Isolating Musical Instruments Using the Fast Fourier Transform

Hamza Golubovic

Abstract

In this project, `Clip.mat` is analyzed, which is a sound clip of the opening ballad in "I'm Shipping Up To Boston," by Dropkick Murphys. By using the Gabor Transform, a spectrogram of frequencies throughout the clip is created. Based on these frequencies, the thresholds of bass versus guitar frequencies can be distinguished. Using these thresholds, the Fast Fourier Transform can be run on the clip, after which the frequencies outside of these thresholds can be eliminated. In the case of extracting the baseline/drumbeat, the threshold is between 50 Hz and 200 Hz. The inverse Fourier Transform is used to revert back to the time domain, giving us a soundclip with an isolated baseline/drumbeat. The same process is repeated for the guitar and higher pitched instruments.

1 Introduction

Musical notes can be distinguished based on their frequencies. The notes a specific instrument can play varies in pitch. The middle C note refers to the middle of an instruments frequency range. For example, a tuba's middle C note will be lower in frequency when compared with a violin's middle C. Hence, tubas can play notes that are lower in frequency, giving them a bass like sound, whereas a violin is able to extend on the spectrum of higher frequency notes. Being able to distinguish instruments based on their frequency ranges, we will be able to isolate instruments from a sound clip.

In the remainder of this report, essential theoretical background is outlined for the numerical analysis of the sound clip. Specifically, the derivation of the Gabor Transform and the Discrete Gabor Transform. The analysis and results are presented towards the end of this report.

2 Theoretical Background

In this section, we will derive the Gabor Transform and the Discrete Gabor Transform.

2.1 The Gabor Transform

The Fourier Transform is unable to localize a signal in frequency and time domains simultaneously, which proves to be one of the major drawbacks in this method. However, the Gabor Transform aims to solve this issue by utilizing the kernel:

$$g_{t,w}(\tau) = e^{i\omega\tau} g(\tau - t)$$

The Gabor transform is then defined as

$$\tilde{f}_g(t, \omega) = \int_{-\infty}^{\infty} f(\tau) \bar{g}(\tau - t) e^{-i\omega\tau} d\tau$$

Here, $\bar{g}$ is the complex conjugate of g, and we observe that the kernel is utilized as a time filter for localizing the signal in some window of time, and integrating with respect to τ allows the filter to traverse the whole signal. Generally, g is taken to be real and symmetric and $||g(t)|| = 1$. Thus, the Gabor transform now becomes

$$\tilde{f}_g(t, \omega) = \int_{-\infty}^{\infty} f(\tau) g(\tau - t) e^{-i\omega\tau} d\tau$$

The inverse Gabor transform is given by

$$f(\tau) = \frac{1}{2\pi}\frac{1}{||g||^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{f}_g(t,\omega)g(\tau - t)e^{i\omega\tau}d\omega dt$$

2.2 The Discrete Gabor Transform

In order to implement the Gabor Transform on a computer, the time and frequency domains must be discretized. We define the discretization as

$$\nu = m\omega_0$$
$$\tau = nt_0$$

with $\omega_0, t_0 > 0$ having constant value and $m, n \in \mathbb{Z}$. The discretization of the kernel $g_{t,\omega}$ is now given by

$$g_{m,n}(t) = e^{i2\pi m\omega_0 t}g(t - nt_0)$$

The discrete Gabor Transform is therefore

$$\tilde{f}(m,n) = \int_{-\infty}^{\infty} f(t)g_{m,n}(t)dt$$

3 Numerical Methods

Using the theoretical background from the previous section, we analyze the opening ballad of "I'm Shipping Up To Boston," which is saved in `CP2.SoundClip.mat`.

We begin by segmenting the entire clip into 4 sections (S_1, S_2, S_3, S_4), as analyzing the entire clip S will be too large for MATLAB to handle. We continue by discretizing the time domain τ and the frequency domain k. We also Fast Fourier Transform k and shift the frequencies.

For each of $S_1, ..., S_4$, we construct for loops iterating over the duration of each of these intervals. For this explanation, we consider the case for S_1. At each step j, we construct a window function with width $a = 400$ with moving center at $\tau(j)$:

`g1` = exp(-a·(t-τ(j)).^2)

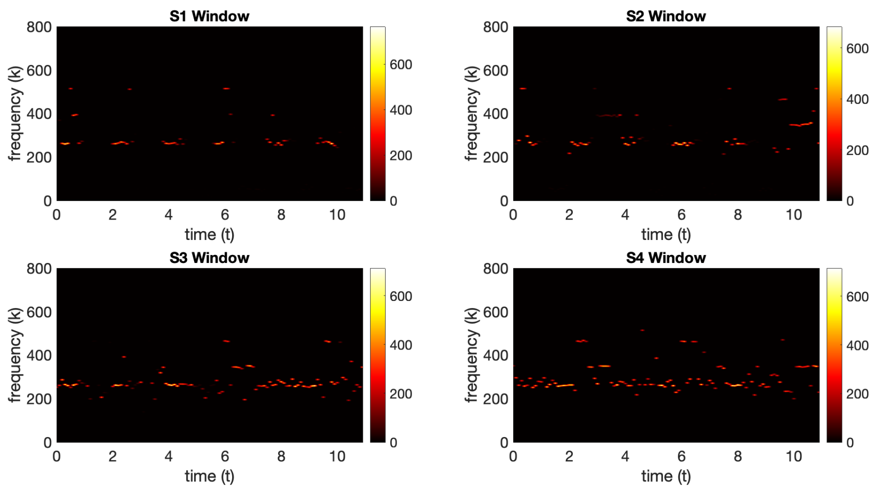
We then define the windowed signal as `Sg1` = `g1`·S1, which we Fast Fourier Trasnform to obtain `Sgt1`. We then find the peak frequency (in absolute value, since we are dealing with real signals, meaning negative and positive frequencies should have no distinction). The index of this frequency is stored as variable `ind1`, and we construct a Gaussian filter with width $1/L$ centered at k(`ind1`) and apply it to the Gabor transformed function `Sgt1`. Lastly, we Fast Fourier Transform the absolute value (only considering real signals) of the filtered Gabor transformed function using `fftshift`. We save the value in the vector `Sgt_spec`, which defines the spectrogram for S_1. This process is then repeated for S_2, S_3, S_4.

After analyzing the spectrogram created, we observe that the thresholds for the low and high frequency instruments. For the baseline, we Fast Fourier Transform the sound clip data S to obtain $fft(s)$. Then, a for loop iterating over the size of S is used to find at what instances `abs(k(i))>200` or `abs(k(i))<50`. For such frequencies, we set `fft(s)(i)=0`. The modified data is then reversed back to the time domain using the Inverse Fast Fourier Transform. The same is done for the guitar and higher pitched instruments, but for i such that frequencies `abs(k(i))>700` or `abs(k(i))<300`, we set `fft(s)(i)=0`.

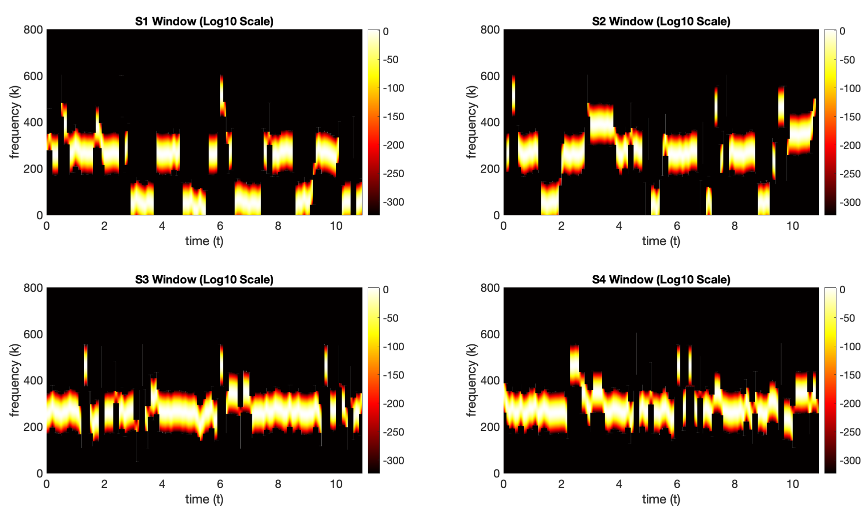
4 Results

The following are spectrogram produced for each window S_1, S_2, S_3, S_4, with the bottom four plots more clearly demonstrating the layers of the different frequencies by using a logarithmic scale plot.

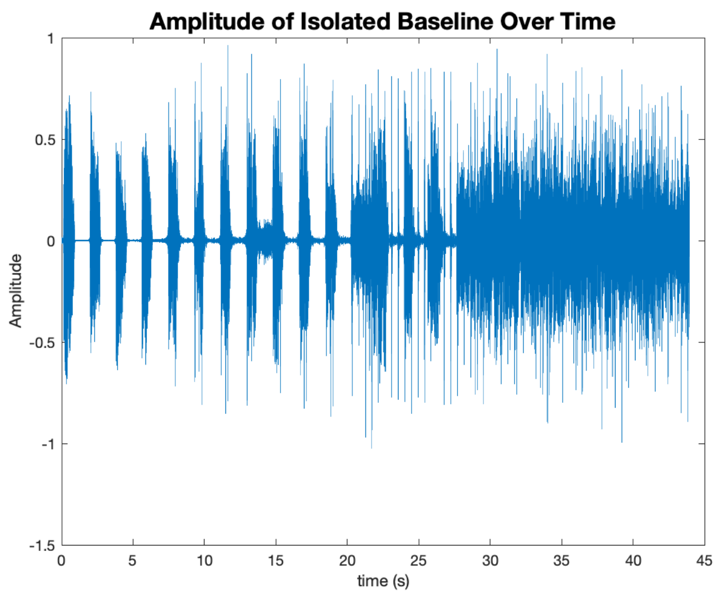
Spectrogram of the Sound Clip Durring Different Windows



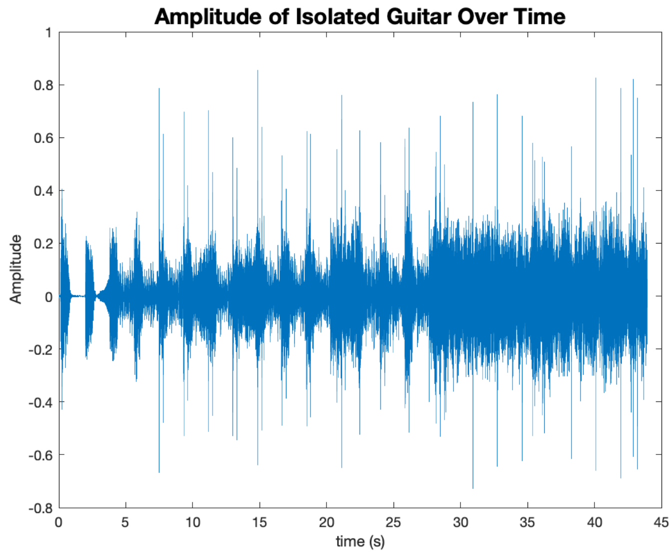
Spectrogram of the Sound Clip Durring Different Windows (Logorithmic Scale)



From the above figures, using the logarithmically scaled plot, we observe that there are clear layers that distinguish instruments of varying frequency being played. The goal of this project is to isolate the bass and guitar frequencies. Therefore, to isolate the lowest layer of frequencies (in the range of roughly 50 Hz to 200 Hz according to the spectrogram) would isolate the baseline. Similarly, isolating the middle to upper frequency layer (in the range of 300 Hz to 700 Hz) would isolate the guitar frequencies and other similar pitched instruments.



After isolating the baseline from the rest of the sound clip, we obtain the plot above, where the amplitude is plotted against time. The instances in time when a bass is heard can easily be visualized in the first half of the clip, where they are played sparsely.



After isolating the guitar and other high frequency instruments from the rest of the sound clip, we obtain the plot above, where the amplitude is plotted against time. It is observed that unlike the bass, the guitar is not in rhythmic intervals. This agrees with what is heard in the song, since a guitar has a more continuous sound. We see that the amplitude range is smaller than the one observed with the baseline.

5 Conclusion

In this project, we ran the Gabor Transform on "I'm Shipping Up to Boston," and obtained a spectrogram of the frequencies at varying times throughout the sound clip. From the spectrogram, we see that the low frequency instruments are in the range of 50 Hz to 200 Hz, while high frequency instruments played between 300 Hz and 700 Hz. After isolating the bass and then the guitar from the original clip, we then verified our results by playing them on our speakers. Plotting the two results' amplitudes against time, we observe that the amplitude for the baseline spanned a larger interval than that of the guitar. Generally, frequency and amplitude are inversely related, which our results follow.