TJHSST Senior Research Project
Computer Music Analysis
Computer Systems Lab, 2007-2008

Josiah Boning

April 4, 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.

Keywords: music analysis

1 Introduction

1.1 Purpose

The purpose of this project is to create a framework of statistical, signal processing, and machine learning components for use with audio data. This combination of computational tools has potential applications in many fields, as sound recognition is useful in contexts ranging from aircraft identification to the human-computer interaction fields of speech recognition and synthesis[1][2]. This project is intended to provide a tool for further research in any such area dealing with computer audio processing. However, while the capabilities this project provides are applicable to a wide variety of problems, this project is targeted at recognition of music and discrimination between music and non-music.

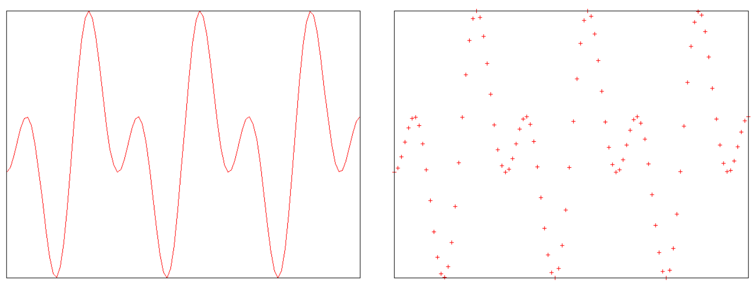
2 Background

Computers have already been used to perform analysis of music. In 1999, Bigerelle and Iost determined that different genres of music could be distinguished by fractal dimension, and in 2004, Basili et al. showed that machine learning techniques could successfully identify musical genres [3][4]. 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[5]. However, as Bigerelle and Iost point out, 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.

3 Methods and Concepts

3.1 Audio Analysis

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.



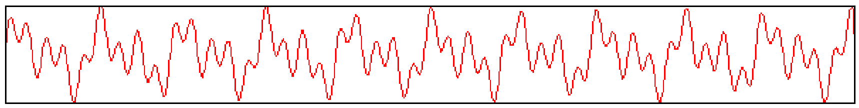
However, while this representation is useful for describing and reproducing the original sound wave, it does not make other information about the content of the wave readily apparent. Therefore, various techniques must be applied to extract more interesting information from the waveform.

3.1.1 Spectral Decomposition

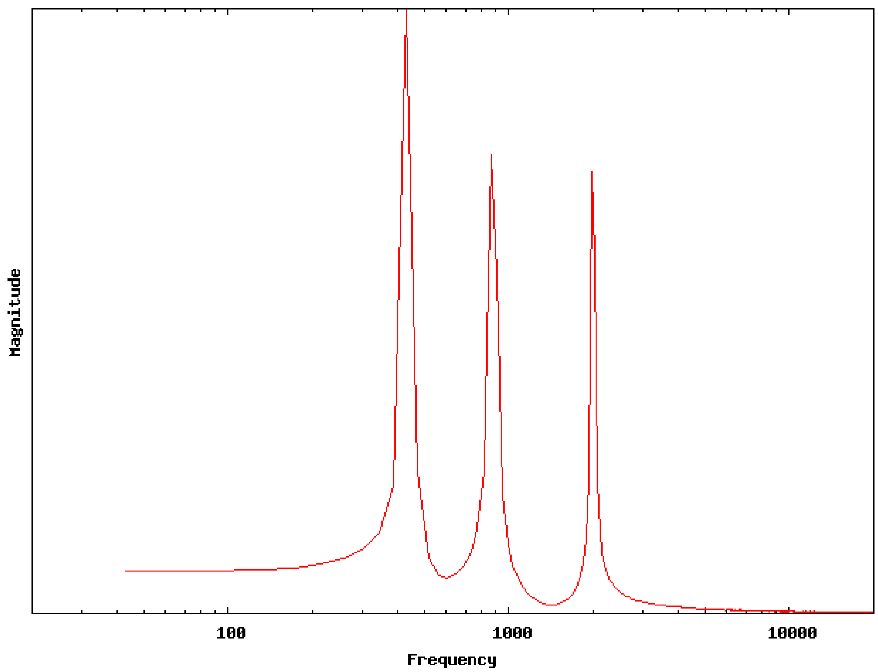
Spectral decomposition transforms audio data from the time domain to the frequency domain using a Fourier transform. The Fourier transform decomposes the audio waveform into sinusoids of varying frequencies and tells how much of each wave is present. The Fourier transform F of a function f is defined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

$$F(v) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i v t} dt \qquad (1)$$

Instead of having the audio data as a function of time, $f(t)$, we have a function of frequency, $F(v)$. Take this waveform as an example:

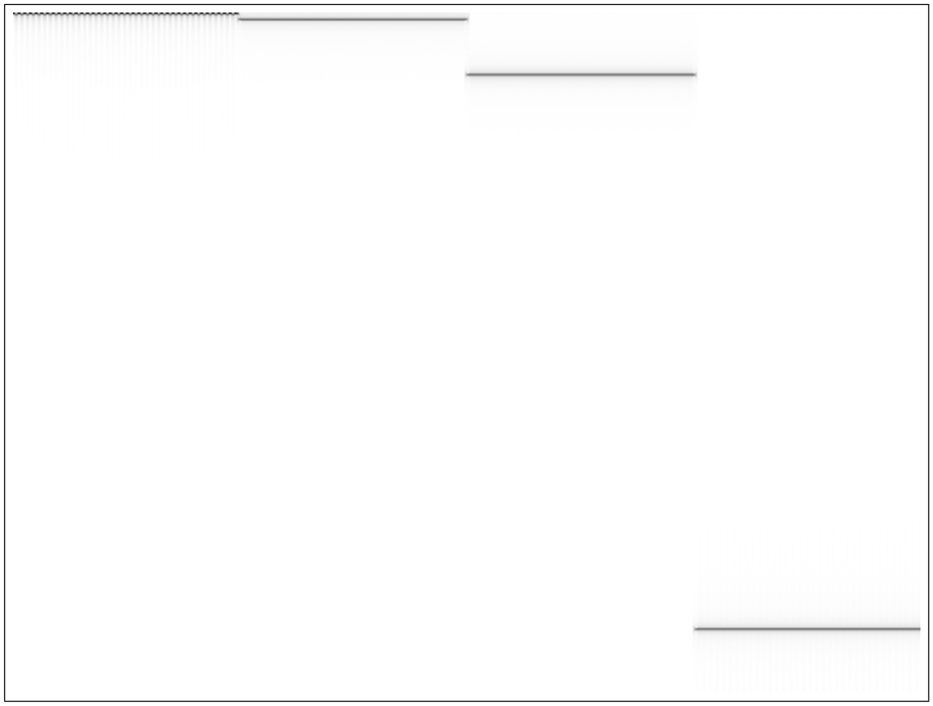


Clearly it is some composition of sine waves. Performing a Fourier transform, however, yields more specific information:



This spectral decomposition correctly shows that the waveform is composed of sine waves at 440, 880, and 2000 Hz.

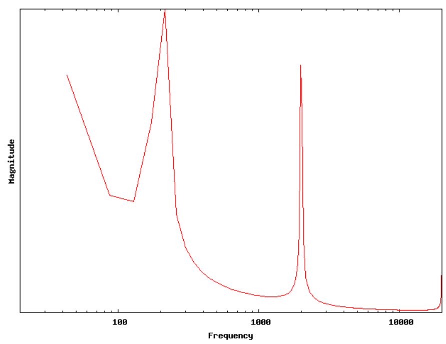
A spectrogram can be produced by performing Fourier transformations on many short chunks of audio data sequentially. Since the spectrogram is constructed from Fourier transforms of different sections of the audio data, it shows graphically how the frequency composition of the sound changes over time. Four example, a recording of four different tones in sequence yields this spectrogram:

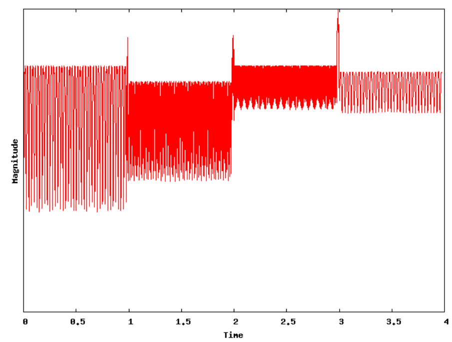


The tones and changes in tone are clearly visible in the spectrogram.

3.1.2 Aggregated Spectral Data

Spectrograms are good at providing a visual overview of the composition of a sequence of sounds. However, it is easier to analyze one-dimensional data. Spectrograms can be aggregated into one-dimensional data in two ways. One option is to total each frequency over the course of the audio; on the spectrogram above, this is the equivalent of summing each row. This yields a total frequency composition for the waveform. Another option is to total the frequency data in each chunk. The data yielded by this is roughly the volume over the course of the audio data. These aggregations are illustrated in the graphs below.





3.1.3 Fractal Dimension

Bigerelle and Iost demonstrated that the fractal dimension of a waveform is linked to its musical content[3]. They used these two formulae to calclulate fractal dimension:

The Variation method:

$$\Delta = \lim_{\tau \to 0} \left(2 - \frac{\log \left(\frac{1}{b-a} \int_a^b \left| max(f(t))_{|x-t|<\tau} - min(f(t))_{|x-t|<\tau} \right| dx \right)}{\log \tau} \right) \quad (2)$$

The ANAM method:

$$\Delta = \lim_{\tau \to 0} \left(2 - \frac{\log \left(\frac{1}{b-a} \int_{x=a}^{x=b} \left[\frac{1}{\tau^2} \int_{t_1=0}^{\tau} \int_{t_2=0}^{\tau} |f(x+t_1) - f(x-t_2)|^{\alpha} \right]^{1/\alpha} dx \right)}{\log \tau} \right) \quad (3)$$

3.2 Machine Learning

A neural network is a computational model for machine learning based on the biological structure of the brain. Each node performs a simple mathematical operation using the outputs from the nodes that feed into it. By tuning each node's operation, the network can collectively learn to model complex functions.

4 Program Design and Development

The program is written in C. The source is organized into files by subject: the input functions, fourier transformation bindings, fractal dimension calculations, and neural network functions each have their own file. The main file is simply a driver that calls the functions in the other files and writes outpt data.

4.1 Input

The input system reads and returns data from an audio file. Currently the only supported format is WAV, but the input functions are modular, allowing for easy expansion.

4.2 Fourier Transforms

The Fourier transform functions are wrappers around the FFTW library. They handle all memory allocation and data management required to perform Fourier transforms using the FFTW library.

4.3 Fractal Dimension

The fractal dimension module performs all operations required to calculate the fractal dimension of a waveform using both the Variation and ANAM methods. The calculations, however, are currently very inaccurate. Improving the resolution of the inner integrations in the ANAM method using linear interpolation was unsuccessful.

4.4 Neural Networks

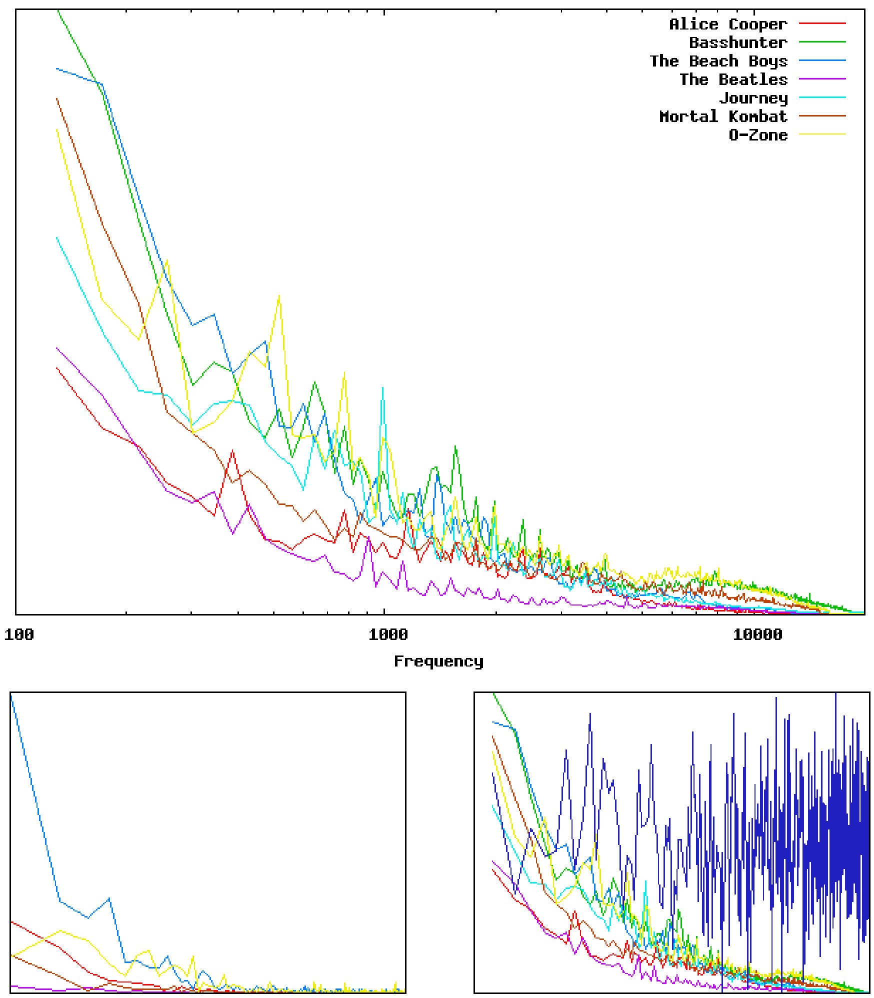
The neural network module provides a barebones framework for future expansion. It includes data structures and functions to create feedforward neural networks and to propagate data through them. However, it does not currently include a learning algorithm, so it should be considered only a basic framework for further development.

4.5 Potential Improvements and Expansion

- Currently, the main driver processes and analyzes the results of the Fourier transforms. It aggregates frequency and amplitude data and performs a second Fourier transform on the aggregated amplitude data. For the sake of organization, these functionalities should be moved to the Fourier transform module.
- The fractal dimension calculations are extremely inaccurate and should be improved.
- The neural network code is currently only a framework. The next step is to add a learning algorithm.
- Another potential for analysis of the audio file is beat detection. Most music has a beat, so beat detection would be useful for music recognition.

5 Results

The program's spectral decomposition worked successfully. The program correctly generated spectrograms and aggregate data useful for analysis. The most interesting result is the total frequency distribution. For audio data, the frequency distribution is inversely correlated with frequency (below). The second Fourier transform, performed on magnitude over the course of the song, however, showed no such trend (below left). These conclusions were confirmed by comparison with random audio data: the frequency distribution of music was a significant departure from the frequency distribution of white noise (below right), while in the second magnitude transform, music and white noise were indistinguishable.



6 Conclusions

This project is a good step toward a multifaceted audio analysis tool. Using its analytical methods, a clear distinction between music and white noise is apparent. However, much remains to be done. Many components of this program can be improved, and additional statistical and signal processing techniques can be added. With significant additions, this program can be a powerful audio analysis tool.

References

[1] Elshafei, M., S. Akhtar, and M.S. Ahmbe. 2000. "Parametric models for helicopter identification using ANN." *IEEE Transactions on Aerospace and Electronic Systems* 26(4):1242-1252.

[2] Graves, Alex, and Jürgen Schmidhuber. 2005. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." *Neural Networks* 18(5-6):602-610.

[3] Bigerelle, M., and A. Iost. 2000. "Fractal Dimension and Classification of Music." *Chaos, Solitons & Fractals.* 11(14):2179-2192.

[4] 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.

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