ECE420 Project Proposal

Let Him Sing

Ethan Zhou, Eric Tang

{yz69, leweit2} @ illinois.edu

I. Introduction

Inspired by "South Park" season 18 episode 4, where Randy, the protagonist's dad, assumes the identity of famous female singer Lorde using auto-tune. The goal for this project is to implement an auto-tune app that takes in any songs with vocals, and audio of the user singing, separates the vocals from the songs, and remaps the user's voice into the song with temporal and pitch correction applied. The Repeating Pattern Extraction Technique (REPET) will be used for music/voice separation, and TD-PSOLA will be used for voice synthesis.

II. Overview of the algorithm

The primary algorithm used in this project would be the Repeating Pattern Extraction Technique (REPET). It achieves voice/music separation. The algorithm is discussed in detail in the paper: "REpeating Pattern Extraction Technique: A Simple Method for Music/Voice Separation" [1].

- **Step 1: Identifying repeating pattern**

  Repeating Periods could be found by using auto-correlation. Given a mixture signal x, first calculate its STFT X, using half-overlapping hamming windows of N samples, then derive the magnitude spectro gram V by taking abs(X). To emphasize the appearance of peaks of periodicity in B, V^2 is used. The overall acoustic self-similarity b of x is obtained by averaging the rows of B. As shown in the following equations.

$$B(i,j) = \frac{1}{m-j+1} \sum_{k=1}^{m-j+1} V(i,k)^2 V(i,k+j-1)^2$$

$$b(j) = \frac{1}{n}\sum_{i=1}^{n} B(i,j)$$

$$then : b(j) = \frac{b(j)}{b(1)}$$

for $i = 1 \ldots n$(frequency)where $n = \frac{N}{2} + 1$

for $j = 1 \ldots m$(lag) where m = number of time frames

A simple procedure for automatically estimating the repeating period p. The idea is to find which period in the beat spectrum has the highest mean accumulated energy over its integer multiples. For each possible period j in b, we check if its integer multiples i correspond to the highest peaks in their respective neighborhoods, where Delta is a variable distance parameter, the function of j. At the end, we minus the mean of the given neighborhood to filter any possible noisy background.

```
**for** each possible period j in the first 1/3 of b **do**
    $\Delta \leftarrow [3j/4], I \leftarrow 0$
    **for** each possible integer multiple i of j in b **do**
        h $\leftarrow$argmax$b(k)(k \in [i - \Delta, i + \Delta])$
        **if** h = $argmax$ b(k) **then**
            I $\leftarrow I + b(h) - avg(b(k))$
        **end if**
    **end for**
    J(i) $\leftarrow I/[l/j]$
**end for**
p $\leftarrow$argmax$_j$ J(j)
```

- **Step 2: Repeating Segment Modeling**

  Once the repeating period p is estimated from the beat spectrum b, we evenly time-segment the spectrogram V in to r segments of length p. Define the repeating segment model S as the element wise median of the r segments.

$$S(i,l) = \underset{k=1,,,r}{\text{median}}\{V(i, l + (k-1))\}$$

for $i = 1 \ldots n$(frequency) and $l = 1 \ldots p$(time) define p = period length, r = number of segments

- **Step 3: Repeating Patterns Extraction**

  Once the repeating segment model S is calculated, it can be used to derive a repeating spectrogram model W, by taking the element wise minimum between S and each of the r segments of the spectrogram V.

$$W(i, l + (k-1)p) = \min S(i, l), V(i, l + (k-1)p)$$

for $i = 1 \ldots n, l = 1 \ldots p$ and $k = 1 \ldots r$

  Once the repeating spectrogram model W is calculated, a soft time-drequency mask M could be derived by normalizing W by V element-wise. The idea is time-frequency bins that are likely to repeat at period p in V will have values near 1 in M and will be weighted toward the repeating background and time-frequency bins that are not likely to repeat at period p in V would have values near 0 in M .

$$M(i, j) = \frac{W(i, j)}{V(i, j)}$$

for i =1 . . . n (frequency) and j = 1 . . . m (time)

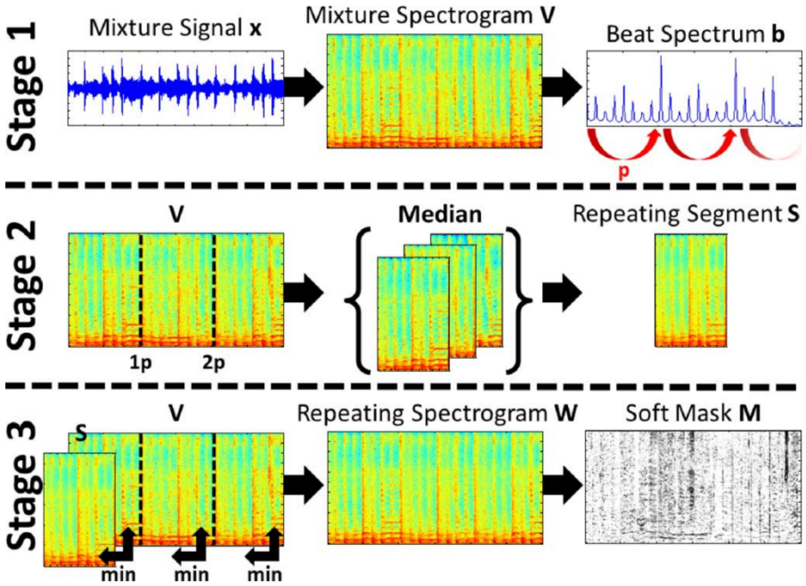

Fig. 1. Overview of the Algorithm

- **Step 4: Detecting Voiced components**

  We will determine whether a period of both user input and the separated voice track. We will use pauses in voiced components to segment the entire soundtrack into sentences, which then allows us to apply temporal correction by directly stretching/compressing user input. Assuming the correction needed is not very large, thus the distortion of sounds produced by the vocal track is not significant.

- **Step 5: Pitch detection**

  we first segment each sentence into 40ms chunks, then apply pitch detection via auto-correlation. This will give us the spectrogram of the original voice in the song, and user input.

- **Step 7: Pitch Synthesis**

  We will map the user input to the original soundtrack by TD-PSOLA

III. Plan for testing and validation

- To demonstrate this algorithm work, we can sing into the microphone and listen to the output
- The inputs we will be using is a pre-existing song, and the microphone on android device to record user singing.
- The output will be the final pitch/temporal corrected song. The metric would be the pitch correction accuracy, which would be measured by the deviation of the target and corrected spectrogram.
- We will start with shorter and simpler soundtracks to test the voice separation effectiveness and work our way to more complex and longer soundtracks and test the accuracy of the entire system.

IV. Contribution

- Ethan Zhou: task A, C, E, F
- Eric Tang: task B, D, E, F

Contents	**Number**
task A	Implement the voice/music separation algorithm in Python
task B	Implement the remaining elements in Python
task C	Implement the voice/music separation algorithm in C++
task D	Implement the remaining elements in C++
task E	Android integration of functions and UI
task F	Project video for extra credit (optional)

TABLE I

TASKS AND ALLOCATIONS

REFERENCES

[1] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 1, pp. 73–84, 2012.