Audio Sources Separation by Clustering Techniques

Wei Fang

[Abstract]

This project tries to present a method to distinguish different components in a piece of audio signal without prior knowledge of instrument types and number of instruments. Spectrum analysis plays a major role in this method. Clustering techniques are applied to eliminate duplicated audio sources.

[Background]

Audio sources, including human vocal cords and musical instruments usually are based on a harmonic oscillator that produces vibrations on a series of integer multiples of a base frequency. The base frequency defines the pitch of the sound produced. Due to different characteristics of these sources, even if we have 2 sounds from 2 different sources whose base frequencies are the same, their power distributions over these harmonics are different. We human ears can identify the differences in the distributions and tell that the instruments produce different timbres. To us, timbres are signatures of sound sources. However, these signatures are not that straightforward to computers. Since these signatures are described in the spectrum domain, my starting point to tackle this problem is from the spectrum of audio signals.

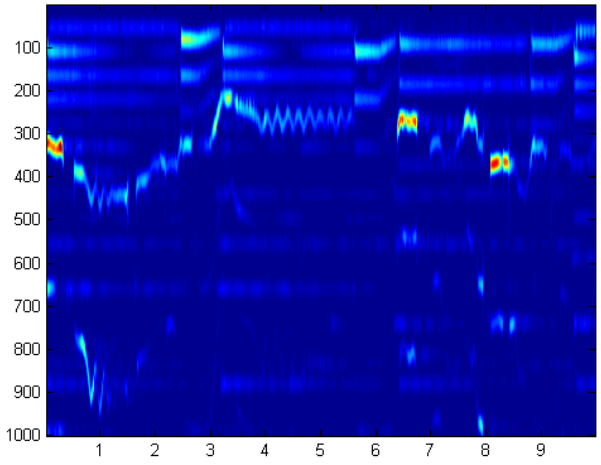

Figure 1: Spectrum calculated from a segment of a song

[Proposed Method]

1. Onset detection

Timbres are most distinguishable when the notes are just set. For example, when the little hammer strikes a string inside the piano, or when human produces a consonant before a vowel, these make the sound much more distinguishable. Thus, onset detection is my first step here.

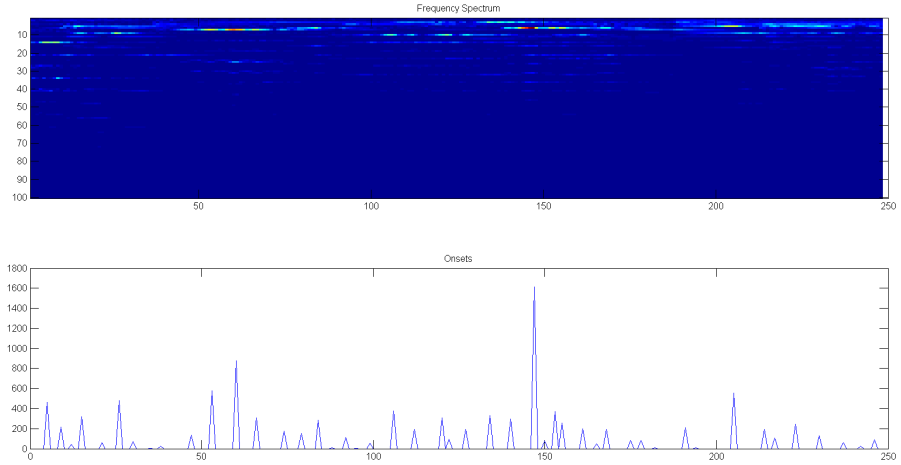

Figure 2: Detected onsets and corresponding spectrums

There are many sophisticated onset detection methods. Here I incorporated a simple one: capturing sharp increases in spectral energy.

2. Generalize models from the above onset positions

In this step, I want to obtain models of audio sources. These models are described by amplitudes and phases on higher order harmonics referencing to the base frequency. For each identified sound source, this step will fill up a table like Table 1.

	Base freq.	1st Harmonic	2nd Harmonic	3rd Harmonic	…
Amplitude	1	Amp1	Amp2	Amp3	…
Phase	0	Phs1	Phs2	Phs3	…

Table 1: Model generalization

2.1. Perform short-time Fourier transform

In order to calculate spectrum of the input signal at a given time spot, short-time Fourier transform is carried out. It is done by taking out a short segment (for audio signal sampled at 44100Hz, 2205 samples would be appropriate since this will allow an identification of frequency as low as 20Hz, the lower limit of human hearing) from the original signal at the given time, windowing it by a Gaussian window (so that consecutive measurements will be smoother), performing zero padding at both left and right side tails of the signal (so that it offers better frequency resolution after the Fourier transform), and performing Fourier transform.

2.2. Take the maximum frequency response; find out its harmonic resonances

2.3. Normalize the amplitudes of the resonances. Note them down as a basis

This is essentially filling up the form Table 1 for the current signal model being processed.

2.4. Subtract the just-obtained basis from the current set of frequencies.

2.5. Do 2.2 to 2.4 until energy left in the Fourier transform is reasonably small.

The effect of these steps is illustrated in Figure 3.

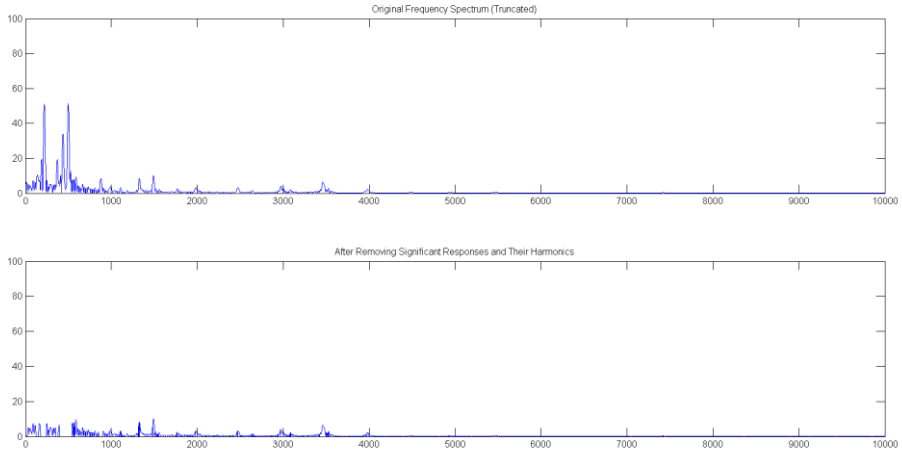

Figure 3: Signal before and after obtaining significant bases

3. Combine bases that probably come from the same instrument

Since the model generalization is performed on every onset position, there should be plenty of similar models in the set of models obtained. These models can be represented as points in C^n space, where n represents the number of harmonics being considered in the models. To simplify the problem, I only took the first 5 harmonics (which are usually more significant than the others), and only consider their amplitudes. This reduces the problem into a R^5 space clustering problem.

Different from the clustering problems described in class, this is a clustering problem with an unknown number of clusters.

The first method I considered is mean-shift clustering: perform gradient ascent to move the mean to the center of one cluster, subtract points belonging to that cluster and perform gradient ascent again. But later on, I discovered that the points in R^5 are quite sparse. Under this condition, the problem might be easier solved by judging the distances between points. A distance map showing Euclidean distances between points is generated.

3.1. Cluster the bases generated in 2 by the distance map

3.2. Replace bases in a cluster with the mean of the cluster

In these steps, a metric of being similar is defined as the Euclidean distance between points. Similar models are combined. The effect of this step is shown in Figure 4 and Figure 5.

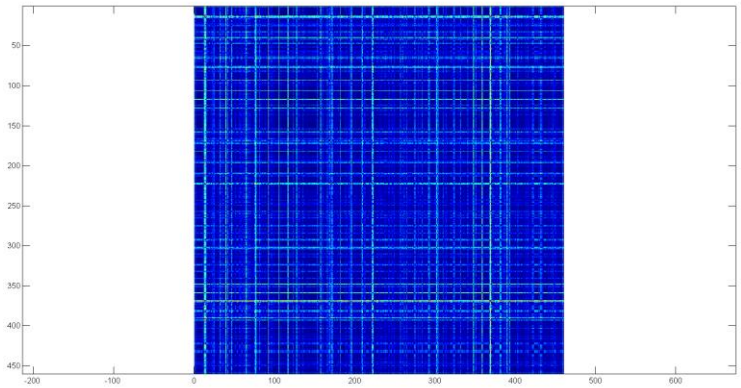

Figure 4: Distances between bases before combining similar bases

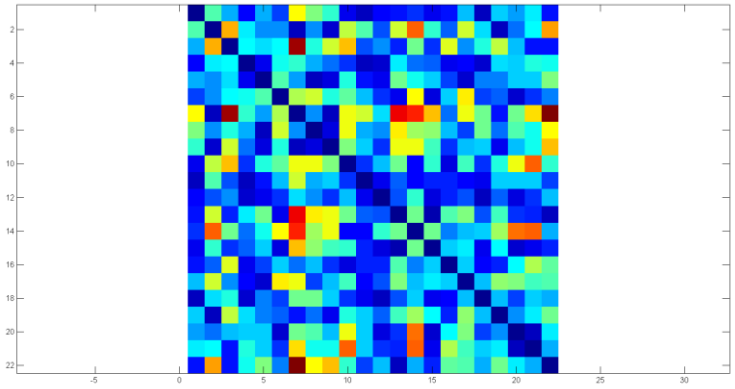

Figure 5: Distances between bases after combining similar bases

4. Recover signal from the bases representation

In this step, I will generate audio signals with specific models being removed. Short-time Fourier transform is applied again on the signal. This time the transform is performed periodically, from the beginning through the end of the input signal.

4.1. Identify frequency components that are similar to the model. Remove them.

This step is again done by the distance map based clustering. Several models are identified from the segment of input signal, and they are being compared with our selected model. If they are similar, the frequency components are removed from this segment of input signal

4.2. Perform inverse short-time Fourier transform to get the time domain signal back.

In this step, the segments of signal are transformed back into time domain signal. Overlapping parts of the segments are properly weighted so that the output signal is a smooth one.

[Experiments and Results]

Experiments showed that the generalized model can properly represent the components in the input signal. Figure 6, Figure 7 and Figure 8 showed models being generated from different types of input signals, which matches the fact.

However, the model removal is not that effective. Though the models are already clustered, they still seem not powerful enough to express a sound source completely.

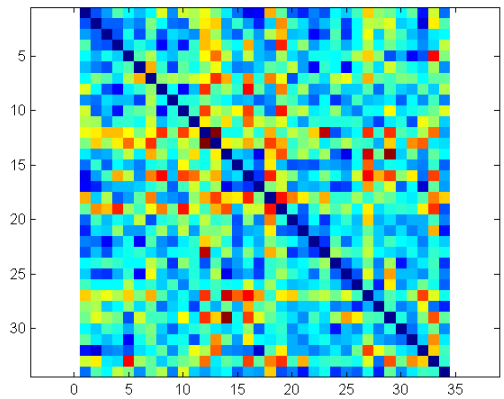

Figure 6: Models generated from a piano-violin piece of music

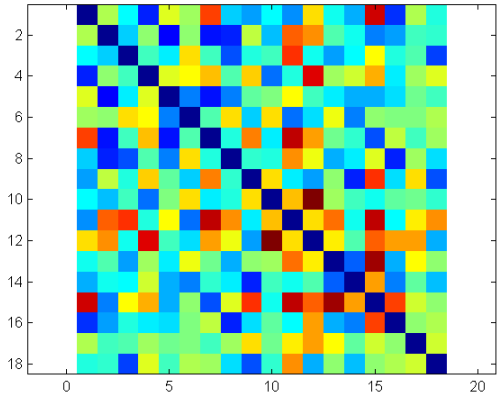

Figure 7: Models generated from a solo piano piece of music

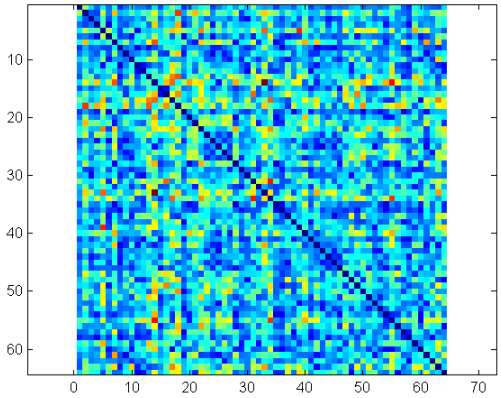

Figure 8: Models generated from a song from a female singer with accompany by a range of instruments

[Future Work]

A better metric to combine models should be developed.

The current algorithm is well adapted to audio sources that put most of their power on the base frequency. However, there are instruments that put their peak of energy on one of their higher order harmonics. There should be ways to deal with this.