Audio Signal Processing Project - Singing, Humming으로 음악 검색하기

12171470 김성우

1. Project Overview

음악에서 후렴구 또는 주 멜로디가 되는 부분만을 분리하여, chromagram으로 저장하였다. 그리고 사용자가 노래를 부르거나 흥얼거리면 어떤 노래를 불렀는지 melody matching을 이용해 찾아주는 시스템을 구축하는 것이 이번 프로젝트의 목표이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 프로젝트 실행 과정

시스템의 각 단계를 간단히 설명하면 다음과 같다.

음악 파일에서 후렴구나 주 멜로디라고 여겨지는 부분을 chroma feature로 추출하여 numpy array 형태로 저장한다. chroma feature를 생성할 때, 노래에서 vocal만 따로 분리한다.

저장된 노래에서 저장한 부분 중 하나를 골라 노래를 직접 부르거나 흥얼거리고, 이를 녹음한다.

녹음된 음성의 chroma feature를 생성한다.

위에서 녹음에 의해 생성된 chroma feature와 저장되어 있는 chroma feature들을 서로 DTW(Dynamic Time Warping)을 통해 cost가 가장 작은 음악은 어떤 것인지 고른다. Cost가 가장 적은 음악이 사용자가 흥얼거린 음악이라고 최종 의사결정을 내린다.

1. Theoretical Background (이론적 배경)

**2-1) Chroma CENS (Chroma Energy Normalized Statistics) Feature [1]**

Chromagram은 음악 또는 오디오의 representation 방법 중 하나로, filter bank를 이용해 오디오에서 12음계의 pitch 성분을 추출한 것이다. CENS feature는 chroma feature에 다음과 같은 단계들을 거쳐서 얻어진다.

**L1 norm으로 normalizing 🡪 Quantization 🡪 window length w로 smoothing하고, factor d로 downsampling 🡪 L2 norm으로 normalizing**

CENS feature는 음계 이외의 요소에 대한 민감성을 줄여 멜로디 매칭에 유리한 특성을 가지고 있다. 아래 figure는 C4에서 C5까지 차례로 흰 건반을 눌렀을 때 Chroma CENS feature를 나타낸 것이다.

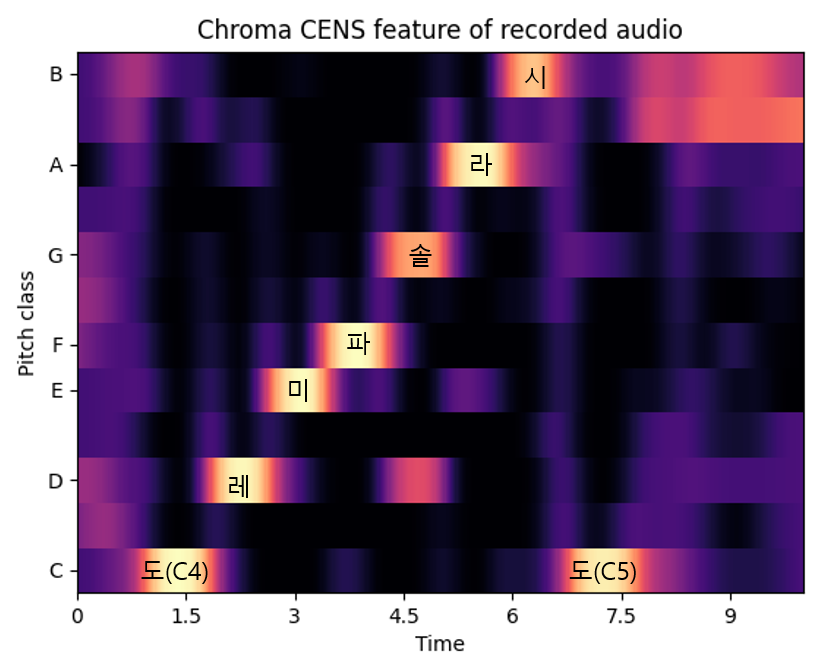


Figure Chroma CENS feature

위에서 알 수 있듯이, C4, C5를 구분하지 않는 것으로 보아 chroma feature는 pitch feature와 달리 옥타브를 구분하지 않음을 알 수 있다.

**2-2) Vocal Extraction (REPET, REpeating Pattern Extraction Technique) [2]**

음악 파일에서, Voice(foreground)는 반복되지 않는 요소(non-repeating)이며 상대적으로 time-frequency representation에서 sparce한 특징을 가지고 있다고 가정하자. 또한 music(background)은 반복되는(repeating) 요소라고 가정하자. 이 때 오디오의 magnitude spectrogram을 V, 그리고 VT x V = S, Similarity Matrix 라고 할 때, similarity를 이용하여 반복되는 요소(~music)와 반복되지 않는 요소(~voice)를 분리하여 masking 하는 것이 이 알고리즘의 핵심이다.

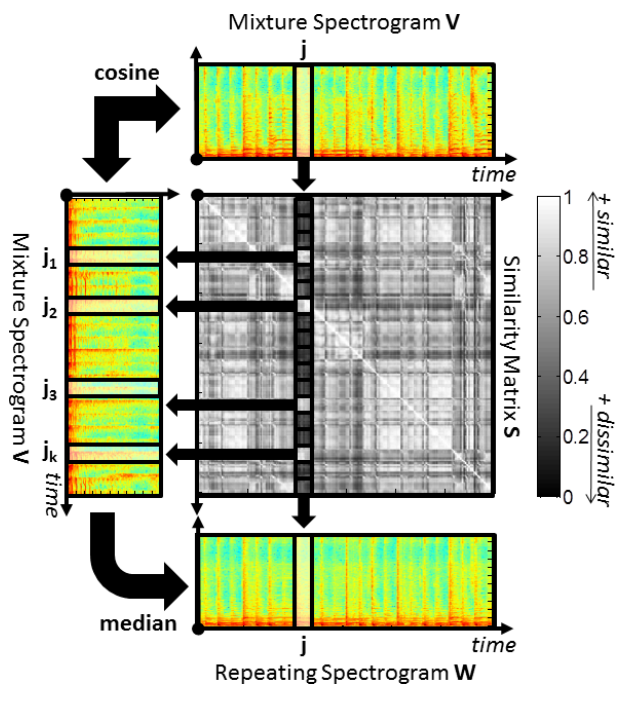


Figure Similarity Matrix

**2-3) DTW (Dynamic Time Warping) [3]**

Dynamic time warping이란, 두 시퀀스의 동기를 맞추는 알고리즘이다. 예를 들어, 같은 노래를 부르더라도 매번 똑 같은 순간마다 똑 같은 음을 내는 것은 사실상 불가능하므로, 두 시퀀스에 반드시 차이가 생기기 마련이다.

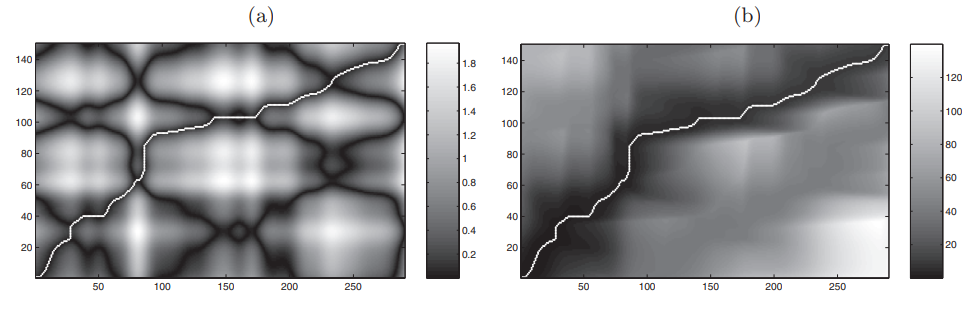
옷걸이이(가) 표시된 사진

자동 생성된 설명

Figure

길이가 다른 두 시퀀스 에 대해 cost matrix를 구성하면 아래 figure의 (a)와 같다.

어두운 부분일수록 두 시퀀스의 값이 서로 비슷하여 cost가 작은 영역이다. 하지만 이와 같이 local cost가 작다고 해서 반드시 경로가 되지 않는다. Optimal path는 누적 cost가 가장 작은 경로로 정한다.



Figure

위 figure의 b는 optimal path에 대한 accumulated cost matrix이다.

따라서 이와 같은 원리를 프로젝트에 적용하여, 두 시퀀스를 chroma CENS feature로 하고, 사용자가 부른 노래가 어떤 노래의 CENS feature와 가장 cost가 적은지 판단하여 melody matching에 적용했다. 이 프로젝트에서는 저장된 보컬 음원의 길이가 모두 다르므로, 서로 다른 음원들과 cost를 비교할 때 cost에 음원의 길이를 나눠서 normalizing하고 비교했다. 그렇게 하지 않으면 가장 짧은 음원과 matching 되는 경우가 많았기 때문이다.

1. How to Operate Project (프로젝트 실행 방법)

**3-1) vocal\_extraction.py**

프로젝트의 music/ 폴더에 음악 파일들을 저장한다. 그리고 vocal만 분리하여 chroma feature를 추출하고 싶은 구간을 song\_list.csv 파일에 미리 수작업으로 정의한다. 이 때 duration은 몇 초 정도로 짧게 한다.

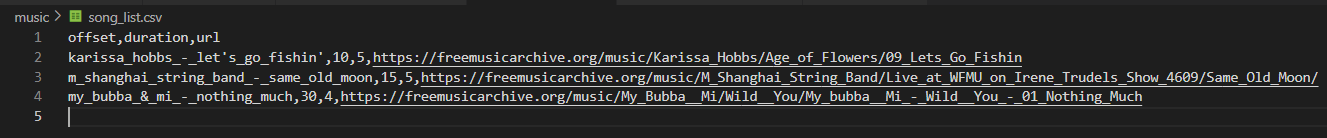


Figure csv에 노래의 어떤 부분을 chroma feature로 저장할지 미리 정의했다. 음악은 free music archive에서 ****Attribution-NonCommercial-ShareAlike 라이선스 음악을 사용했다.****

코드를 실행시켜, chroma feature를 추출하고 싶은 곡을 선택한다. 그러면 보컬만 분리된 부분의 음원이 저장되고, 그 부분의 chroma CENS feature 또한 numpy array 형태로 저장된다. (이 부분은 미리 실행되어 있고, 다시 재현해보고자 한다면 .npy 파일과 \_vocal.wav로 끝나는 파일들을 삭제하고 다시 실행해보면 된다.)

|  |  |  |
| --- | --- | --- |
|  | 🡪 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |

Figure

**3-2) record.py**

코드를 실행시키고, 터미널에 recording이라는 메시지가 뜨면, 사용자가 직접 저장된 곡 중에서 골라 멜로디를 불러야 한다. 녹음 시간은 RECORD\_SECONDS 변수를 통해 조절할 수 있다. 녹음이 종료되면, 사용자가 흥얼거린 부분의 chroma feature를 먼저 보여준다.

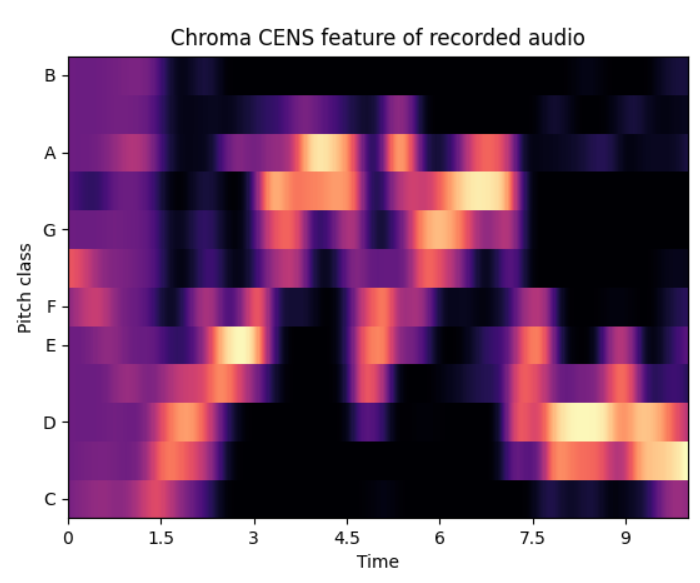


Figure 사용자가 녹음한 음성에 대한 chroma CENS feature

그리고 /music 폴더 내에 저장된 각 노래에 대해 chroma feature와 DTW cost matrix를 차례대로 보여준다.

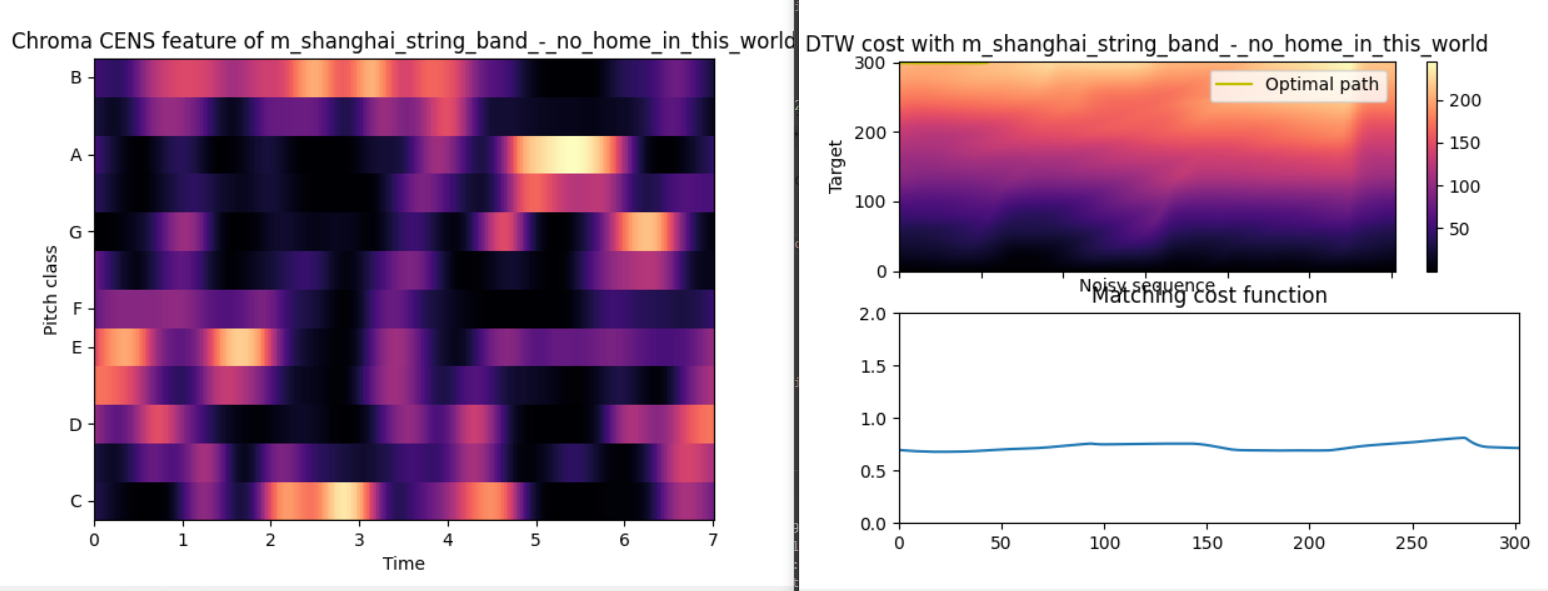


Figure 저장된 곡에서 분리된 vocal의 chroma CENS feature, 그리고 optimal DTW path와 DTW cost

최종적으로, 사용자가 녹음한 음성과 DTW path cost가 가장 작은 곡이 사용자가 부른 노래라고 판단하여 제목을 출력한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 직접 녹음한 음성과 가장 cost가 작은 노래가 무엇인지 터미널에 출력한다.

1. 한계 및 개선해야 할 점
2. Vocal extraction 후의 음성을 들어보면, 아직까지는 완벽하게 vocal과 background가 분리되지는 않았고, articulation이 발생함을 알 수 있다. 또한, 이로 인해 저장되는 chroma feature에도 영향을 주어 정확한 feature를 얻기에는 한계가 있다.
3. Chroma feature를 통해 음계 이외의 정보에 민감하지 않도록 CENS를 사용하였지만, 결국 사용자가 노래를 부를 때 찾고자 하는 음악의 음을 정확히 불러야 노래를 찾는데 더 도움이 되므로, 이런 부분이 사용자가 시스템을 이용하는데 더 어려움을 줄 수 있다. 상대음감을 가지고 있는 사용자가 사용하기에 좋은 시스템을 구축하기 위해 다른 방법을 강구해 보아야 할 것이다.

**References**

[1] M. Müller and S. Ewert, “Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features.”

[2] Z. Rafii and B. Pardo, “Music/Voice Separation Using the Similarity Matrix.,” in ISMIR, 2012, pp. 583–588.

[3] M. Müller, Information Retrieval for Music and Motion. Springer Verlag, 2007.