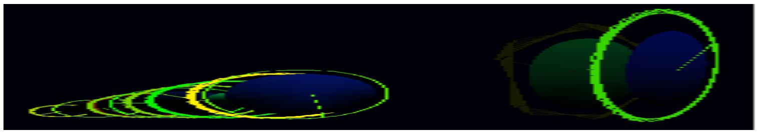
# A View of Music

Ellen Gethner[*], Shannon Steinmetz, Joseph Verbeke
University of Colorado Denver
College of Engineering and Applied Sciences
College of Arts and Media

June 11, 2015



## Abstract

Inspired by the idea of *synesthesia*, or the intermingling of senses [10], we have developed an algorithm that transforms raw sound into a pictographic representation using the physics of a signal. We apply our method to music, the outcome of which is an animation that is not only synchronized with the music, but that mimics aspects of the music itself. We discuss the results of our concept and an implementation, which includes illustrations provided from a variety of music sources. As well, we provide our current parameterization technique and discuss the internal behaviors. We close with a discussion of perception, consonance, and dissonance, and a road-map for future work.

## Introduction

We have begun a research effort that merges the scientific and engineering disciplines of Signal Processing, Computer Animation and Mathematics with artistic creativity. For example, imagine a guitarist with the instrument connected to his or her computer. As the musician plays s/he sees amazing patterns, shapes, colors, and transitions that behave "congruently" to the music in real-time. Our research goal is to construct a formulation (algorithms and software) capable of transforming the physical "shape" of a sound wave in to a pictographic representation that captures a culturally neutral "perception" of the sound. In this paper, we propose an algorithm and discuss our initial implementation for animating the behavior of sound and music.

## Overview and Approach

This article is organized in the following way; first we identify a framework and application implementation for processing raw signal information in an animation format. Second, we speak briefly about the basics of a signal and demonstrate the specific approach we used for constructing a visual geometry based upon raw digital sound data (in the time domain). We conclude with some examples of our results, a brief discussion of consonance versus dissonance and finally our ongoing research, including how we intend to advance this technology beyond the prototype experiments.

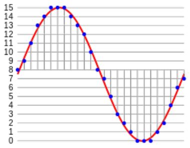
[*] Research supported in part by a Simons Foundation Collaboration Grant for Mathematicians

## Engineering the Animation

We constructed a sound processing framework using Microsoft .Net. The *SoundBus* processing framework utilizes an animation plug-in system where each animation plug-in acts as an interface that can receive both sound data messages and requests to render their current content. This allows us to experiment with different algorithms without losing any previous work. We employ the *Microsoft XNA .Net Framework* along with *C# .Net* and the *NAudio* sound processing package, which acts as the driver connecting our system to the sound input device. Our implementation is capable of seamlessly processing raw pulse data from a raw MP3, .wav file, or direct microphone input. There are several challenges when dealing with an attempt to synchronize the visualization of sound and imagery. At 44K impulses per second, a 100Hz refresh and drawing one image per frame the backlog grows arithmetically over time as $\beta(t) = (44100)t + (-10000)t$. Even if we increase to 100 images per frame after 10 seconds we have a backlog of $\beta(10) = 341000$ images to be drawn. As a consequence we can never keep up with the sound that is playing in real-time without creating tremendous clutter on screen. To alleviate this we abandon the one-to-one rasterization but not the one-to-one calculation. Our framework provides a stream of sound information to an interface designed to process individual samples at a time. Simultaneously there exists another interface mechanism which is called on a 30FPS interval to refresh the computer display. In order to keep the different streams synchronized we employ a timer system that calculates the current temporal backlog and flushes data to the graphics calculation. The processing implementation system then chews off individual data blocks and continuously incorporates the data into a set of running parameters for our geometric figures. At any given time, the interface is asked to render itself in its *current state*. The end result is that we receive a fluid animation that generally mirrors the pace of the sound and neither gets too far behind, nor too far ahead if reading from a sound file.

## The Basics of Signals

Our research depends upon the behavior of a sound wave. Sound waves travel through the atmosphere and generally range from 25Hz to 25Khz [7]. In particular, this represents 25 cycles per second to 25 thousand cycles per second and observe that humans hear in the 25Khz range. As we age our range decreases because ear fibers become brittle over time and can no longer sense changes at such a high rate. The human ear perceives sound by the changes in pressure generated by the frequency on both the up and down cycle of the wave [8]. The speed at which that pressure changes *(e.g., the frequency)* is the way in which a brain interprets information as sound. The faster the change *(higher frequency)* the higher the pitch and vice versa. The relationship so described provides a conduit to decomposing the raw information into its basic parts and in turn algorithmically interpreting and processing information about sound. Most users generally listen to music in the form of a Compact Disc, MP3 Player, or from a television or other stereo source; all of these systems use an encoding scheme called ***PCM***. *PCM* stands for *P*ulse *C*ode *M*odulation and is the preferred means of transmitting and storing digital sound information electronically [1]. The following graph illustrates a simple time domain signal.



*A simple graph of a signal sampled over time*
*(Image taken from Wiki PCM page)*

**Figure 1.**

If one observes the red line as a measurement of how intense a sound is recorded over a period of time (going from left to right on our graph) then one can gain a good idea of how a sound wave is received. In Figure 1 the small blue dots are discrete points identified along the curve. These *Sample* points are where a microprocessor system would measure the sound wave height and store it for use. The number of times sound is sampled determines how accurately the digital copy represents the real sound. The *Shannon-Nyquist Theorem* [7] states that to accurately represent a signal in digital form one must sample at least two times the maximum frequency. Because most music is sampled at a rate of 44.1Khz, we tend to max out around 22Khz on average, which is a standard measure in music. From such an encoding we have all the information we need to break down the original sound into its core components and identify behaviors.

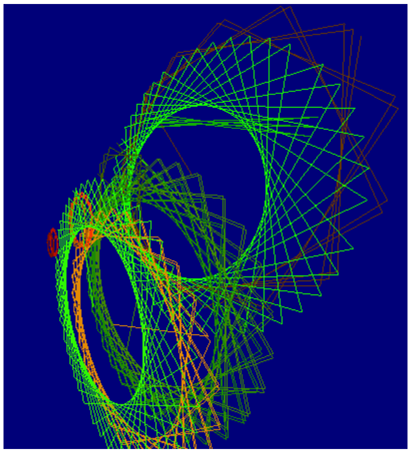
## Time Domain Parameterization

The images shown in the upcoming results section are created using the following approach: we primarily take advantage of statistical characteristics of a sound wave in the time domain. The implementation receives the PCM over time and parameterizes a set of simple dihedral geometric figures whose edges are drawn in stages over time based upon initial parameters. We begin with the set $\mathbf{X} = \{x \,|\, x \in \mathbb{Z}, -2^k \leq x \leq 2^k\}$, which describes the amplitude data. To avoid clutter we limit the total number of animations on screen at any time to $n \in \mathbb{Z}$. Let $\mathbf{S} = \{s_k \,|\, 0 < k < n\}$ represent our parameterized animation elements such that $s_k = (\theta, \phi, \alpha, \vec{P}, r, m, v)$ with $0 < \theta \leq \alpha \leq \phi < 2x\pi$ the starting and ending angle, $\alpha$ the current angle of rotation, $\vec{P} \in \mathbb{Z}^3$ the centroid, and $r, m$ are the radius and color respectively (the color here is the integer form of a bitwise combined RGB value). Finally, $v$ represents the step that determines the number of vertices in the geometric figure. We then define a set of mappings $f_p : \mathbf{X} \to \mathbf{S}$ that map time parameters to an animation element, $f_d : \mathbf{S} \to \mathbb{R}^3$, which maps an animation element to the display (a 2x2x2 bounded region in $\mathbb{R}^3$) and $f_n : \mathbb{R} \to \mathbb{Z}^+$ where $f_n(x) = \lfloor min + x(max - min)(mod\,(max - min))\rfloor$, which normalizes data for screen display. The following is a table containing the initialization parameters that maps elements in $\mathbf{X}$ to elements in $\mathbf{S}$.

| **Parameter** | **Value** | Description |
|---|---|---|
| $x$ | | Current amplitude |
| $x_{k-1}$ | | Previous amplitude |
| $\sigma$ | | Signal to Noise Ratio |
| $g$ | | Gain |
| $\theta$ | $\lvert x/255 * 2\pi \rvert$ | Starting angle |
| $\phi$ | $\theta + \lvert x_{k-1}/255 * 2\pi \rvert$ | Ending angle |
| $\alpha$ | $\theta$ | Current angle |
| $r$ | $x/(max(x_k) * 2)$ | Radius |
| $v$ | $(\phi - \theta)/30$ | Rotational velocity |
| $ColorRed$ | $f_n(x)\ (mod\,255)$ | RGB Red Value |
| $ColorGreen$ | $x_{k-1}\ (mod\,255)$ | RGB Green Value |
| $ColorBlue$ | $\sigma * 255\ (mod\,255)$ | RGB Blue Value |
| $P_x$ | $f_n(rand() + 2g - 1)$ | Centroid X |
| $P_y$ | $f_n(rand() + 2g - 1)$ | Centroid Y |
| $P_z$ | $f_n(rand() + 4g - 1)$ | Centroid Z |

*Time Domain Initial Parameterizations*

**Table 1.**



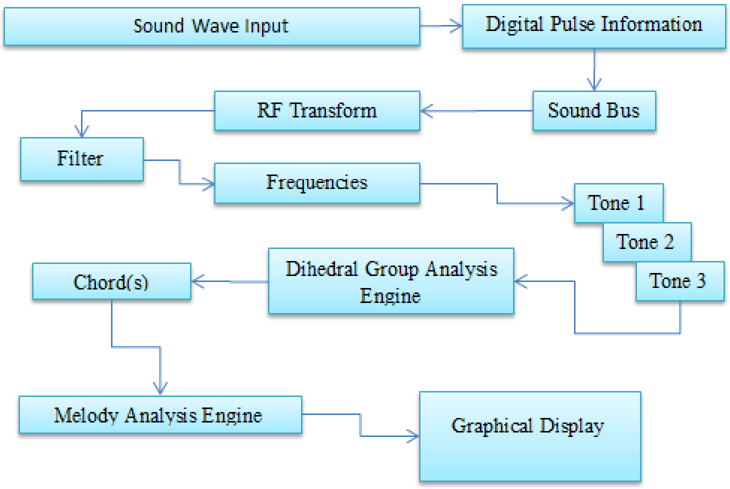
*Example Set of Parameterized Geometric Figures*

**Figure 2.**

As a new amplitude is received, an initial state that represents the signal at that time is constructed by way of the parameters defined in Table 1. There are two key stages consisting of a **paint interval** and **time step**. When a paint interval occurs the elements in **S** are rendered as a curve extrapolated from the set of rotations $R = \{i \,|\, i \in \mathbb{Z}^+, i_0 = \theta, i_{k+1} = i_k + (\phi - \theta)/60, i \leq \alpha\}$. The shape is then mapped to the display with a simple linear transformation $f_{di}(s_k) = (r\,cos(i) + P(s_k)_x, r\,sin(i) + P(s_k)_y, P(s_k)_z)$; this essentially connects the vertices of some partially, or fully formed regular polygon on screen over time. Depending on the current rotational perspective we also paint a *disc* at the centroid of a geometric figure whose size is determined by $r - (\alpha - \theta)/r\phi$ where $r, \phi \neq 0$ and that produces a visual singularity effect. Simultaneously, at each time step we increment the current angle $\alpha$ of each element $s_k$ by $\alpha = \alpha + v$. An animation reaches its life's end when $\alpha \geq \phi$ at which time it is purged. The size, color and vertex count of an animation element is a direct representation of the shape of the pulse waveform at the time it is created. As an additional visual element we also set a gradient tone for the background based upon the current signal strength where $Background\,RGB = (0, 0, f_n((E[X]_k - E[X]_{k-1})/E[X]_k))\,(mod\,128)$. Note that our display rotates the entire view matrix about the $y$-axis (assuming $y$ points north) very slowly in a counterclockwise direction. The rotation is not associated with the data; however the $z$-coordinate is necessary in our mappings and serves to give the user a panoramic perspective on the visuals as they occur.

## Road Map

Our long term goal is to see if one can find a pictographic representation that mirrors the flow, tempo and harmony of music. We know from the work of [8, 9] and [2] that the structure of chords can be represented by the dihedral group $D_{12}$. Is it possible that the number of vertices and structure of our geometric figures are in some way associated with the underlying chord progressions? Figure 3 is an illustration that outlines a path towards creating a valid hypothesis.



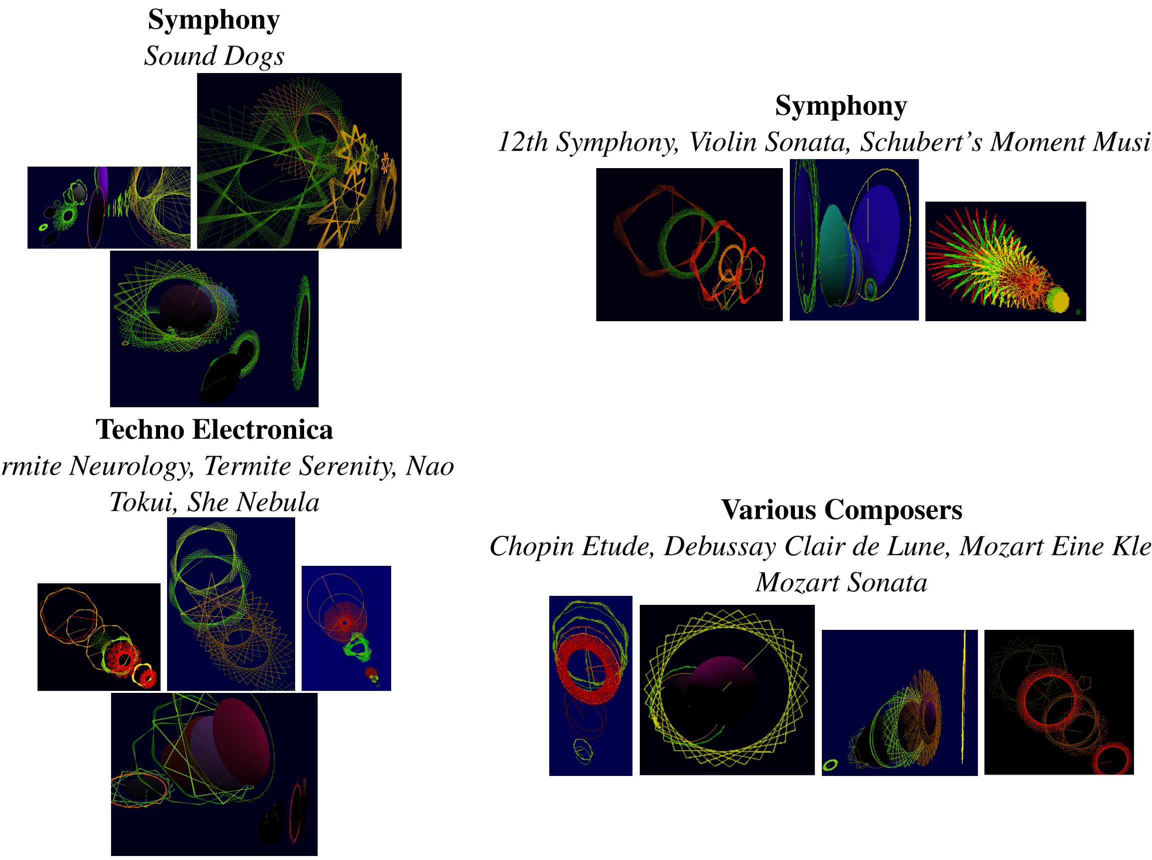
*The flow of our search*

**Figure 3.**

## Experimentation and Results

Our application was run against a handful of music files, which in this case came from symphony music downloaded from the internet. To use the application one simply selects the input source, in this case an

MP3 file, and then click play. Figure 4 displays snapshots of screen captures at different times while playing a variety of symphonies. The actual number of screen shots produced is in the hundreds of thousands of various animations and these are some of the few captured that looked aesthetically pleasing.



**Symphony**
*Sound Dogs*

**Symphony**
*12th Symphony, Violin Sonata, Schubert's Moment Musical*

**Techno Electronica**
*Termite Neurology, Termite Serenity, Nao Tokui, She Nebula*

**Various Composers**
*Chopin Etude, Debussay Clair de Lune, Mozart Eine Kleine, Mozart Sonata*

**Figure 4.**

The images in Figure 4 were generated using the technique described in the **Time Domain Parameterization** section. Notice the conic nature of the successive geometric figures; we conjecture this is due to the changes in amplitude and we are, in reality, seeing snippets of the waveform increase or decrease over time.

## Conclusions

Throughout Western history the concepts of consonance and dissonance, from a music theory standpoint, has changed in almost every era. Before the Baroque era, every harmony with the exception of a unison, perfect fourth, perfect fifth, or octave was essentially considered to be a dissonant harmonic interval. What we associated in the modern day as consonant (e.g. major thirds, major sixths) eventually became accepted within musical composition, and largely thanks to Jean-Phillippe Rameau's publication Traité de l'harmonie, dissonant harmonies were realized as a very integral part of composition due to the resolving progressions they created in conjunction with consonant harmonies [5]. Though such composition techniques have rules that are fundamentally derived from discoveries by Pythagoras, there is another more modern way to analyze consonance from a psychoacoustic standpoint. Through analyzing solely the harmonic components of two tones, we can compare these frequencies with a critical bandwidth to determine mathematically whether the two tones are consonant [3]. Despite having these tools to create or evaluate properties of music, there is no objective method for interpreting how a listener experiences or enjoys a piece of music. Our research and experimentation have provided a springboard toward limitless possibilities of evaluating music by way of graphical depiction, and in a small way has provided a measurement that decouples the idea of perception from aesthetics.

Although there is much work to be done we have successfully constructed a prototype implementation that reads sound from virtually any digital source and creates beautiful animations structurally consistent with the sound wave input. We created a software framework capable of seamless integration of independent processing algorithms as our research continues. We have overcome challenges of synchronizing large volumes of samples with small volume frame rates when dealing with the time domain.

In the time domain implementation we noticed pictographic transitions in the prototype which largely mirror the intensity of the tones; this makes intuitive sense since our parameters are constructed primarily by the changes in amplitude. However, after a demonstration given during a public outreach lecture on Mathematics and Art [4] by the first author, we received feedback indicating that the animations were not well aligned with the live music being performed. Our next step is to extract and utilize the frequency domain for our parameterizations. We will leverage the Prime Factorization Algorithm (a variant of the Discrete Fourier Transform) and associate color to frequency using the color scale due to *Johannes Itten* [6]. In the long run, we intend to leverage the geometry of musical chords and uncover a fundamental connection between the shape of a wave and the mathematics of music.

## References

[1] Midi Manufacturer's Association. History of midi. `http://www.midi.org/aboutmidi/tut_history.php`, 2013.

[2] Alissa S. Crans, Thomas M. Fiore, and Ramon Satyendra. Musical actions of dihedral groups. *Amer. Math. Monthly*, 116(6):479–495, 2009.

[3] F.A. Everest. *Critical Listening Skills for Audio Professionals*. Thomson Course Technology, 2007.

[4] Ellen Gethner. Mining the mesermerizing miraculous mysteries of mathematics...for Art! Mini-STEM School, University of Colorado, 2014.

[5] D.J. Grout, J.P. Burkholder, and C.V. Palisca. *A History of Western Music*. W. W. Norton, 2010.

[6] Johannes Itten. *The Art of Color*. Wiley & Sons INC, 2 edition, 1973.

[7] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, USA, 1997.

[8] Dimitri Tymoczko. The geometry of muscial chords. *Science*, 313(0036-8075):72, 2006.

[9] Dimitri Tymoczko. *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford Studies in Music Theory. Oxford University Press, USA, 2011.

[10] C. van Campen. *The Hidden Sense: Synesthesia in Art and Science*. Leonardo (Series) (Cambridge, Mass.). MIT Press, 2008.