Music Analysis

Josiah Boning
TJHSST Computer Systems Lab, 2007-2008

Abstract

Although music is one of the most universal aspects of human culture, it is very difficult to define. Most definitions of music have been dependent on attributes such as rhythm, melody, and harmony, which are extremely subjective, so the ability to identify music has been limited to humans. This project explores statistical and signal processing techniques for computational analysis of music and provides a basic framework for more advanced music recognition and identification efforts.

Background

Computers have already been used to perform analysis of music. Research has shown that different genres of music can be distinguished by fractal dimension and that machine learning techniques could successfully identify musical genres[2][1]. Other research has attempted to deconstruct music in terms of rhythmic and melodic patterns, and even looked at writing software to generate music conforming to such patterns[3]. However, each instrument has a different sound quality, and composers write music with these timbral differences in mind. Simply analyzing the notes on sheet music precludes the use of these differences in the analysis. Audio recordings, in contrast, allow analysis of exactly what the composer intended his audience to hear.

Methods & Concepts

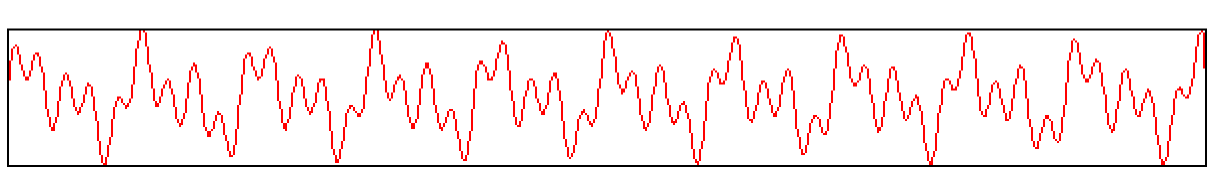
What humans perceive as sound is the variation in pressure of waves passing through the air. Computers store audio data as a sequence of discrete samples of the pressure waveform. According to the Nyquist-Shannon sampling theorem, as long as the waveform contains no frequencies equal to or higher than 1/2 of the sampling frequency, this representation loses no information, and the original waveform can be reconstructed precisely. However, while this representation is useful for describing and reproducing the original sound wave, it does not make other information about the content of the wave readily apparent. Therefore, various techniques must be applied to extract more interesting information from the waveform.

Spectral Decomposition

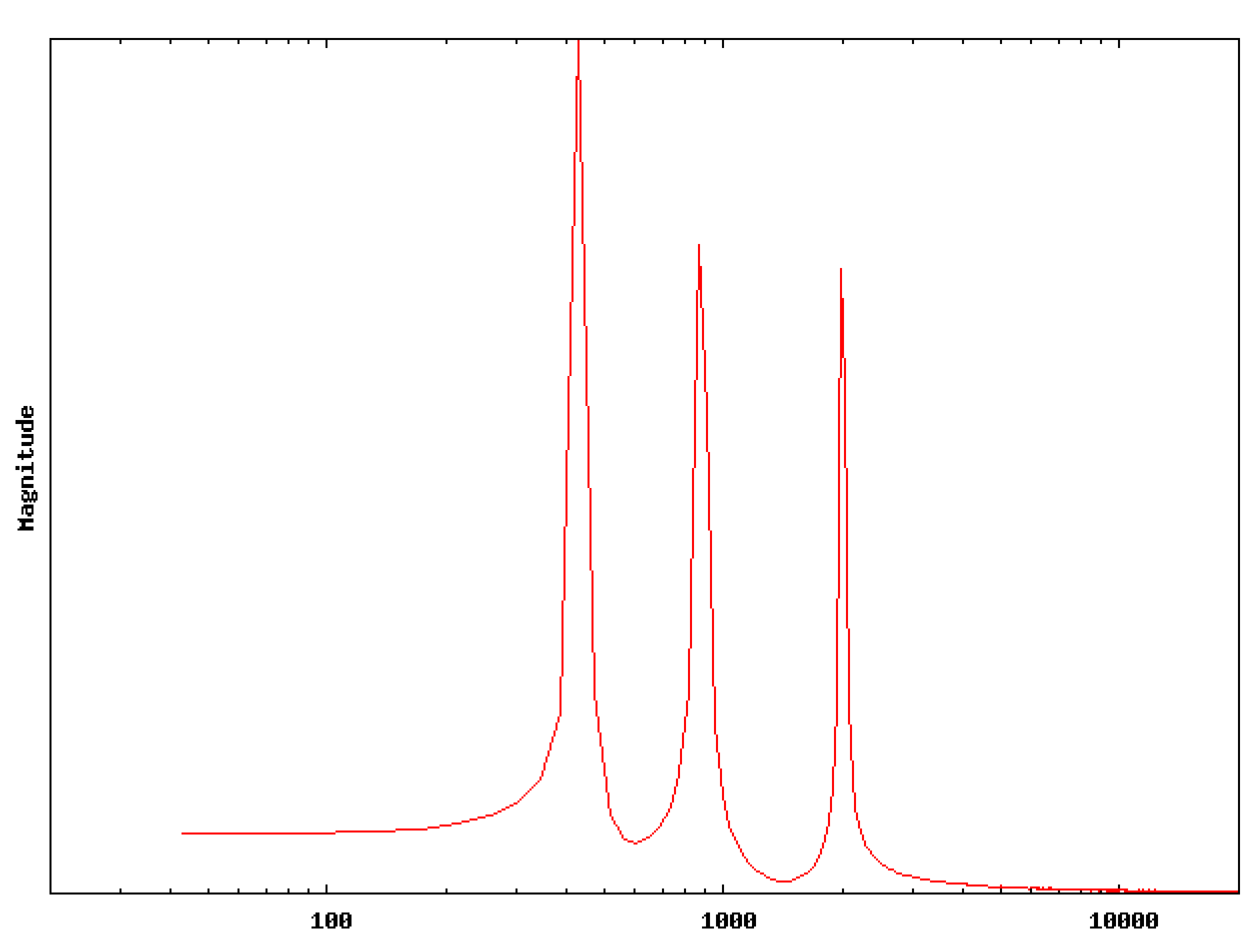
Spectral decomposition transforms audio data from the time domain to the frequency domain using a Fourier transform. The Fourier transform decomposes the audio waveform into sinusoids of varying frequencies and tells how much of each wave is present. The Fourier transform is defined as:

$$F(v)=\int_{-\infty}^{\infty} f(t)\, e^{-2\, pi\, i v t}\, dt$$

Instead of having the sound as *f*(*t*), a function of time, we have it as *F*(*v*), a function of frequency.



The waveform above is clearly some composition of sine waves. Performing a Fourier transform, however, yields much more information about its composition. The frequency spectrum below shows that the wave is a combination of sine waves at 440, 880, and 2000 Hz.



Fractal Dimension

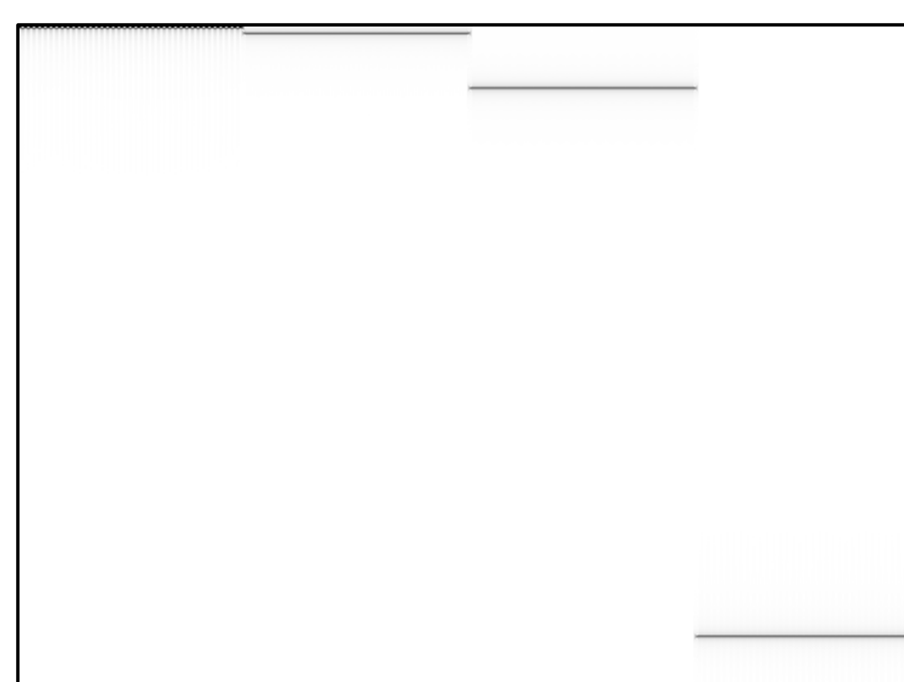
The equations below are evaluated numerically to yield the fractal dimension of the audio waveform. Since different genres of music are distinguishable by their fractal dimensions, it is reasonable to suspect that music itself might be distinguishable by its fractal dimension[2]. The equations are evaluated over discrete audio data, so their accuracy will be increased by the use of cubic splines to interpolate between data points and allow smaller differential values when doing numerical integration.

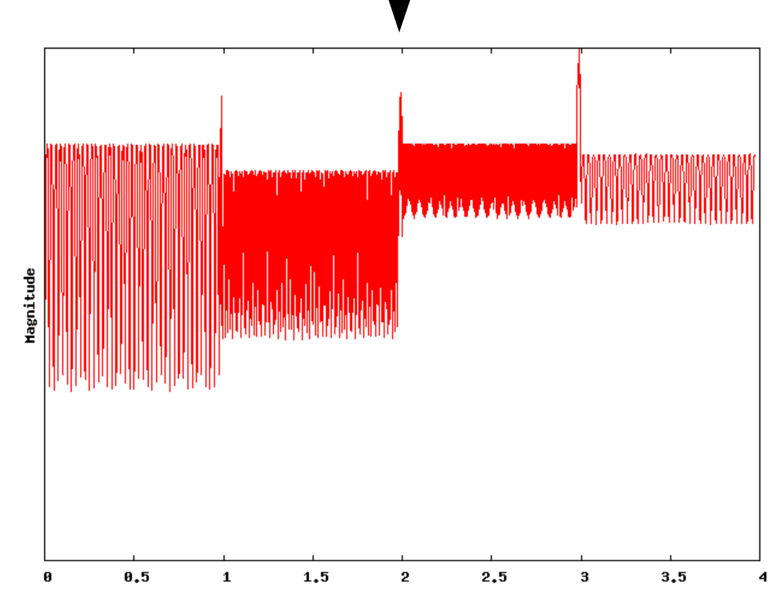
$$\lim_{\tau\to 0}\left(2-\frac{\log\left(\frac{1}{b-a}\int_a^b \left|\max_{|x-t|<\tau}(f(t))-\min_{|x-t|<\tau}(f(t))\right|dx\right)}{\log\tau}\right)$$

$$\lim_{\tau\to 0}\left(2-\frac{\log\left(\frac{1}{b-a}\int_{x=a}^{x=b}\left[\frac{1}{\tau^2}\int_{t_1=0}^{\tau}\int_{t_2=0}^{\tau}\left|f(x+t_1)-f(x-t_2)\right|^{\alpha}dt_1\,dt_2\right]^{1/\alpha}dx\right)}{\log\tau}\right)$$

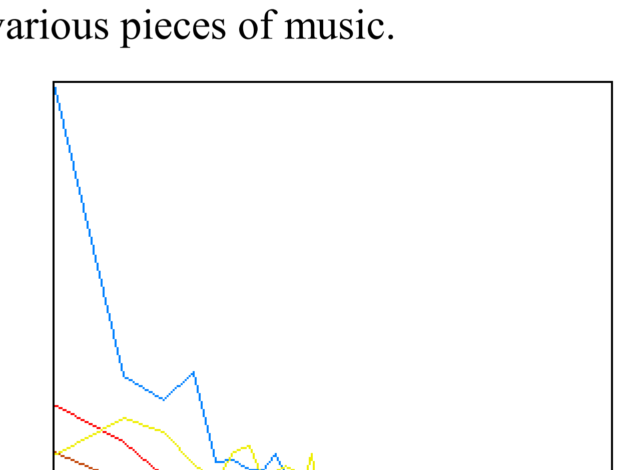
Spectrograms and Data Aggregation

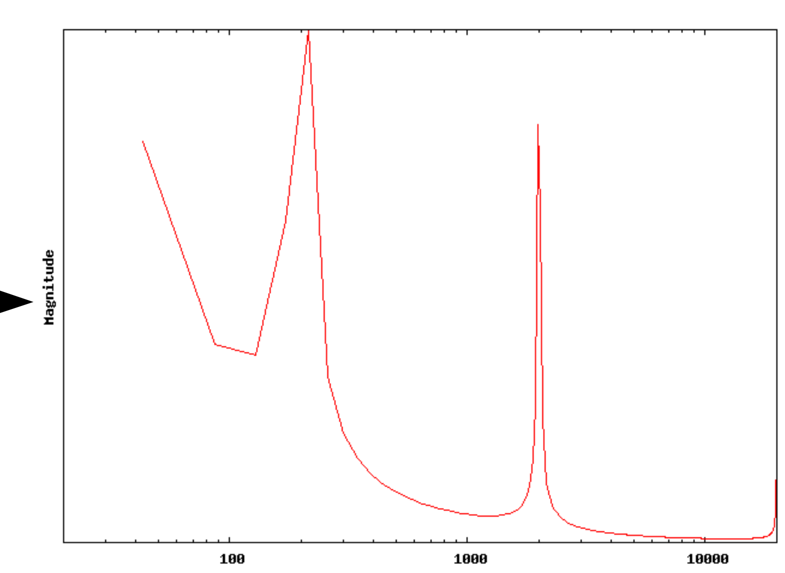
Calculating many Fourier transforms in sequence results in a spectrogram, which provides a graphic representation of the audio data's frequency distribution over time. For example, this spectrogram clearly shows four different tones in sequence:



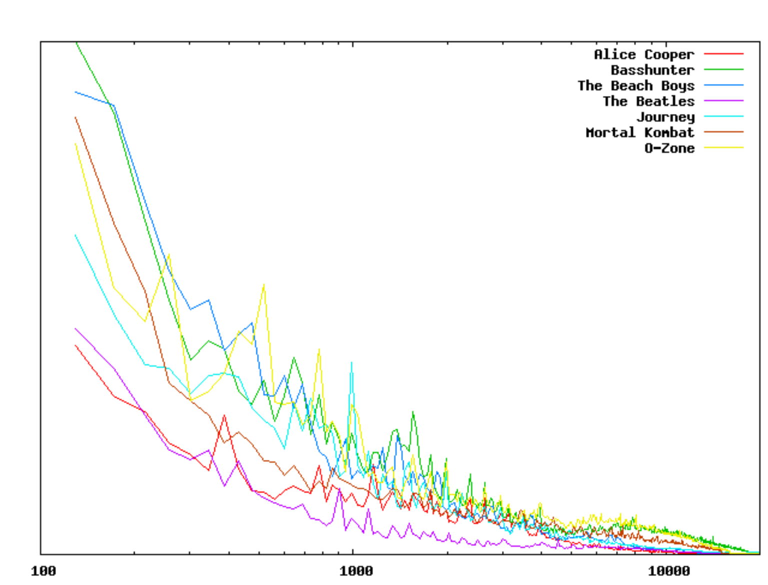


Summing the spectrogram vertically yields a rough measurement of the volume at each point. This is not particularly useful by itself, and performing a Fourier transform on this aggregate data also yielded no useful trends. Below are the results of this Fourier transform on various pieces of music.

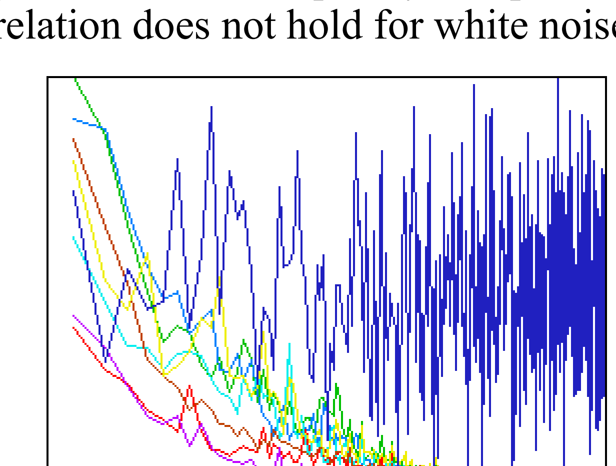




Summing the spectrogram horizontally yields the total frequency composition of the audio data, the aggregate Fourier transform of the entire audio sample over time. Performing this aggregation on spectrograms of music yields an interesting result:



All of the songs have a strong inverse correlation between frequency and the magnitude of that frequency component. This correlation does not hold for white noise:



Works Cited

[1] Basili, Roberto, Alfredo Serafini, and Armando Stellato. 2004. "Classification of Musical Genre: A Machine Learning Approach." Presented at the 5th International Conference on Music Information Retrieval.

[2] Bigerelle, M., and A. Iost. 2000. "Fractal Dimension and Classification of Music." *Chaos, Solitons & Fractals*. 11(14):2179-92.

[3] Leach, Jeremy, and John Fitch. 1995. "Nature, Music, and Algorithmic Composition." *Computer Music Journal*. 19(2):22-23.