멜로디 추출 알고리즘으로 살펴보는 음악정보검색

금상은
한국과학기술원
문화기술대학원

남주한
한국과학기술원
문화기술대학원

Ⅰ. 서 론

음반 시장이 LP, 카세트테이프, CD, MP3 파일로 이어지는 소유 중심에서 스트리밍 기반 서비스로 재편되면서, 스마트폰만 있으면 언제 어디서나 원하는 음악을 들 수 있는 시대가 되었다. 이러한 음악에 대한 접근성의 향상과 더불어, 사용자가 원하는 음악을 보다 쉽게 검색하거나 추천 받는 등의 새로운 서비스 또한 증가하고 있다. 예를 들어, 카페에서 흘러나오는 음악이 어떤 곡인지 스마트폰 마이크 이용하여 알려 주거나, 사용자의 음악 소비 형태를 분석하고 이를 바탕으로 라디오 형태로 맞춤형 선곡을 해주는 서비스 등이 있다. iTunes, Pandora, Shazam, SoundHound, Spotify 해외 업체들은 이러한 기술들을 활용하여 적극적으로 사용자에게 양질의 서비스를 제공하고 있다〈그림 1〉. 이러한 음원 서비스에는 음악 신호를 분석하고 이로부터 음악에 대한 다양한 정보를 추출하는 기술이 그 근간을 이루고 있는데, 이렇게 음악 정보를 분석하고 관련된 어플리케이션을 개발하는 학제간 학문을 Music Information Retrieval (MIR) 이라고 한다. MIR에서는 음악학, 심리학, 신호처리, 기계학습 등 다양한 학문이 결합되어 음악에 관련된 여러 가지 과제들을 해결해 나가고 있다[1]. 예를 들면 곡 인식 (song identification), 커버 곡 (cover song) 검색, 장르 및 무드 분류 (genre and mood classification), 자동 악보 채보 (automatic music transcription), 코드 인식 (chord recognition) 등 다양한 종류가 있다. 또한 이런 과제들에 대한 알고리즘 평가는 Music Information Retrieval Evaluation eXchange (MIREX)[1]을 통해서 매년 활발히 이루어지고 있다.

1) http://www.music-ir.org/mirex/wiki/MIREX_HOME

〈그림 1〉 디지털 음악 서비스 예, Spotify, Pandora, Shazam, SoundHound

음악에 담겨있는 정보를 추출하기 위해서는 기본적으로 음악의 주요 3요소인 멜로디, 리듬, 화성을 분석하는 것이 중요하다. 그 중에서도 멜로디는 각각의 곡의 특징을 가장 잘 설명해주는 요소이며 특별한 교육 없이도 직관적으로 인지 할 수 있는 기본적인 음악 정보이다. 사람은 음악을 들을 때 어떤 소리가 멜로디인지 쉽게 구별할 수 있지만, 이것을 컴퓨터로 구현하기 위해서는 해결해야 할 문제점이 많다. 이 글에서는 그러한 문제점들을 어떻게 해결하고 음악에서 멜로디를 추출하는지에 대해서 알아보고자 한다.

다음 장에서는 멜로디의 일반적인 정의와 공학적으로 멜로디를 어떻게 정의하는지 다루며, 멜로디 추출 알고리즘을 접근 방식에 따라 크게 3가지로 나눠서 설명한다. 그리고 멜로디 추출로 가능한 적용 분야를 소개하며 본문을 마무리 하고자 한다.

Ⅱ. 멜로디란 무엇인가?

1. 멜로디의 정의

우리가 평상시 듣는 대부분의 음악들은 주요 멜로디를 중심으로 반주가 덧붙여 만들어 진다. 따라서 멜로디에 작곡자의 주요 창작 의도가 담겨 있으며, 서로 다른 음악과 구별할 수 있는 중요한 척도가 된다. 그러므로 음악에서 멜로디를 추출하는 것은 그 음악의 가장 중요한 특징을 뽑아내는 과정이다.

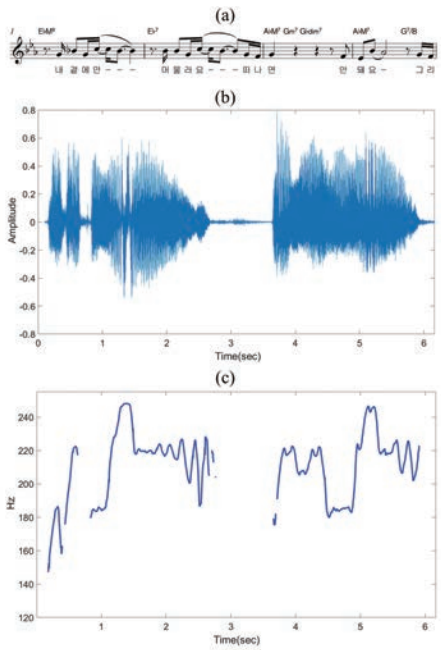
그렇다면 멜로디의 정의는 무엇인가? 멜로디의 사전적 정의는 "음의 높낮이의 변화가 리듬과 연결되어 하나의 음악적 통합으로 형성되는 음의 흐름"이다. 누군가 어떤 음악인지 물어봤을 때, 그 음악을 가장 직관적으로 묘사하는 방법은 기억나는 음을 흥얼거리는 것이다. 이처럼 멜로디는 사람이 음악을 듣고 본능적으로 노래나 허밍으로 표현되는 것으로도 정의될 수 있다[2]. 멜로디를 기호로 기록하는 하는 방법으로 〈그림 2(a)〉 에서와 같이 악보로 표현하는 방법이 있다.

2. 멜로디의 공학적인 접근

위의 정의는 음악학적 정의이다. 그렇다면 공학적으로는 어떻게 정의할 수 있을까? 객관적이고 정량적으로 멜로디를 측정하기 위해서는 명료한 정의가 필요하다. 그래서 MIR 분야에서는 "멜로디는 각각 음원의 pitch 변화 흐름 중 가장 두드러진 것"[3] 이라고 통용된다.

> MIR 분야에서는 "멜로디는 각각 음원의 pitch(음고) 변화 흐름 중 가장 두드러진 것"이라고 통용된다.

그렇다면 pitch (음고) 는 무엇인가? Pitch는 음의 주파수가 높고 낮음을 뜻하는 것으로 사람의 청각 인지에



〈그림 2〉 이문세의 '소녀' (a) 악보, (b) raw waveform, (c) F_0변화 흐름 (=멜로디), YIN 알고리즘을 사용하여 추출

따른 용어다[4]. Pitch를 이야기할 때 가장 물리학적인 용어로 가장 연관되는 것이 기본 주파수 (fundamental frequency) 이다. 기본 주파수 (F_0)는 주기 신호에서 주기 (T)의 역수로 정의되며 신호의 첫 번째 하모닉스 성분을 의미한다.

$$F_0 = \frac{1}{T}$$

일반적인 오디오 신호는 F_0의 배수가 되는 하모닉스 성분으로 구성되므로 F_0을 기본 주파수로 부른다. 이때 F_0는 측정 가능한 물리학적인 양으로 정의되며, 오디오 신호처리에서는 F_0와 pitch를 서로 같은 개념으로 생각한다. 즉 멜로디란 F_0변화의 흐름이다.

따라서, 멜로디를 추출한다는 것은 여러 음원 가운데에서 멜로디로 구별되는 음원의 F_0변화 흐름을 알아내는 것이다. 〈그림 2(b)〉와 같이 음악 신호의 파형 (waveform)을 분석하여 〈그림 2(c)〉와 같이 F_0변화의 흐름으로 멜로디를 표현 할 수 있다.

Ⅲ. 멜로디 추출 알고리즘

1. 단선율 멜로디 추출

〈그림 2(c)〉와 같이 F_0을 구하기 위해서 어떤 방법을 사용할 수 있을까? 단선율 (monophonic) 오디오 신호인 경우에는 F_0을 알아내기 위해 시간 영역에서 Auto Correlation Function (ACF)[5] 이나 Average Magnitude Difference Function (AMDF)[6] 가 흔히들 사용된다. ACF를 이용하는 경우 아래와 같이 pitch를 추출할 수 있다

$$f_{monophnic} = \arg\max_f \sum_\tau S_y(f,\tau) + C(f)$$

$$S_y(f,\tau) = \frac{1}{W}\int_{\tau - W/2}^{\tau + W/2} y(t)y(t+\frac{1}{f})dt$$

여기서 $f_{monophonic}$은 매 프레임 τ에 해당하는 pitch값을 의미한다. W는 ACF 의 윈도우 사이즈이며, $y(t)$는 분석하고자 하는 단선율 멜로디 신호이다.

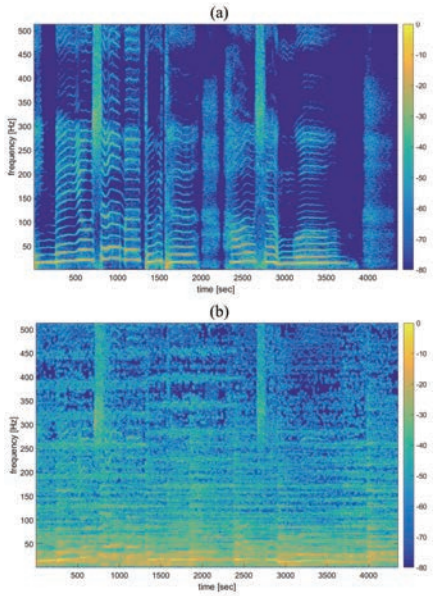
또 다른 방법으로는 오디오 신호를 Short-Time Fourier Transform (STFT) 으로 변환 후 주파수 영역에서 F_0을 구하는 방법이 있다. 이러한 방법에는 Harmonic pattern matching[7], Cepstrum[8], Harmonic-Product-Sum[9] 등 다양한 알고리즘이 존재한다.

단선율 멜로디 추출 알고리즘 중에선 YIN 알고리즘[6]이 가장 대표적으로 사용되는데, 오차가 2% 내외로 높은 정확도로 pitch를 알아낼 수 있다. 따라서 단선율의 음악 신호 같은 경우에는 큰 문제없이 F_0을 구할 수 있다고 여겨지며, 다성음으로 구성된 음악에서 멜로디 추출 알고리즘의 성능을 평가할 때, 평가 기준이 되는 ground truth pitch를 구하는데 사용된다.

2. 다성음 멜로디 추출

우리가 즐겨듣는 음악의 대부분은 여러 개의 악기 소리로 구성되어 있고, 여러 개의 pitch가 섞여있는 다성음 (polyphony)을 가지고 있다. 이러한 음악 신호로부터 멜로디를 추출하는 것은 다음과 같은 이유로 매우 어려운 문제이다.

첫째, 멜로디를 추출하기 위해서는 음악 신호로부터 멜로디에 해당되는 부분과 반주에 해당되는 부분을 구분해야한다. 〈그림 3(a)〉은 사람 목소리만 존재하는 단선율 음악의 스펙트로그램 (spectrogram) 이며, 〈그림 3(b)〉



〈그림 3〉 (a) monophonic 스펙트럼 (목소리) (b) polyphonic 스펙트럼 (목소리 + 반주)

은 목소리와 함께 반주도 포함된 다성 음악의 스펙트로그램이다. 〈그림 3(b)〉에서 볼 수 있듯이 〈그림3 (a)〉에 비해 하모닉스가 불분명하고 F_0에 해당하는 하모닉스 성분을 구분하기가 더 어렵다. 특히, 멜로디와 반주는 화성적으로 어울리도록 배치되기 때문에 멜로디 음의 하모닉스와 반주 음의 하모닉스가 서로 섞여서 구분하기가 더욱 어려워진다.

둘째, 음악 제작 시 여러 트랙의 악기 소리를 믹싱 (mixing) 단계에서 소리의 변형이 일어난다. 예를 들어, 잔향 (reverberation) 나 컴프레서 (compressor) 등 오디오 효과를 이용하여 후처리를 하는 경우가 많은데, 이는 본래 신호를 변형시켜서 각각의 소리를 분리하기 더욱 어렵게 만든다.

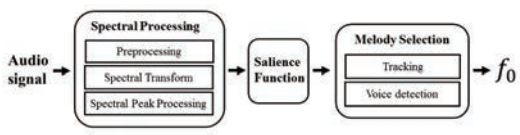
셋째, 멜로디가 존재하는 구간을 파악해야 한다. 대중음악에서 멜로디는 일반적으로 보컬 음에 의해서 결정되는데, 보컬 음은 노래를 부르는 중간에 호흡이나 음악적 효과를 위해 쉬는 구간 (short pause)이 있으며 전주와 간주 부분에는 전체적으로 쉬어 간다. 따라서 멜로디 추출과 별도로 음원을 검출 (voice detection) 하는 알고리즘이 필요하다.

멜로디 추출 기법은 Salience function을 사용한 방법, 음원 분리 기반 방법, 그리고 머신 러닝을 이용한 데이터 기반의 방법으로 크게 3가지로 나눌 수 있다.

이러한 문제점을 극복하고 보다 높은 성능으로 멜로디를 추출하기 위해 다양한 방법이 제안되어 왔는데, 이들은 접근 방식을 볼 때 Salience function을 사용한 방법, 음원 분리 기반 방법, 그리고 머신 러닝을 이용한 데이터 기반의 방법으로 크게 3가지로 나눌 수 있다.

2.1 Salience 기반 방법

Salience 기반 방법은 멜로디 추출 알고리즘 중 가장 많이 사용되는 방식으로 salience function을 사용하여 pitch 성분을 찾아내는 방식이다. 그 순서는 각각의 알고리즘마다 다르지만 크게는 〈그림 4〉의 순서를 따른다 [10].



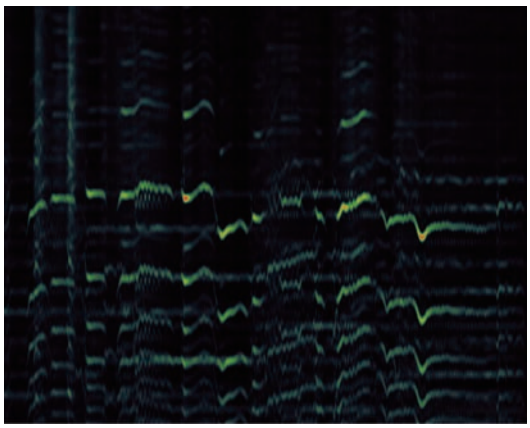
〈그림 4〉 Salience 기반 방법 Block diagram

2.1.1 Preprocessing

본격적으로 신호를 분석하기 전에 원 신호를 적절히 변형을 해주어 원하는 결과를 얻기에 수월하도록 전처리 과정을 거친다. 전통적인 방법으로는 band pass filter를 사용하여 신호에서 멜로디에 해당하는 주파수 성분만을 집중적으로 관찰하거나, 신호의 크기를 정규화 해주는 방법이 있다. 특별한 방법으로는 Harmonic-Percussive Sound Separation (HPSS)을 사용하여 멜로디와 반주 성분을 먼저 분리하기도 한다.

2.1.2 Spectral Transform

다성음으로 구성된 음악 신호에서는 시간 축 상의 정보만으로 멜로디를 추출하는 것이 불가능하므로 스펙트럼 형태로 변형을 해주는 것이 일반적이다. 가장 대표적으로 Short-Time Fourier Transform (STFT)을 사용하는데 〈그림 3〉처럼 시간에 변화에 따라 주파수 성분의 변화를 볼 수 있다. 하지만 STFT는 주파수 축과 시간 축 간의 해상도가 서로 trade-off되기 때문에 주파수 대역별로 해상도의 한계가 존재한다. 우리가 듣는 pitch는 한 옥타브가 증가할 때마다 주파수는 2배씩 기하급수적으로 증가하기 때문이다 (예. A3 (라) = 220Hz, A4 = 440Hz). 이를 보완하고자 constant-Q



〈그림 5〉 Melodia를 사용하여 추출한 Salience

transform[11], multi-resolution FFT[12]를 사용하기도 한다.

2.1.3 Salience function

Spectral Transform 다음으로는 멜로디가 될 수 있는 pitch 성분들에 대해서 salience를 구한다. 이 방법 역시 다양한 방법들이 존재하는데 가장 기본적인 방법은 harmonic 성분들의 크기에 가중치를 주어 합하는 방법이다[10].

$$f_{salinece} = \arg\max_f \sum_\tau S'_y(f_\tau,\tau) + C'(f)$$

$$S'_y(f_\tau,\tau) = \sum_{h=1}^{N_h} g(f_\tau,h)|Y(h \cdot f,\tau)|$$

$$Y(f,\tau) = \int_{-W/2}^{W/2} w(t)y(\tau+t)e^{-j2\pi ft}dt$$

$f_{salinece}$는 salience를 사용하여 얻은 멜로디의 pitch이다. $Y(f,\tau)$는 신호 $y(t)$의 스펙트럼이며 $w(t)$는 윈도우 함수를 의미한다. $S'_y(f_\tau,\tau)$는 salience function이며 N_h는 스펙트럼 상에서 하모닉스의 개수를 의미한다. 각 하모닉스에 곱해지는 $g(f_\tau,h)$는 가중치를 의미한다.

다른 방법으로는 멜로디에 해당하지 않을 것 같은 성분들을 하나씩 제거하는 방법이 있다[13]. Melodia[2]는 이 방법을 사용하여 salience를 구하는데, 〈그림 5〉는 Melodia를 사용하여 얻은 salience를 보여주고 있다. salience의 peak값들은 멜로디 pitch의 후보군에 포함된다.

2.1.4 Tracking

Salience function의 peak에서 어떤 peak가 멜로디인지 결정해야 하는데, 가장 대표적인 방법은 가장 확률이 높은 salience peak 을 멜로디로 고르는 것이다. 또는 Hidden Markov Model (HMM)을 사용하거나 dynamic programming을 사용해서 멜로디를 정하는 방법도 있다.

2.1.5 Voice Detection

멜로디 추출을 한 뒤 그 부분이 실제 멜로디 부분인지 아닌지를 판단하는 것이다. 실제로는 멜로디가 없는 부분이지만 반주 부분의 성분으로 인해 멜로디로 오판될 수 있다. 따라서 voice detection을 사용하여 멜로디 부분과 반주 부분을 명확하게 구분해 주어야 정확도를 높일 수 있다. 가장 간단하게는 energy threshold를 사용하여 그 이상의 에너지를 가진 프레임을 멜로디로 구별하는 방법이 있으며, 이외에 dynamic threshold나 loudness filter를 사용하거나, salience function의 분포를 보고 결정하는 방법이 있다.

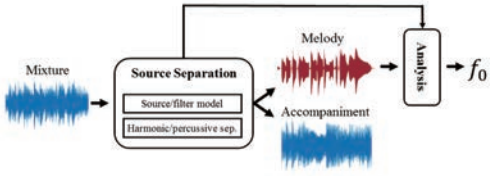
2.2 음원 분리 기반 방법

음원 분리 기반 방법은 〈그림 6〉처럼 기본적으로 멜로디와 반주를 분리한 다음, 멜로디의 pitch를 알아내는 방법이다.

Source/filter model[14]로 분리하거나 Harmonic-Percussive Sound Separation (HPSS)을 사용하여 멜로디를 추출하는 방법[15]이 있다.

HPSS는 원래 harmonic sound (H)와 percussive sound (P)를 분리하기 위해 고안되었다[16]. HPSS를 사용하여 H와 P를 분리하면 Harmonic 부분에는 시간적 안정성이 있는 성분, 즉 반주에 해당하는 정보가 존재하며, percussive 부분에는 시간적 변동이 있는 성분 즉, melody + percussive 소리 정보가 존재하게 된다.

[15]에서는 HPSS를 두 번 사용하여서 멜로디를 추출한다. 첫 번째 단계에서는 window 길이를 크게 하여서 (≃200ms) 주파수 축 해상도를 높여 melody + percussive 성분 P^1을 얻는다. 두 번째 HPSS을 사용할 때는 window 길이를 작게 하여 (≃30ms) 시간 축 해상도를 높여 멜로디 성분을 분리한다. 이때 멜로디 정보들은 Harmonic 부분 H^2에 존재한다.



〈그림 6〉 음원 분리 기반 접근 방법 Block diagram

2) http://mtg.upf.edu/technologies/melodia

$$W(t) \xrightarrow{HPSS\ with\ long\ window} H^1(t), P^1(t),$$
$$P^1(t) \xrightarrow{HPSS\ with\ short\ window} H^2(t), P^2(t).$$

이렇게 멜로디를 반주와 분리한 다음 F_0을 구하는 방법을 사용한다.

Hsu[17]는 singing voice의 대략적인 pitch를 예측한 다음 반주와 목소리를 분리하는 과정을 거친다. 예측한 pitch와 분리한 멜로디의 pitch가 수렴할 때까지 같은 작업을 반복하여 최종 멜로디의 pitch를 구하는 방법을 사용하고 있다. 음원을 분리하는 기술은 멜로디 추출 뿐만 아니라 다른 응용 프로그램에도 적용할 수 있어 활용도가 높은 알고리즘이다.

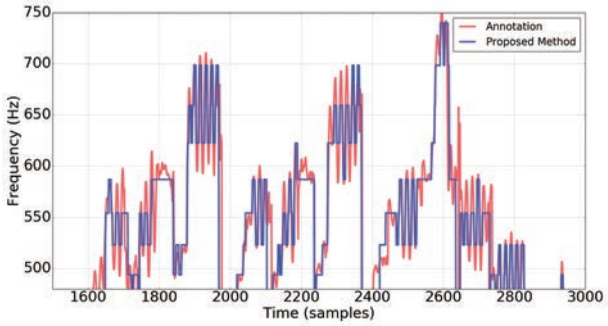
2.3 Data-driven 방법

멜로디 추출 기법은 주로 salience 방법과 음원 분리 방법이 사용된다. 위의 두 방법과는 달리 기계학습을 이용한 data-driven 방법을 사용한 알고리즘도 제시되었다. Poliner[18]은 Support Vector Machine (SVM) 을 이용하여 스펙트럼과 멜로디의 MIDI note를 학습시켜 MIDI 스케일의 pitch를 알아내는 방법을 사용하였다.

오디오 신호를 8 kHz 로 down-sampling 을 한 후 스펙트럼으로 변환한다. 스펙트럼의 hop size는 10 ms 가 되도록 FFT 크기와 원도우 크기를 조정해 준다. 이에 각 프레임에 해당하는 멜로디의 주파수를 MIDI note로 변환시켜 양자화 시킨다. 연속된 주파수를 학습 데이터의 결과로 사용할 수는 없기 때문이다. MIDI note는 다음과 같이 계산한다.

$$MIDI = \mathrm{round}(69 + 12\log_2(f_m/440Hz))$$

이때 총 output label는 G2에서 F#7에 해당하는 미디 값으로 총 60개이다. 스펙트럼의 각각의 프레임은 60개의 결과 label 중 하나와 대응되며, SVM 을 이용하여 멜로디 추출 모델을 학습시킨다. 학습된 모델을 사용하여 60개의 MIDI note 가 멜로디가 될 확률을 test data의 각각의 프레임에 대해서 구한다. 최종 결과는 hidden Markov model를 사용하여 가장 가능성이 높은 경로를 선택하여 멜로디를 추출한다.



〈그림 7〉 Data-driven 기반 방법 예시. 붉은 선은 테스트 음원의 실제 멜로디 주파수를 표시하고 있음. 파란 선은 MIDI스케일로 학습한 모델로 테스트 음원의 멜로디를 추출한 것.

$$\prod_t p(c_t|q_t)p(q_t|q_{t-1})$$

c_t는 주어진 ground truth pitch q_t에 대한 예측값을 의미하며, $p(q_t|q_{t-1})$는 학습시킨 멜로디의 ground truth에서 transition matrix를 구할 수 있다. $p(c_t|q_t)$를 구하기 위해서 베이즈 정리를 사용한다.

$$p(q_t|x_t) \propto p(x_t|q_t)p(q_t)$$

$p(q_t)$는 ground truth에서 각 pitch에 따른 확률을 구할 수 있으며, $p(q_t|x_t)$는 SVM에서 예측한 확률을 사용한다.

Data-driven의 방법은 기계 학습을 통해 멜로디의 특징들을 추출하여, 다른 두 방법에 비해 알고리즘이 비교적 간단하다는 장점이 있다. 그러나 〈그림 7〉과 같이 결과 pitch 값이 양자화 된다는 단점이 있다. 또한 기계학습의 특성상 많은 양의 학습 데이터가 필요하며, 특정 학습 데이터만 과도하게 학습이 되어 (overfitting) 다른 테스트 데이터에는 정확도가 낮을 가능성이 있는 단점도 존재한다.

최근에는 딥러닝 (deep learning)과 같은 기계학습 기법이 다방면에 이용되며 혁신을 불러일으키고 있다. MIR 분야에서도 기계학습을 활용하여 각 분야의 문제들을 풀어나가고 있다. 하지만 기계학습의 특성상 레이블 된 양질의 학습 데이터를 대량으로 구하는 것이 쉽지 않다는 한계점이 있다. 그럼에도 불구하고 기존의 데이터의 전체

적인 pitch를 변경시키거나, loudness를 바꾸는 등 다양한 변화를 통해 기존 학습 데이터의 양을 늘려주는 방법[19]을 사용할 수도 있다. 또한 다양한 기계학습 기법들이 개발되고 있어 앞으로의 역할이 더욱 기대된다.

Ⅳ. 멜로디 추출 알고리즘 활용

지금까지 오디오 신호에서 멜로디를 추출하는 방법에 대해서 알아보았다. 그렇다면 음악에서 멜로디를 알 수 있다면 어떤 것들을 할 수 있을까?

대표적으로 사용자가 찾고자 하는 노래의 멜로디를 흥얼거리면 동일한 노래를 찾아 주는 음악 검색 서비스를 만들 수 있다. 이를 Query By Humming (QBH) 라고 한다. QBH는 이를 구현하기 위한 다양한 알고리즘이 존재하지만 기본적으로는 사용자가 부르고 있는 음의 pitch를 알아야하며, 찾고자 하는 곡의 멜로디가 제대로 추출 되어 있어야 한다. 이를 활용한 대표적인 서비스로는 SoundHound가 있다.

또한, 커버 곡 (cover song)을 찾는 곳에 활용 할 수 있다. 커버 곡은 메인 멜로디는 원곡과 같거나 비슷하지만 반주의 악기 구성, 코드 진행, 장르 등의 요소들이 다른 곡을 말한다. 원곡과 커버 곡을 구별하기 위해서는 멜로디와 반주를 구별하여 분리할 수 있어야 한다. 커버 곡 검색을 사용하면 현재 재생되고 있는 특정 음원 뿐 아니라 다른 버전의 음원들도 함께 검색할 수 있어 사용자에게 더욱 만족스러운 서비스를 제공할 수 있다. 사용자가 악기를 연주하거나 노래를 부르면 바로 악보로 표현해주는 기술 (automatic transcription)에도 사용할 수도 있다. 여기에는 동시에 연주되고 있는 음들의 pitch를 알아야하며 (multi-pitch detection) 박자, 그리고 언제 음이 시작되고 (onset detection) 끝나는지 (offset detection)를 정확하게 알아야 악보로 표현할 수 있다. 더 나아가, 멜로디 추출 알고리즘은 연주자가 연주하는 부분이 악보의 어느 부분인지 인지하고 자동으로 추적하거나 (score following)과 반주를 연주할 때도 적용이 가능하다 (auto-accompaniment).

멜로디 추출에 관련된 기술들은 음악 관련 분야 뿐 아니라 다른 오디오 신호 처리 분야에도 적용 될 수 있다. 주위 환경의 소리를 듣고 음원의 종류를 인지 (auditory scene analysis) 하는 등 다른 분야에도 활용 될 수 있어 앞으로 많은 응용 가능성이 있는 분야이다.

멜로디 추출에 관련된 기술들은 음악 관련 분야 뿐 아니라 주위 환경의 소리를 듣고 음원의 종류를 인지 (auditory scene analysis) 하는 등 다른 분야에도 활용 될 수 있다.

Ⅴ. 결 론

음악에 담겨있는 정보를 분석하는 방법들은 지금도 활발히 연구되고 있다. 그 연구의 결과는 매년 개최되는 MIREX에서 살펴볼 수 있다. 특별히 이 글에서는 범위를 좁혀 멜로디를 추출하는 방법에 대해서 간략하게 살펴보았다. 또한 추출한 멜로디를 활용하여 어떻게 활용할 수 있는지 소개하였다.

하지만 멜로디 정보만 가지고 있다면 4장에서 소개하였던 응용과제들을 완성할 수 없다. 예를 들어, 자동 채보 (automatic transcription) 가 가능하기 위해선 multi-pitch detection과 함께 onset/offset detection이 동시에 분석되어야 한다. 하지만 그 각각의 주제 자체가 큰 연구 분야 중 하나이며 각각 세분화 되어있게 때문에, 실제적인 응용을 위해서는 여러 알고리즘을 통합하여 실제 서비스와 결합해야 한다.

멜로디 추출 알고리즘 경우에는 현재까지 높은 정확도를 보여주고 있는 알고리즘들이 평균적인 정확도가 80% 정도에 다다르지만 2012년 이후로는 정확도가 이를 넘지 못하고 정체되고 있다[10]. 이를 위해서는 최근 각광 받고 있는 딥러닝과 같은 새로운 방법론을 적용해 볼 수 있다.

참 고 문 헌

[1] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based Music Information Retrieval: Current Directions and Future Challenges," IEEE Proc., vol. 96, no. 4,

pp. 668–696, 2008.

[2] G. E. Poliner, D. P. W. Ellis, a F. Ehmann, E. Gomez, S. Streich, and B. Ong, "Melody Transcription From Music–Audio: Approaches and Evaluation," IEEE Trans. Audio Speech Lang. Process., vol. 15, no. 4, p. 1247, 2007.

[3] R. P. Paiva, T. Mendes, and A. Cardoso, "Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience, and Melodic Smoothness," Comput. Music J., vol. 30, pp. 80–98, 2006.

[4] M. Mueller, D. Ellis, a. Klapuri, and G. Richard, "Signal Processing for Music Analysis," Sel. Top. Signal Process. IEEE J., vol. 6, no. 1, pp. 1–1, 2011.

[5] M. Sondhi, "New Methods of Pitch Extraction," IEEE Trans. on Audio and Electroacoustics, vol. 16, no. 2, pp. 262–266, 1968.

[6] A. de Cheveigne and H. Kawahara, "YIN, A Fundamental Frequency Estimator For Speech and Music.," J. Acoust. Soc. Am., vol. 111, no. 4, pp. 1917–1930, 2002.

[7] M. S. Puckette, T. Apel, and D. D. Zicarelli, "Real–time Audio Analysis Tools for PD and MSP," Analysis, vol. 74, pp. 109–112, 1998.

[8] C. P. Determination, "Cepstrum Pitch Determination," The journal of the acoustical society of America, no. August, pp.293–309, 1967.

[9] A. Noll, "Pitch Determination of Human Speech by the Harmonic Product Spectrum, the harmonic sum spectrum and a maximum likelihood estimate", Proceedings of the symposium on computer processing communications. vol. 779, 1969

[10] J. Salamon, E. Gomez, D. P. W. Ellis, and G. Richard, "Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges," IEEE Signal Process. Mag., vol. 31, no. February 2014, pp. 118–134, 2014.

[11] K. Dressler, "An Auditory Streaming Approach on Melody Extraction," MIREX Audio Melody Extraction Contest Abstract, pp. 19–24, 2006.

[12] T. Yeh, M. Wu, J. R. Jang, W. Chang, and I. Liao, "A Hybrid Approach to Singing Pitch Extraction Based on Trend Estimation and Hidden Markov Models," IEEE Int. Conf. Acoust. Speech Signal Process. – Proc., pp. 457–460, 2012.

[13] J. Salamon and E. Gomez, "Melody Extraction from Polyphonic Music Signals Using Pitch Contour Characteristics," IEEE Trans. Audio, Speech Lang. Process., vol. 20, no. 6, pp. 1759–1770, 2012.

[14] J. Durrieu and B. David, "Source / Filter Model for Unsupervised Main Melody," IEEE Trans. Audio, Speech Lang. Process., pp. 564–575, 2010.

[15] N. Ono, K. Miyamoto, H. Kameoka, J. Le Roux, Y. Uchiyama, E. Tsunoo, T. Nishimoto, and S. Sagayama, "Harmonic and Percussive Sound Separation and its Application to MIR–related Tasks," Stud. Comput. Intell., vol. 274, pp. 213–236, 2010.

[16] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, "Melody Line Estimation in Homophonic Music Audio Signals Based on Temporal–variability of Melodic Source," IEEE Int. Conf. Acoust. Speech Signal Process. – Proc., pp. 425–428, 2010.

[17] C. L. Hsu, D. Wang, J. S. R. Jang, and K. Hu, "A Tandem Algorithm for Singing Pitch Extraction and Voice Separation from Music Accompaniment," IEEE Trans. Audio, Speech Lang. Process., vol. 20, no. 5, pp. 1454–1463, 2012.

[18] D. P. W. Ellis and G. E. Poliner, "Classification–based Melody Transcription," Mach. Learn., vol. 65, no. 2–3, pp. 439–456, 2006.

[19] J. Schluter and T. Grill, "Exploring Data Augmentation for Improved Singing Voice Detection With Neural Networks," In Proceedings of the 16th International Society for Music Information Retrieval Conference, pp.121–126, 2015.

금상은

• 2014년 8월 경북대학교 전자공학부 졸
• 2014년 9월~현재 한국과학기술원 문화기술대학원 석사과정

〈관심분야〉
audio signal processing, MIR, melody extraction, deep learning

남주한

• 1998년 2월 서울대학교, 전기공학부, 학사
• 2010년 3월 Stanford University, EE, M.S.
• 2013년 1월 Stanford University, Music, Ph.D.
• 2001년 4월~2006년 7월 영창 뮤직, 소프트웨어 엔지니어
• 2012년 10월~2014년 9월 Qualcomm Technologies Inc.
• 2014년 9월~현재 카이스트 문화기술대학원

〈관심분야〉
audio signal processing, MIR, sound synthesis, digital audio effects, musical HCI