Accuracy comparison of humming and reference melody

Members : Advait Parmar, B. Malavika

The main objective of our project was to compare the similarity between a given reference melody and the corresponding user humming. To accomplish this objective we use many methods such as chromagram from a given time series data, dynamic time warping, optimal path detection, harmonic–percussive source separation(HPSS) and beat onset detection to calculate the accuracy between the given two audio files, frame by frame.

A general overview of what the above algorithms are. Each melody has a “main”(harmonic) component which is accompanied by other components(percussive). In our case we would like to focus only on the harmonic part of the song, as that is the one which the user will hum. Next, since the pitches of the user and song are bound to vary, we use something to mitigate this effect, namely using the chromagram vectors to compute similarity. It would also most likely be the case that user humming may not be sung perfectly frame by frame, so we need to make sure we calculate the accuracies between the correct time frames. For this we use dynamic time warping and optimal path detection to map the user humming time frame to the reference time frame.

While the program does give various parameters such as tempo detection as output itself, the final outputs of our program are the following. First is the pitch comparison accuracy, calculated once per time frame. Note, this is calculated for each time frame corresponding to the user humming, as the main objective of the program is to tell the user where they are going wrong (or right). The time frame in the reference melody with which the accuracy is compared is computed by dynamic time warping and optimal path detection. The value of this accuracy is the dot product of the corresponding chroma vectors divided by their respective norms.
Next, the program detects where there is beat onset in the reference melody and generates a graph giving the accuracy at these specific onsets, so that the user knows which beat they got right and which they didn’t.
Finally, the program outputs a 2 second snippet of both the audio files, which correspond to the sections of the respective audios, where the accuracy calculated was the least.
The data used for these can be anything ranging from a simple melody like the C-major scale to complex songs. One could even use snippets forma song and compare it to get the accuracy for only that subset.

Github link : https://github.com/Advait32/Musical-similarity-detection