렇게 동일 데이터베이스를 다양하게 이용하는 원 소스 멀티 유즈

(One Source Multi-Use) 차원의 연구뿐만 아니라 이미 언급했듯이 다양한 솔루션에 공통적으로 최적의 성능을 얻을 수 있는 핑거프린팅 기법 및 오디오 신호처리 기법 역시 추후 연구 대상이 될 수 있을 것이다.

참 고 문 헌

[1] Placido Domingo, *Digital Music Report 2012*, International Federation of the Phonographic Industry (IFPI), 2012.

[2] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers, *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, McKinsey & Company, New York, 2011.

[3] Matthew Cooper and Jonathan Foote, "Automatic Music Summarization via Similarity Analysis," *Proc. IRCAM*, pp. 81-85, 2002.

[4] Sangho Kim, Sungtak Kim, Suk-bong Kwon, and Hoirin Kim, "A Music Summarization Scheme using Tempo Tracking and Two Stage Clustering," *IEEE workshop on Multimedia Signal Processing 2006*, Vol. 1. pp. 225-228, 2006.

[5] Geoffroy Peeters, Amaury La Burthe, and Xavier Rodet, "Toward Automatic Music Audio Summary Generation from Signal Analysis," *Proc. Int. Conf. Music Information Retrieval*, pp. 94-100, 2002.

[6] Sangho Kim, Sugntak Kim, and Horin Kim, "Automatic Music Summarization using Vector Quantization and Segment Similarity," *The Journal of the Acoustical Society of Korea*, Vol. 27, No. 2E, pp. 51-56, 2008.

[7] C. Burges, D. Plastina, J. Platt, E. Renshaw, and H. Malvar, "Using Audio Fingerprinting for Duplicate Detection and Thumbnail Generation," *Proce. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 9-12, 2005.

[8] 허태관, 조황원, 남기표, 이재현, 이석필, 박성주, 박강령, "내용 기반 음원 검출 시스템 구현에 관한 연구," 멀티미디어학회논문지, 제12권, 제11호, pp. 1581-1592, 2009.

[9] J.A. Haitsma and T.Kalker, "A Highly Robust Audio Fingerprinting System," *Proc. of ISMIR 2002*, pp. 144-148, 2002.

[10] Mansoo Park, Hoi-Rin Kim, and Seung Hyun Yang, "Frequency-Temporal Filtering for a Robust Audio Fingerprinting Scheme in Real-Noise Environments," *ETRI Journal*, Vol. 28, No. 4, pp. 509-512, 2006.

[11] B. Logan and S. Chu, "Music Summarization using Key Phrases," *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing*, pp. 749-752, 2000.

[12] Is Your Big Data Hot, Warm, or Cold?, http://ibmdatamag.com/2012/06/is-your-big-data-hot-warm-or-cold, 2012.

김 민 성

2002년 세종대학교 전자공학과 학사
2007년 한국과학기술원 전자공학과 석사
2002년～2004년 인켈
2007년～2008년 삼성전자
2010년～2011년 한국특허정보원
2012년～현재 국방기술품질원, 연구원
관심분야 : 디지털 음악 신호처리

박 만 수

2000년 인하대학교 정보통신공학과 학사
2002년 한국과학기술원 전자공학과 석사
2006년 한국과학기술원 전자공학과 박사
2003년～현재 코난테크놀로지, 팀장
관심분야 : 내용기반 멀티미디어 정보처리 및 검색

김 회 린

1984년 한양대학교 전자공학과 학사
1987년 한국과학기술원 전자공학과 석사
1992년 한국과학기술원 전자공학과 박사
1987년～1999년 ETRI 선임연구원
1994년～1995년 ATR-ITL 방문연구원
2006년～2007년 UCSD 방문교수
2000년～현재 한국과학기술원 전자공학과 부교수
관심분야 : 음성인식, 화자인식, 음향코딩, 음향정보 인덱싱