

# 음악의 특성에 따른 피아노 솔로 음악으로 부터의 멜로디 추출

(Extracting Melodies from Piano  
Solo Music Based on its  
Characteristics)

최윤재<sup>†</sup>      박종철<sup>\*\*</sup>  
(Yoonjae Choi)      (Jong C. Park)

**요약** 최근 디지털 음반 시장의 발전으로 인해 음악 검색 및 추천에 대한 수요가 증가하고 있는데 이러한 서비스를 수행하는 음악 기반 응용 시스템의 성능 향상을 위해서는 일반적인 음악의 형태인 다음(Polyphonic) 음악에서 멜로디를 추출하는 과정이 필수적이다. 본 논문에서는 다음의 복잡도가 높고 넓은 음역을 가지는 음악을 만들 수 있는 피아노 솔로 음악에서 멜로디를 추출하는 방법을 제안한다. 본 연구는 피아노 음악을 음악의 특성에 따라 세 가지 유형으로 분류해서 유형별로 멜로디를 추출하는 방법을 살펴본다. 제안한 방법에 따라 구현된 시스템을 이용해서 성능을 측정한 결과 다양한 피아노 솔로 음악에 적용 가능함을 확인했다.

**키워드** : 멜로디 추출, 피아노 음악, 다음 음악

**Abstract** The recent growth of a digital music market induces increasing demands for music searching and recommendation services. In order to improve the performance of music-based application services, the process of extracting melodies from polyphonic music is

· 이 연구(논문)는 산업자원부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었습니다.

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '음악의 특성에 따른 피아노 솔로 음악으로부터의 멜로디 추출의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 비회원 : 한국과학기술원 전산학과  
mp2893@gmail.com

<sup>\*\*</sup> 종신회원 : 한국과학기술원 전산학과 교수  
park@nlp.kaist.ac.kr

논문접수 : 2009년 8월 13일

심사완료 : 2009년 10월 14일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제12호(2009.12)

essential. In this paper, we propose a method to extract melodies from piano solo music which is highly polyphonic and has a wide pitch range. We categorize piano music into three classes taking into account the characteristics of music, and extract melodies according to each class. The performance evaluation for the implemented system showed that our method works successfully on a variety of piano solo music.

**Key words** : Melody Extraction, Piano Music, Polyphonic Music

## 1. 서론

인터넷을 기반으로 하는 정보기술과 검색기술이 발전하면서 최근에는 기존의 텍스트 형태의 정보뿐만 아니라 사진이나 음악, 동영상과 같은 멀티미디어 자료의 검색 및 활용 방법에 대한 연구도 활발히 진행되고 있다. 특히 디지털 음반 시장이 빠른 속도로 성장해 가면서[1] 음악 검색 및 추천에 대한 수요도 증가하고 있다. 음악을 검색하기 위해서는 음악의 제목이나 작곡가 등의 음악 정보를 자연언어의 형태로 질의하거나 허밍 등을 통해 멜로디를 기반으로 질의를 할 수 있다. 음악 정보를 이용하여 질의를 하는 경우, 제목이나 작곡가 등 검색에서 활용할 수 있는 형태의 정보가 사전에 모든 음악에 기록되어 있어야 한다. 이러한 음악 정보를 반복적인 작업 없이 효율적으로 제공하기 위해서는 검색하고자 하는 음악 데이터 베이스 내에서 동일한 음악을 자동으로 분류할 수 있어야 하고, 이 작업은 허밍을 통해 동일한 멜로디를 가지는 음악을 검색하는 것과 같은 문제로 생각할 수 있다. 일반적인 다음(Polyphonic) 음악에서는 그 음악의 특징을 나타내는 멜로디 이외에도 반주(Accompaniment) 등의 정보가 함께 나타나므로 음악 정보 기반의 질의 시스템이나 허밍 기반 질의 시스템에서 효율적인 검색을 위해서는 각 음악이 가지는 특징적인 멜로디를 추출하는 것이 관련 시스템의 성능 향상에 필수적이다[2].

본 연구에서는 음악에서 멜로디를 추출하는 방법에 대해 논의하며 그 대상으로 피아노 솔로 음악을 고려한다. 음악을 만들 수 있는 여러 악기들 중 피아노는 가장 넓은 음역을 가지며, 복잡한 다음 음악도 단일 악기만으로 표현할 수 있다. 또한 오케스트라가 연주하는 교향곡을 피아노로 편곡할 수 있을 만큼 피아노는 표현의 정도가 크다. 따라서 피아노의 음악에서 멜로디를 추출할 수 있다면 그 방법을 다른 음악에도 적용해서 멜로디를 추출하는 데 큰 도움이 되리라 기대한다.

## 2. 관련연구

음악에서 멜로디를 추출하는 방법은 두 가지로 나눌

수 있다. 먼저 MP3나 Wave 형식의 파일로부터 오디오 정보를 받아들여 신호처리 과정을 통해 멜로디를 추출하는 방법이 있다. 이 방법의 대표적인 예로는 Goto[3]와 Ryyananen, Klapuri[4]가 있다. 이러한 방식은 실제로 녹음된 음악에서 멜로디를 추출할 수 있다는 장점이 있지만 음의 높이나 음의 지속 시간, 음들 간의 관계, 음들의 배치와 같은 음악 자체의 특성을 효과적으로 사용하기 어렵다는 단점이 있다.

다른 방법으로는 미디(MIDI)나 NIFF, ESac와 같이 음악을 상징적인 기호로 표현한 파일을 받아들여서 멜로디를 추출하는 것이다. 이러한 방향으로는 여러 종류의 시도가 있었지만 현재까지 Uitdenbogerd와 Zobel[2]의 Skyline Algorithm, 또는 이의 변형된 알고리즘이 가장 성공적이다[5].

본 연구에서는 멜로디의 효율적인 추출을 위해서는 음악의 특성을 우선적으로 고려하는 것이 중요하다고 보아 음표로 표현된 음악으로부터 멜로디를 추출하는 방법에 대해 논의한다. 특히 본 연구에서는 음악의 특성에 따라 음악을 세 가지 형태로 분류하여 각 유형에 맞는 원칙을 사용하여 멜로디를 추출하는 방식을 제안한다. 이와 같은 방법은 Ozacn, Isikhan, Alpkokak [6]에서 실험 결과, 음악의 특성이나 스타일이 멜로디 추출에 큰 영향을 준다는 점을 인식하고 음악의 특성에 따라서 다른 알고리즘을 적용해야 함을 지적한 것과 그 맥을 같이 한다.

### 3. 자료분석

본 논문은 피아노 솔로 음악의 구조적 특징을 고려해서 아래의 세 가지 유형으로 나누는 방식을 제안한다.

1. 동시에 발생하는 음들의 지속 시간이 모두 동일하고 그 외에 음들 간의 겹침이 없는 경우(Equal Occurrence: E 유형)
2. 멜로디가 상단부에 위치하는 경우(High Melody: H 유형)
3. 멜로디가 하단부에 위치하는 경우(Low Melody: L 유형)

#### 3.1 E 유형

E 유형의 음악은 반주와 멜로디가 확연히 구분되지 않고 전체가 화음으로만 구성된 구조다. 동시에 시작된 음들은 대개 동시에 끝나고 다음에 발생하는 음들과는 겹침이 없기 때문에 음들 간의 수직적인 겹침은 존재해도 수평적인 겹침은 존재하지 않는다.

그림 1을 보면 첫 부분의 못갖춘마디의 4분음표 세 개는 모두 동시에 시작하고 동시에 끝난다. 그래서 다음 마디의 2분음표들과 겹치지 않는다. 즉 첫째 마디의 음들은 수직적인 겹침만을 가진다고 할 수 있다. 둘째 마디의 음들도 마찬가지이다.



그림 1 베토벤의 월광 소나타 2악장

#### 3.2 H 유형

기본적인 피아노 연주법에서 오른손은 멜로디를, 왼손은 반주를 담당한다[7]. 그러므로 일반적인 피아노 음악은 대부분 H 유형에 속하고 그만큼 H 유형에는 많은 패턴이 존재한다.

그림 2, 3, 4 세 작품 모두 멜로디는 상단부에 존재한다. 그러므로 하단의 오선지에 있는 음표들은 모두 반주에 해당한다. 하지만 네 작품에서 그 이외의 공통점은 찾기 힘들다. 그림 2에서는 반주음들이 모두 8분음표로 동일한 지속 시간을 갖고 있다.

그림 3의 경우 하단부의 반주가 그 형식이 매우 자유로워서 일정한 패턴을 찾기 힘들다. 그림 4의 경우 위의 두 음악과는 또 다른 구조를 가진다. 그림 4의 상단부에는 16분 음표가 반복적으로 나오고 있다. 그림 2와 3은 상단 오선지의 음들이 단음이거나 화음으로 이루어진 구조인데 그림 4는 상단에도 제 2 반주가 존재하는 구조를 가지고 있다.



그림 2 베토벤의 비창 소나타 3악장



그림 3 Fly Me to the Moon



그림 4 베토벤의 비창 소나타 2악장

#### 3.3 L 유형

음악의 멜로디가 하단에 존재하는 경우는 흔하지 않다. 피아노 음악의 많은 경우는 대개 H 유형에 속하기 때문이다. 상단의 음들이 일정하고 하단의 음들이 더 흥미로운 패턴을 이루고 있을 때 하단의 음들이 멜로디로 인식된다[8]. 그러므로 L 유형의 음악에서 주목해야 할 특징은 바로 상단의 음들이 일정하다는 것이다.

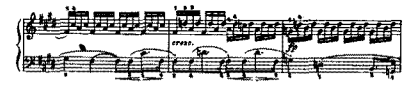


그림 5 베토벤의 월광 소나타 3악장

그림 5는 상단부의 음들이 반주에 해당하고 하단부의 음들이 멜로디를 구성한다. 이러한 L 유형의 특징은 상단부의 반주음들의 지속 시간이 전부는 아닐지라도 매우 일정하다는 것이다. 또 많은 경우 반주음들의 진행 속도가 멜로디음들의 진행 속도보다 빠르다. 이 역시 그림 5를 통해 확인할 수 있다.

#### 4. 유형 자동 판별과 멜로디 추출

음악에서 멜로디를 추출하기 위해서는 우선 그 음악이 3절에서 살펴본 세 가지 유형 중에서 어디에 속하는지를 판단한 뒤 이에 맞게 멜로디를 추출해야 한다. 이를 위해 음악을 미디 파일의 형태로 입력 받는데 우선 여러 트랙 및 채널에 분산되어있는 음들을 모두 하나의 트랙과 채널로 모은 후에 마디를 단위로 나눠서 처리하도록 한다. 각각 다른 마디에 속한 두 음이 붙임줄로 이어져 있는 경우 두 마디를 동시에 처리하도록 한다.

4절에서는 Uitdenbogerd, Zobel[2]이나 Ozcan, Isikhan, Alpkocak[6] 등의 제안과는 다르게 음들 사이의 다양한 겹침에 주목하여 수식을 통해 음악의 특징을 표현한다.

##### 4.1 E 유형의 판별 및 멜로디 추출

E 유형의 음악에서는 다른 유형의 음악에서 보기 어려운 수직적인 겹침이 수평적인 겹침보다 자주 나타나는 특징을 가지고 있다. 따라서 이러한 E 유형의 특징을 아래와 같은 수식으로 나타낼 수 있다.

$$\text{수직적 겹침 비율} = \frac{\text{수직적 겹침 수}}{\text{전체음의 수}}$$

$$\text{수평적 겹침 비율} = \frac{\text{수평적 겹침 수}}{\text{전체음의 수}}$$

수직적 겹침 수는 동시에 발생하며 지속 시간이 동일한 음들의 개수를 뜻한다. 그림 1에서 첫째 마디에서는 수직적 겹침 수가 3이 되고 두 번째 마디에서는 6이 된다. 수평적 겹침 수는 둘 이상의 음이 서로 조금이라도 겹치는 부분이 존재하는데 수직적 겹침이 아닌 모든 경우의 수를 의미한다. 그림 1에서는 수직적 겹침만이 존재하므로 수평적 겹침 수는 0이 된다.

$$\text{평균 수직적 겹침 비율} = \frac{\sum(\text{수직적 겹침 비율})}{\text{처리한 마디 수}}$$

$$\text{평균 수평적 겹침 비율} = \frac{\sum(\text{수평적 겹침 비율})}{\text{처리한 마디 수}}$$

위의 수식을 이용해 평균 수직적 겹침 비율과 평균 수평적 겹침 비율을 구한다. 여러 실험을 거친 결과 평균 수직적 겹침 비율이 0.8이상이고 평균 수평적 겹침 비율이 0.5이하라면 입력 받은 음악은 E 유형에 속한다고 볼 수 있음을 확인했다.

E 유형의 음악들은 수평적 겹침은 드물고 수직적 겹침이 음악의 대부분을 구성하므로 수평적 겹침에 대해

고려할 필요가 적다. 또한 E 유형의 음악들은 대부분 화음으로 구성되어 있는데 화음의 경우에는 가장 높은 음이 멜로디에 해당한다는 기본 원칙[2]에서 벗어나는 경우가 드물다. 그러므로 E 유형의 음악에서 멜로디를 추출하는 알고리즘으로는 Skyline Algorithm을 그대로 사용해도 큰 문제가 없을 것이다.

##### 4.2 H 유형의 판별 및 멜로디 추출

H 유형의 경우에는 E 유형이나 L 유형과는 달리 특별히 일관되는 특징을 찾기 어렵다는 것을 3.2절에서 이미 확인했다. 그러므로 입력 받은 음악이 H 유형인지 파악하는 모듈을 따로 제작하기 보다는 E 유형과 L 유형의 검사 절차를 밟은 후에 어느 쪽에도 속하지 않는다면 입력 받은 음악은 H 유형에 속한다고 판단할 수 있다.

H 유형의 음악에는 반주와 멜로디가 존재하기 때문에 Skyline Algorithm처럼 수평적 겹침을 무시할 수는 없다. 그러므로 H 유형의 음악으로부터 멜로디를 추출할 때에는 특별한 경우가 아니라면 선택된 멜로디음들의 지속 시간을 지켜야 한다. 화음으로 이루어진 음들의 경우에는 여러 음들이 동시에 발생하는 경우 사람들은 대개 가장 높은 음을 멜로디로 인식한다는 기본 원칙[2]을 따라 가장 높은 음을 선택하도록 한다.

##### 4.3 L 유형의 판별 및 멜로디 추출

L 유형은 상단부의 반주음들의 지속 시간이 매우 일정하고 진행 속도가 하단부의 멜로디음들보다 빠르다는 특징을 갖고 있는데 이는 아래와 같이 표현할 수 있다.

$$\text{다수음의 평균 음높이} = \frac{\sum(\text{다수음의 음높이})}{\text{다수음의 개수}}$$

$$\text{나머지음의 평균 음높이} = \frac{\sum(\text{나머지음의 음높이})}{\text{나머지음의 개수}}$$

다수음이란 한 마디 내에서 지속 시간이 동일한 음들 중에서 가장 개수가 많은 음을 뜻한다. 나머지음이란 다수음외의 모든 음을 뜻한다. 그림 5의 경우에는 모든 마디에서 16분음표가 다수음이 된다.

L 유형의 특징을 고려하면 하나의 마디 내에서 지속 시간을 기준으로 구분했을 때 가장 많이 존재하는 음을 반주음이라고 볼 수 있다. 즉 하나의 마디 내에서 다수음이 바로 반주음이라고 판단할 수 있다. 그리고 L 유형에서 반주음은 당연히 상단부에 위치하므로 다수음의 평균 음높이는 나머지음의 평균 음높이보다 높을 것이다.

평균 음높이 차 = 한 마디 내에서의(다수음의 평균 음높이 - 나머지음의 평균 음높이)

$$\text{평균 음높이 차의 평균값} = \frac{\sum(\text{평균 음높이 차})}{\text{처리한 미디 수}}$$

위 수식을 이용해 입력 받은 음악의 전체 평균 음높이 차의 평균값과 표준 편차를 구한다. 미디 파일에서는 음의 높이가 양의 정수로 표현되는데 1의 차이가 바로

반음 하나의 차이를 뜻하고 음의 높이가 높을수록 큰 정수 값을 갖는다. L 유형의 음악이라면 당연히 평균 음높이 차의 평균값이 양의 값을 가질 것이다. 여러 실험을 거친 결과 평균값이 10보다 크다면 입력 받은 음악이 L 유형에 속한다고 볼 수 있음을 확인했다.

L 유형의 음악은 H 유형과는 달리 음의 지속 시간이 문제가 되는 경우는 거의 없다고 볼 수 있지만 동시에 발생하는 음들의 경우 어떤 음을 선택할지를 판단하는 것이 매우 어렵다. 반주음들이 상단부에 존재하므로 동시에 발생한 음들 중에서 가장 높은 음을 택하거나 가장 낮은 음을 택하는 방법을 취할 수 없기 때문이다.

L 유형은 반주와 멜로디를 분리하면 비교적 쉽게 멜로디를 추출할 수 있다. 같은 시간에 여러 음이 발생하는 경우에는 다수음, 즉 반주에 해당하는 음을 모두 제거한 후에 남은 음들 중에서 가장 음높이가 높은 음을 택하고 그 음의 지속 시간을 지켜주면 L 유형의 음악에서 멜로디를 성공적으로 추출할 수 있다. 그림 5의 첫째 마디의 경우에는 16분음표가 다수음이다. 그러므로 첫째 마디의 첫 발생음들 중에서 다수음인 16분음표를 제거하면 G#이 남고 이것이 바로 멜로디음이므로 성공적으로 멜로디를 추출한 것이 된다.

5. 구현 및 결과

5.1 시스템 구조

본 연구에서 구현한 시스템의 구조는 그림 6과 같다. 시스템은 리눅스를 운영체제로 하고 파이썬을 이용해서 구현했다. 전처리 과정은 미디 파일을 입력 받아서 이후 모듈에서 처리하기 좋은 리스트의 형태로 음들을 저장하기 위한 절차이다. 유형 판별 모듈은 저장된 음들의 리스트를 넘겨 받아 입력 받은 음악이 어떤 유형에 속하는지 결정한다. 멜로디 추출 모듈은 입력 받은 음악의 유형이 결정된 후 리스트에 담긴 음들을 받아 유형에 맞춰 멜로디를 추출해낸다. 미디 파일 생성 모듈은 추출한 멜로디 리스트를 미디 파일로 다시 제작해주는 역할을 한다.

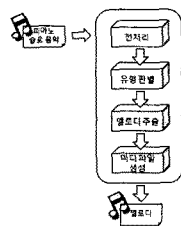


그림 6 시스템 구조도

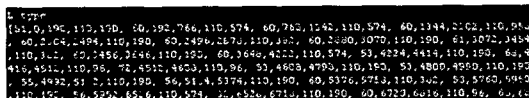


그림 7 리스트의 사랑의 꿈의 멜로디 추출 결과

5.2 평가

평가는 유형 자동 판별 모듈과 멜로디 추출 모듈의 성능을 각각 측정하는 방식으로 한다. 테스트 대상은 자료 분석 및 시스템 구현 과정에서 사용한 다양한 장르의 음악 22곡, 즉 Training Set(표 1)과 연구 과정에서 사용하지 않은 여러 장르의 음악 20곡, 즉 Test Set(표 2)으로 구성된다.<sup>1)</sup> 모든 음악은 발췌 음악으로서 10초부터 1분여까지 길이가 다양하다.

그림 8은 유형 판별 모듈의 실험 결과이다. 그림에서 X축은 마디 수를 나타내며 Y축은 정확도(Precision)을 뜻한다. 우선 Training Set과 Test Set의 각 음악을 한 마디 단위로 잘라서 각 마디가 어떤 유형인지 판별했다. 그 후에는 N-Gram 형식과 같이 보는 대상을 한 마디씩 증가시키면서 유형 판별 실험을 했다. 그림 8을 보면 알 수 있듯이 두 그룹 모두 마디를 많이 볼수록 정확도가 증가하며 로그 형태의 곡선을 보이고 있다.

표 1과 표 2는 멜로디 추출 모듈의 실험 결과로 재현율(Recall)과 정확도(Precision)을 이용해 성능을 측정했다. 각각은 아래와 같은 수식으로 정의된다.

$$\text{재현율} = \frac{\text{추출된 정답멜로디음의 개수}}{\text{전체 정답멜로디음의 개수}} \times 100$$

$$\text{정확도} = \frac{\text{추출된 정답멜로디음의 개수}}{\text{추출된 전체음의 개수}} \times 100$$

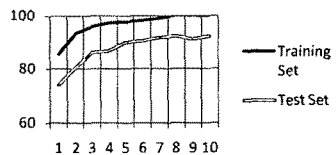


그림 8 유형 판별 모듈의 실험 결과

표 1 Training Set의 성능 측정 결과

음악 이름	재현율	정확도
쇼팽, 군대 폴로네이즈	100.0	100.0
베토벤, 월광 소나타 2악장	100.0	81.58
요코 칸노, Piano Bar	100.0	97.87
.	.	.
평균값	99.05	88.62

표 2 Test Set의 성능 측정 결과

음악 이름	재현율	정확도
베토벤, 비창 1악장	100.0	100.0
라흐마니노프, 전주곡 Op.23 No.5	100.0	100.0
트비시, 달빛	100.0	100.0
.	.	.
평균값	94.76	89.93

1) 전체 음악 목록은 [http://nlp.kaist.ac.kr/~yjchoi/music\\_list](http://nlp.kaist.ac.kr/~yjchoi/music_list)에서 확인할 수 있다.

Training Set의 평균 재현율과 평균 정확도는 각각 99.05, 88.62(표 1)이며 Test Set의 평균 재현율과 평균 정확도는 각각 94.76, 89.93(표 2)이다. 결과 수치를 보면 알 수 있듯이 구현된 시스템은 대부분의 경우 음악으로부터 멜로디를 성공적으로 추출하고 있다.

그림 9는 Training Set과 Test Set을 대상으로 기존의 멜로디 추출 알고리즘들과 제안한 알고리즘의 성능을 비교 평가한 것이다. Skyline Algorithm[2]의 경우에는 최대한 많은 음을 추출한다는 원칙에 의해 재현율이 타 알고리즘들보다 높은 수치를 기록하고 있지만 그러한 원칙 때문에 정확도가 떨어지는 것을 볼 수 있다. Revised Skyline Algorithm[9]의 경우에는 Time Overlap Parameter를 사용해 음들을 제거해 나가는 방식을 채택했기 때문에 재현율은 조금 떨어지지만 정확도는 전반적으로 높다. Skyline Algorithm과 Revised Skyline Algorithm은 모두 멜로디가 하단에 존재하는 음악은 제대로 처리하지 못한다는 한계점을 가지는데 그에 비해 Melody Lines[10]는 하단에 멜로디가 존재하는 경우에도 어느 정도 대처하는 모습을 보이긴 했지만 전반적인 성능은 다른 두 알고리즘에 비해 떨어졌다. 제안된 알고리즘은 재현율과 정확도에서 모두 기존 알고리즘들보다 높은 성능을 보임으로써 음악을 특성에 따라 분류해서 각 유형에 맞춰 자기 다른 멜로디 추출 방법을 적용하는 방법이 유효하다는 것을 보여준다.

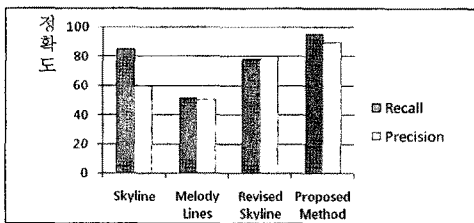


그림 9 전체 데이터를 대상으로 한 성능 비교

성능 평가 결과 전체적으로 높은 수치를 기록했지만 예외적인 경우도 있었는데 쇼팽의 혁명 연습곡과 같이 매우 빠른 반주음들과 휴식 기간이 짧은 멜로디를 갖는 음악의 경우에는 정확도가 50% 전후를 기록하기도 했다. 쇼팽의 혁명 연습곡의 경우, 반주의 속도가 매우 빠르기 때문에 짧은 휴식 기간에도 많은 양의 반주음이 포함될 수 있어서 정확도가 더욱 떨어진다

쇼팽의 왈츠 Op.64 No.2의 경우에는 재현율과 정확도 모두 20% 미만을 기록했는데 이는 초기에 유형 판별 모듈에서 곡의 유형을 잘못 판단했기 때문이다. 이 음악이 맞는 유형으로 판별된다고 가정하고 재현율과 정확도를 계산해 본 결과 두 수치 모두 100%를 기록했다.

## 6. 결론 및 향후 계획

본 논문에서는 피아노 솔로 음악을 음악의 특성에 따라 세 가지 유형으로 나누어 분석하여 유형별 특징을 조사하고 각 유형으로부터 멜로디를 추출하는 방법에 대해 논의하였다. 제안된 방법을 이용해 멜로디 추출 시스템을 구현하였으며 재현율과 정확도를 기준으로 시스템의 성능을 측정한 결과 다양한 피아노 솔로 음악에 적용 가능함을 확인했다. 특히 제안한 시스템은 과거 10여 년간 다양한 형태로 폭넓게 사용되어 온 Skyline Algorithm에 비하여 매우 높은 성능을 제공하는 것으로서 유형의 분류가 적절하게 선행되는 것이 정확한 멜로디의 추출에 매우 중요하다는 것을 보인다.

추후에는 하나의 음악 내에 둘 이상의 유형이 혼재하는 경우를 다룰 수 있도록 음악을 의미 있게 분할하는 연구를 진행할 계획이다. 그리고 5절에서 확인했듯이 아직 유형 판별이 완벽하게 이루어지는 것이 아니므로 더욱 많은 피아노 음악을 분석해 유형 판별 모듈의 성능을 향상시키면 시스템의 전체적인 완성도가 한층 높아질 것이라 기대된다.

## 참고 문헌

- [1] Digital Music Report 2009, International Federation of the Phonographic Industry(IFPI).
- [2] A. Uitdenbogerd, J. Zobel, "Manipulation of Music For Melody Matching," *Proc. ACM international conference on Multimedia*, pp.235-240, 1998.
- [3] M. Goto, "A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, pp.311-329, 2004.
- [4] M. Rynanen, A. Klapuri, "Transcription of the Singing Melody in Polyphonic Music," *Proc. ISMIR*, pp.222-227, 2006.
- [5] C. Isikhan, G. Ozcan, "A Survey of Melody Extraction Techniques For Music Information Retrieval," *CIM08 Abstract Proc.*, pp.82-83, 2008.
- [6] G. Ozcan, C. Isikhan, A. Alpkocak, "Melody Extraction on MIDI Music Files," *Proc. of the Seventh IEEE International Symposium on Multimedia*, pp. 414-422, 2005.
- [7] B. Hill, "A Complete Idiot's Guide to Playing Piano," *Alpha Books*, p.109, 2001.
- [8] R. Frances, "La Perception de la Musique," Vrin, 1958.
- [9] W. Chai, "Melody Retrieval on the Web," MS Thesis, Massachusetts Institute of Technology, 2000.
- [10] MidiLib Project Team, [Http://www-mmdb.iai.uni-bonn.de/forschungprojekte/midilib/english](http://www-mmdb.iai.uni-bonn.de/forschungprojekte/midilib/english).