# INTERACTIVE MUSICAL VISUALIZATION BASED ON EMOTIONAL AND COLOR THEORY

A Thesis

by

KARESSA NATEE BOWENS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2008

Major Subject: Visualization Sciences

# INTERACTIVE MUSICAL VISUALIZATION BASED ON EMOTIONAL AND COLOR THEORY

A Thesis

by

KARESSA NATEE BOWENS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

| | |
|---|---|
| Chair of Committee, | Frederic Parke |
| Committee Members, | Mary Saslow |
| | Jeffrey Morris |
| Head of Department, | Tim McLaughlin |

December 2008

Major Subject: Visualization Sciences

# ABSTRACT

Interactive Musical Visualization
Based on Emotional and Color Theory. (December 2008)
Karessa Natee Bowens, B.F.A, The University of Texas at Austin
Chair of Advisory Committee: Dr. Frederic Parke

Influenced by synesthesia, the creators of such ‘visual musics’ as abstract art, color organs, abstract film, and most recently visualizers, have attempted to illustrate correspondences between the senses. This thesis attempts to develop a framework for music visualization founded on emotional analogues between visual art and music. The framework implements audio signal spectrum analysis, mood modeling, and color theory to produce pertinent data for use in visualizations. The research is manifest as a computer program that creates a simple visualizer. Built in Max/MSP/Jitter, a programming environment especially for musical and multimedia processing, it analyzes data and produces images in real-time.

The program employs spectrum analysis to extract musical data such as loudness, brightness, and note attacks from the audio signals of AIFF song files. These musical features are used to calculate the Energy and Stress of the song, which determine the general mood of the music. The mood can fall into one of the four general categories of Exuberance, Contentment, Depression, and Anxious/Frantic. This method of automatic mood classification resulted in an eighty-five percent accuracy rate. Applying color expression theory yields a color palette that reflects the musical mood. The color palette and the musical features are then supplied to four different animation schemes to produce visuals. The visualizer generates shapes and forms in a three-dimensional environment and animates them in response to the real-time musical data. The visualizer allows user input to actively direct the creation of a variety of different visualizations. This personalization of the synesthetic effects of the visualizer invites the viewer to actively consider his or her own unique associations and facilitates understanding of the phenomenon of synesthesia and sensory fusion.

# DEDICATION

To my family, who are always there to coax, cajole, and convince me that I can when I think that I can’t.

# ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Parke, and my committee members, Dr. Morris and Prof. Saslow, for their guidance and support throughout the course of this research.

Thanks also go to my colleagues and the faculty and staff in both the Department of Visualization Sciences and the Department of Performance Studies for making my time at Texas A&M University a great experience.

Finally, thanks to my mother and father for their support and dedication and to my sister for her patience and encouragement.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER I

# INTRODUCTION

The phenomenon of synesthesia has influenced and inspired numerous artists and musicians to create cross-modal works that attempt to illustrate correspondences between senses. There is a long but relatively unexamined history of ‘visual music’, a blanket term for visual arts that aspire to musical analogy (Brougher, Strick, Wiseman, Zilker, & Mattis, 2005, p. 10). Examples of such visual music include abstract art, color organs, experimental abstract film, and music visualizers.

The aspirations of visual music to convey unity of the senses are well served by developing technologies. Often in the past the sense perceptions in one medium, such as music, were conceptually broken down into individual parameters and then mapped onto equivalents in another medium, such as painting or film (Brougher et al., 2005). In such cases the links were personally created and determined by the artists. Digital technology now makes it possible to break down rudimentary information into discrete packets of numbers and/or electrical signals. This enables computers to automate mappings of various analogous structural characteristics from one type of media to another. This is a technique used by many modern music visualizers. However, when the technology is used to simply link rudimentary musical elements to rudimentary artistic elements, then

---

This thesis follows the style of *Computer Music Journal.*

the emotional and expressive aspects of synesthetic art efforts are easily lost to mechanical logic and the resulting products are mere exercises in computing. To ensure the vitality of the final results, more than just numerical synonyms need to be taken into account. As researcher Brian Evans says, “New media may require a new literacy, yet traditional ideas still have much to offer” (2005, p. 23). While technology may make mappings between elements easier to perform, applying theories of aesthetic expression can result in visuals that better reflect the overall mood and characteristics of the music. The aesthetic principles and the creator’s perceptions of the various corresponding media should be analyzed, compared, and then used as criteria when attempting to create digital synesthetic art.

In most examples of visual music, the visuals are informed by and created in response to the music. However, the majority of these works were not created in real-time. Visual music artists listened to a piece of music, then created images based on their personal interpretations and responses to the expression of the music. They had the opportunity to scrutinize the overall emotion of the music, and to mark musical events of importance such as beats, rhythms, and phrases changes. They could then reference this data in the creation of images. Non-real-time generation of visual music allows for more careful and nuanced analysis of the technical and expressive elements of the music. It also results in more carefully considered visuals. However, non-real-time visual music lacks the performance aspect of real-time visualization.

Real-time visual music works, such as light shows, have a creative urgency in their responses. Such immediate reactions result from inherent personal mappings. The creator only has time to respond with the images that immediately come to mind, and so these improvisations are more akin to synesthesia. Digital visualizers produce real-time imagery based on music, but the visuals are often cold and mechanical and so become uninteresting after a short while. Generally, real-time visualizers generate images based on simple correspondences between basic information such as perceived volume mapped to visual size.

The problem is that real-time digital visualizers are supplied with insufficient information to effectively inform their visuals. There is a middle ground between non-real-time creation of visual music, which relies heavily on personal response and careful analysis of musical elements, and real-time creation that does not adequately reflect the music's emotional effects and so results in generic and impersonal imagery. While it is currently impossible for a computer to understand as much information from a music clip as a human does, there are ways for the computer to analyze the digitized music for emotional and expressive content, rather than to just respond to one or two obvious cues such as beat or volume. A real-time digital visualizer should attempt to extract as much of the expressive and technical information as possible from the music, then apply this data to affect the generation of images.

## Artistic Statement

In an attempt to reach this middle ground, I developed a framework for the creation of a more informed interactive real-time visualizer. Using the framework, I created a computer visualization program that implements musical spectrum analysis, mood modeling, and color theory to create images. The program analyzes digital music signals for pertinent musical elements such as beat, loudness, and brightness, and uses these features to determine the general mood of the music. Since color is arguably one of the most expressive visual elements, the primary goal was to illustrate the music mood through color relationships. Color expression theory, based on the writings of artist Johannes Itten, determined the mappings between the music mood and color. Other simple mappings, such as beat to motion cues and tempo to motion speed, further related the visuals to the music. In addition, I included an element of interactivity in the visualizer by making a number of modifiable parameters available to the viewer. This ensured that viewer has the choice of customizing the visuals to fit her own personal interpretations. The personalization and immediacy of this characteristic demands that the viewer focus on her unique perceptions to create individually relevant synesthetic effects.

This aligned the visualizer with Paul Hertz's (1999) conclusion that

> In synesthetic art we may not be able to establish lasting or universal correspondences but we can create structures that work on multiple media, upon multiple senses, and point to a variety of cultural and historical antecedents for their interpretation … the arbitrary nature of correspondences enables us to use aleoretic juxtaposition, partial mappings and associations, parallel couplings and complements freely as a metalanguage of cross-modal composition, a counterpoint of rupture and cohesion. (p. 403-404)

Much like abstract art where the viewer brings his or her own interpretation and creates meaning gleaned from personal experiences and perceptions, digital synesthetic art, such as this visualizer, ultimately gains value by serving as a tool or vehicle through which the viewer may experience new emotional and or intellectual insight.

# CHAPTER II

# BACKGROUND

## The Phenomenon of Synesthesia

From the Greek roots of the word the simplest definition of synesthesia is 'joined sensation'; *syn* means 'together' or 'union', and *aesthesis* translates to 'perception' or 'sensation' (Cytowic, 2002, p. 2). The phenomenon of synesthesia is defined by contemporary scientists as "an involuntary joining in which the real information of one sense is accompanied by a perception in another sense" (Dann, 1998, p. 5) or "the involuntary physical experience of a cross-modal association" (Hertz, 1999, p. 400). This describes the capacity in some people to hear colors, taste shapes, see scents, etc. Most of the reported cases of synesthesia deal with *chromesthesia*, the association of color with sound (Berman, 1999).

## The Study of Synesthesia

The ancient Greeks knew of the phenomenon of synesthesia; the mathematician Pythagoras makes mention of it. But it was not given its current name until the seventeenth or eighteenth century, when many renaissance scientists and composers attempted to create instruments that would make color music (Berman, 1999). In 1690, John Locke described a blind man who synesthetically linked the color scarlet with the sound of a trumpet. Isaac Newton studied the correlation between music and color scientifically, equating the spaces occupied by the seven colors of the spectrum to the relative intervals between the notes of the octave (Dann, 1998). In the early 1800's, Johann Wolfgang von Goethe wrote on the color theory and the relation between sound and color *in Zur Farbenlehre*.

Generally, these Renaissance men were actually only speculating on possible analogies between sound and color, not directly discussing synesthesia as an actual perception. However, their explorations into such subjects do share with synesthesia the notion of a set of "transcendental properties of human sensory capacity" (Dann, 1998, p.11). The first scientific consideration of actual synesthesia took place in the field of psychology, and for a long time the phenomenon was only known to a few in the medical field as an oddity. This changed with the 1883 publication of the poem 'Voyelles' by Symbolist poet Arthur Rimbaud. This sonnet introduced the idea of synesthesia to a much wider audience (Dann, 1998). Romantic artists and philosophers were quickly taken by the concept, championing synesthesia as a higher spiritual ability to transcend the everyday world.

While Romantic thinkers saw synesthesia as a form of perception privileged by a "higher human vision", others were more prosaic (Dann, 1998, p. 17 – 18). Scientists viewed the phenomenon as a problem of optics or as an abnormality of the visual system since the majority of the reported cases dealt with *chromesthesia*. It was referred to as 'color-hearing' or 'sound seeing' until 1892 when Jules Millet suggested the term 'synesthesia in his thesis (Dann, 1998). After the publication of Rimbaud's Voyelles, the number of scientific papers on the subject rose dramatically. 1893 marked the year with the highest scientific interest, with some twenty-six articles published. Most of the studies involved observation and documentation of individuals who experienced authentic strong synesthesia (Dann, 1998). Many scientists saw synesthesia and artists' desire to mimic it as sure signs of the pathological degeneration of the age and linked it to mental illness, hysteria, and homosexuality. It was not until the 1920's that a study by Raymond Holder Wheeler and his blind synesthate student Thomas D. Cutsforth proposed instead that synesthates experienced their secondary sensations not as hysteria or spirituality but as an essential part of their thought processes. Rather than being indicative of a transcendental set of meanings, the images that synesthates see carry their own meaning to each individual person. This study was the very first that made no assumptions that absolute correspondences existed between sight and sound. This research should have done much to demystify synesthesia, but it was not recognized as significant and even outright dismissed within the larger arguments between science and art (Dann, 1998). While many avant-garde artists embraced the concept of synesthesia as inspiration for revolutionary art, their enthusiasm with such ideas was not shared by many. The spirit of scientific objectivity that developed during the late 19th century quelled the popular preoccupation with mysticism, and wide-spread mass interest in synesthesia waned (Berman, 1999).

In recent years research into the subject in both scientific and artistic areas has resumed. With new medical technology such as PET scans, it is now possible to show that medically, synesthesia does exist just as synesthates have long described. Images of the brain prove that in strong synesthetic subjects, blood flow to areas of the brain that control different senses increases simultaneously when stimuli are introduced (Hertz, 1999). Neurologist Richard E. Cytowic has proposed the first new theories for the mechanism of syntesthesia in over fifty years. His studies are also the first since the early twentieth century to include scientific observations of a large number of people who experience strong synesthesia. In his research he has developed five clear diagnostic signs of synesthesia. The criteria are as follows:

> 1) Synesthesia is involuntary. A strong synesthate does not have to try to have a synesthetic response, nor can he or she prevent one from happening.
>
> 2) Synesthetic perceptions are projected outside the body. They appear distinctly from the body, but are continuous with the immediate space around the person.
>
> 3) Synesthesia is durable and generic. Synesthates may see shapes, colors, lines, etc, but not complete and detailed landscapes or portraits. These shapes or colors remain stable over time; if a synesthate hears an E note as blue, then she will always and consistently hear blue E notes.
>
> 4) Synesthesia is memorable. Extraordinary capabilities of memory often accompany strong synesthesia.
>
> 5) In synesthesia, experience and emotion take precedence over thought and reason (Hertz, 1999, p. 400).

As the renewed interest in synesthesia continues to grow, both scientific and artistic researchers often cite Cytowic's studies and writings.

## Terminology and Degrees of Synesthesia

It is important to take into consideration the meaning of the word ‘synesthesia’ and the phenomenon that it has described over the years. As previously stated, the study of synesthesia began in the discipline of psychology and made its way into several other fields. However, the term also has a separate designation in linguistics. In 1901, ‘synesthesia’ began to be used as a literary device to refer to cross-sensory metaphors in literature, and by the 1940’s was expanded to also include the sensory experiences represented by speech sounds and ‘rhetorical tropes’ (Cytowic, 2002, p. 6; Dann, 1998, p. 11).

Many of synesthesia’s current researchers make a definite distinction between synesthesia as an actual neurological phenomenon and synesthesia as an idea. Some go so far as to create new terms or subsets of synesthesia. Researcher Patricia Duffy (2001) differentiates between ‘developmental synesthesia’, which fits the scientific definition of being innate and involuntary, and ‘metaphorical synesthesia’ which refers to the devices used in art or linguistics to express one sensory experience in terms of another (p. 42). The latter is usually used by those who do not experience scientific synesthetic phenomenon.

Researcher Lawrence Marks acknowledges the difference between the two but believes that they are linked. Marks and Dr. Gail Martino instead conceptualize synesthesia as a continuum that spans a spectrum from ‘weak synesthates’ to ‘strong synesthates’ (Duffy, 2001, p. 45 -49). Marks and Martino’s experiments indicate that people “cannot help but make automatic connections between corresponding auditory and visual input" (Duffy, 2001, p. 48). The ubiquitous presence of cross-sensory metaphors in poetry, art, and everyday languages have convinced them that there is indeed a deep connection between our senses and that all humans share an ability to perceive such correspondences.

## Synesthesia and the Arts

### *Synesthesia as Inspiration*

The concept of synesthesia indicated to many European artists of the late 19th and early 20th centuries that art and music should strive to fuse together as one. The previously mentioned Arthur Rimbaud was part of a circle of artists, poets, and philosophers intrigued by mysticism and transcendent states of mind. While pouring through medical journals searching for information on non-ordinary states of consciousness, he happened upon an article on synesthetic 'colored hearing'. He was inspired by the idea and subsequently penned the imagery filled sonnet "Voyelles" which linked color to vowel sounds. Many artists, especially Romantics, embraced this intriguing new phenomenon, championing it as a vehicle to transcendence. Even despite evidence brought forward that Rimbaud's poem was inspired by an actual medical phenomenon, few paid any attention to such a rational and frankly, boring explanation. They were more interested in the fantastical and mysterious aspects of synesthesia and instead upheld the idea that Rimbaud had managed, by practicing a disordering of the senses, to achieve the ability to see on a higher spiritual level (Dann, 1998). French

Symbolist poets were especially taken with the idea. They believed that synesthesia was an enviable visionary ability, and that those who experienced it possessed a special mystical power of unifying perception beyond that of the mundane, non-synesthate world (Duffy, 2001).

The infatuation occurred not only in France, but also spread to German and Russian Symbolist artists and intellectuals as well. In Germany a contemporary philosophical development in the late 1880's also began to question the boundaries between the arts. Many of these same Romantics and Symbolists infatuated with synesthesia were similarly intrigued by German composer Richard Wagner's concept of the *Gesamtkunstwerk.*

## Gesamtkunstwerk

The *Gesamtkunstwerk* was Wagner's self-proclaimed 'art work of the Future', manifesting itself in grand operatic spectacle meant to excite multiple senses. It was the union of distinct art forms, forming a third in which the two separate initial forms would find fulfillment and reach their true potential (Shaw-Miller, 2002). Curiously this future art was grounded in the philosophies of ancient Greek art. Wagner envisioned an artistic past where Drama unified dance, music, and poetry into a spiritual guide. He harkened back to Plato's view of the power of music as a tool for the most effective emotional communication, as well as to Aristotle's belief that musical mimesis speaks directly to the soul (Shaw-Miller, 2002).

Friedrich Nietzche, inspired by Wagner, went further into the study of ancient Greek music and categorized its interpretations into an Apollonian view and a Dionysian view. The art of sculpture, says Nietzche, was indicative of Apollo's realm of fair appearance, introspection, and individuality. The art of music, however, was ruled by the Dionysian ideals of union between people and union with nature (Shaw-Miller, 2002). Apollo provides the structure and control while Dionysus moves and compels the emotions. In his opinion music, the Dionysian, has a more primal, fundamental truth than the illusions of the mimetic art of the Apollonian. Nietzsche praised Wagner for letting the Dionysian spirit of music free. Wagner, in turn credited Beethoven as the composer who had freed music from rational Apollonian control and returned it to its role as an articulator of pure feeling (Shaw-Miller, 2002). The Apollonian, however, was necessary and even desired because it provided the illusion of a "visible middle world" that lends to a greater connection to and understanding of music, lest the power of the music overwhelm the listener (Shaw-Miller, 2002, p. 43). Whereas the music makes the visuals more intense and compelling, the visuals make the music make sense. Music lets the listener fall into and become absorbed into the visuals, while the visuals pull the listener back from the music just enough so that it does not over-power self-knowledge and can be conceivable to the mind.

Nietzsche, Wagner, and philosopher Arthur Schopenhauer dismissed the idea that music was a mimetic of speech and natural sounds. They instead saw it as a mode of universal communication that was older and more effective than language (Shaw-Miller, 2002). The old idea of music as representative changed into a view of music as expressive, and so music began to be held as the example by which to compare the other arts. Schopenhauer, a contemporary of Wagner's who had great influence on his later writings, wrote: "…Therefore music is by no means like the other arts, namely a copy of Ideas, but a *copy of the will itself*, the objectivity of which are the Ideas. For this reason the effect of music is so very much more powerful and penetrating than is that of the other arts, for these others speak only of the shadows, but music of the essence" (Shaw-Miller, p. 37-38). German writers elevated music to the highest and most pure of the arts because it was the least imitative. All other arts were charged with aspiring to music's imprecise and expressive nature (Shaw-Miller, 2002).

### *Abstraction in Art*

In the early 20th century a number of artists also began to see the abstract nature of music as a concept to aspire to in their own work. The old normative idea of mimesis was questioned and then rejected as too restrictive to portray the true nature of human experience (Gooding, 2001). This signaled a great shift in creative thinking among artists. They were living at the beginning of a dramatically changing new world; science had smashed old speculations and revealed new wonders, technology made the unimaginable possible, politics and society were shifting into new forms, cities and countries were transforming at exponential rates. What had been 'true' to artists in the past did not seem to apply in this new age. Visual artists were now intentionally trying to break away from what they saw as the rigid adherence to representation and seeking a new way to communicate. They began to experiment with abstract expressions of their ideas instead of literal portrayals. “They set out to create an art that would reveal aspects of reality that seemed inaccessible to the techniques and conventions of figurative art” (Gooding, 2001, p. 6).

The main purpose of abstraction in art has been to move “away from the *representation* of recognizable objects in pictorial space” and instead strive “towards the presentation of a painting or sculpture as a real object in real space” (Gooding, 2001, p.7). This is the difference in the purpose and effect between abstract and figurative art. Figurative arts present an image to the eye, while abstract arts seek to represent something to the mind (Gooding, 2001). A number of 20th century artists hoped that abstraction might imbue their work with a sensuous or spiritual energy in the space around it, or that the art object would become something of a sacred icon that brought about a sense of transcendence.

“Above all, music provided the example of a purely non-representational art with variations of formal structure and great affective power”, and was of great inspiration to visual artists looking to break away from imitation (Gooding, 2001, p. 9). In much the same way that music often expresses a subjective meaning dependent on each viewer, abstract art is also subject to diverse personal readings. The meaning of abstract art is created in the receiver’s imagination when the actualities of the piece are experienced. Abstract art “demands the actual encounter, the sensation of the thing itself. It depends for its effects, whether they are simple or complex, sensuous or conceptual, upon the presence of the viewer, who brings possibilities of meaning to its presentations of forms and colours, its visible patterns and rhythms, its forms, shapes, and textures”

(Gooding, p. 11). It demands the imagination to work because it will not outright literally reveal its references. Like music, abstract art “gives the spectator an unprecedented freedom of imaginative response” (Gooding, p. 10).

# CHAPTER III

# PREVIOUS WORKS

"In affect, the idea of synesthesia served to mediate between music and visual art in the early 20th century and proved essential to the development of abstraction" (Brougher, Strick, Wiseman, Zilcer, & Mattis, 2005, p. 16). The foundation for most early abstraction was art that aspired to the condition of music by pairing tone with color, termed 'visual music' (Brougher et al., p. 17-18). According to museum directors Ned Rifkin and Jeremy Strick, the goal of visual music that developed over time is to "invent a kinetic nonrepresentational art akin to pure instrumentational music" (Brougher et al., p. 7). Because of its attempts to correlate sound and visuals, visual music can also be called synesthetic art.

## A Brief History of Visual Music

The influence of synesthesia and musical analogies on the development of visual art can be traced from the late 19th century to the present. The relatively unexplored history of synesthetic art includes such diverse fields as abstract painting, experiments with color organ light projectors, abstract films, video and digital recordings, and multimedia installations (Brougher et al., 2005).

### *Abstract Painting*

A number of painters attempted to capture the connections between art and music on canvas. Some artists used the formal and compositional elements of music to arrange their visuals, while others sought to create visual representations of music or evoke sound with their paintings. Others attempted to correlate the musical scale and the color spectrum (Brougher et al., 2005). There were also those who illustrated the musical expression of their works by using musical terms as the titles of their works. Often the titles suggested the relationship to musical forms, such as Georgia O'Keefe's *Blue and Green Music* and Frantisek Kupka's *Fugue in Two Colors* (Brougher et al., 2005). Wassily Kandinsky, one of the most famous abstract painters, was especially prone to using musical terms as titles of his works. He created series of *Impressions*, *Improvisations*, and *Compositions*. To achieve abstraction, Kandinsky went further than any artist of his time in completely using musical analogy to develop an entire aesthetic theory (Brougher et al., 2005). Mel Gooding quotes Kandinsky, who described his *Impressions* paintings as a "largely unconscious, spontaneous expression of an inner impulse" (Gooding, 2001, p.23). This is much in the same way, Gooding theorizes, as a musician might spontaneously improvise a musical piece. The repetition of forms and interplay of colors and textures in Kandinsky's painting *Controlled Improvisation: Fugue* (Fig. 1) are analogous to the repetition of rhythm and form and interplay of notes and timbres in a musical piece. (Gooding, 2001).

Figure 1. *Controlled Improvisation: Fugue.* Wassily Kandinsky, 1914. [Gooding, 2001, p. 22].

Unfortunately, abstract painting has an inevitable short-coming; music is time-based and paintings are not. Paintings may convey a sense of motion, but they are frozen in time. To overcome this shortcoming, a number of artistic tinkerers looked instead to instruments for inspiration, making their synesthetic art into performance instead (Brougher et al., 2005). The color organs and light art they created provided a moving alternative to painting.

### *Color Organs*

The term color organ includes various devices that play colored light in conjunction with either recorded or performed music. Color organs have a long history that predates the idea of abstraction in art. Renaissance men such as Isaac Newton were experimenting with colored music from the seventeenth century. Newton studied the correlation between music and color scientifically, equating the spaces occupied by the seven colors of the spectrum to the relative intervals between the notes of the octave (Dann, 1998). Inspired by Newton's work, Father Louis Bertrand Castel invented the *clavecin oculaire*, a color organ that produced colored light to accompany musical notes. Erasmus Darwin, Charles Darwin's grandfather, attempted to create a color-harpsichord towards the end of the eighteenth century (Duffy, 2001).

Daniel Baranoff-Rassine invented a polychromatic palette in 1909 in his quest to create an art of mobile color. His color organ, the *piano optophonique*, consisted of hand-painted colored disks that were mounted with mirrors, filters, and lenses in a light projector. The keyboard controlled the moving color projections, producing dynamic abstract light mixtures that brought the color organ closer to music than pure painting could hope to achieve (Brougher et al., 2005).

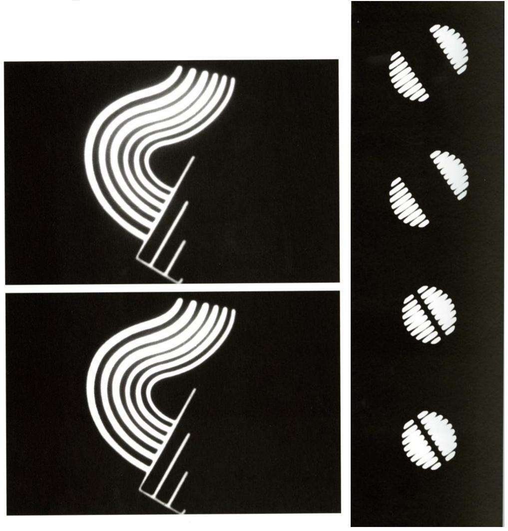
Alexander Scriabian, who claimed to be a true synesthate, was one of the best known composers of synesthetic music. His symphonic poem, *Prometheus, The Poem of Fire* included in the score an imaginary fantastical instrument he called the *luce* which would have converted sound into colored light. Complex spatial patterns of organized color made by the luce were to interact with musical pitches of the orchestral instruments in a complex counterpoint. The production was never finished, but it was ahead of its time and predicted music-kinetic art, performance art, and multimedia (Berman, 1999).

### *Abstract Film*

In the early 1920's, the new medium of film and the technology of cinema made new forms of visual music possible. Abstract film united the concepts of abstract painting and the temporal motion and performance aspect of light art (Brougher et al., 2005). The originators, such as Hans Richter and Viking Eggeling, worked in black and white to create sequences of geometric forms that moved over time. As film technologies developed further to include color and sound, the experimental films used these advantages to create more fully synesthetic art. Artists like Oskar Fischinger and Norman McLaren created extended pieces with color, form, sound, and most importantly, action (Brougher et al., 2005). Similarly to artists producing static abstract works, they sought to create films that aspired to the condition of music, rather than simply illustrating music. Cinema offered invaluable means to this end, providing "flexible montage of time and space, measured pacing and control of gaze, exact repetition, single frame diversity and continuity" (Moritz, 1989, para. 3). Leopold Survage, an artist who produced a series of *Rhythm colore* paintings with the goal of turning them into an animated film, suggested that artists must "get rid of [painting's] last and principle shackle—immobility—so as to become as supple and rich a means of expressing our emotions as music is" (Brougher et al., p. 97).

Some of the most well-known abstract filmmakers, Richter, Eggeling, and Walter Ruttman, fused film and painting. Ruttman's *Light-Play Opus 1*, a hand-tinted work of cinematic art, was the first experimental film to be shown publicly in theatres (Moritz, 1989). Richter and Eggeling had previously worked together producing extremely horizontal or vertical canvases and continuous scrolls to extend painting and map out motion sequences. Film allowed them to make those sequences possible. Eggeling's only existing film, *Symphonie Diagonal,* (Fig. 2) shows his distinctive style (Brougher et al., 2005).

Figure 2. Stills from *Symphonie Diagonale.* Viking Eggeling, 1924. [Brougher et al., 2005, p. 102].



In the 1930's Hollywood began to pay attention to this new avant-garde film. Richter, Ruttman, Eggeling, and even Fischinger's works were known by film studios. Paramount studios brought Fischinger over from Europe to work for them. He also contributed much to Disney's famous *Fantasia*, perhaps the best known example of visual music. His film *Allegretto* (Fig. 3) (1936/1943) was set to a jazz score. Influenced by Hollywood animation, it was multi-layered, colorful, and dynamic. His works owe much to Kandinsky, Kupka, and other 20th century abstract painters, though his imagery was more geometric. In his films, the simple geometric forms "through expansion and contraction, begin to transform on multiple planes, climaxing in pyrotechnical explosions of color and flicker" (Brougher et al., 2005, p. 110). Fishchinger's visual language was based on both temporal and static interconnections and used the massive size of the movie screen to emphasize the opposing motion arcs. His films made abstract work public spectacle.

Figure 3. Still from *Allegretto*. Oskar Fischinger, 1943. [Brougher et al., 2005, p. 108].

### *The Influence of Technology*

Gene Youngblood hypothesized in his 1970 book *Expanded Cinema* that technology was leading art to a new 'synesthetic cinema' of intermedia, fusing digital, holographs, videos, and lasers (Brougher et al., 2005, p. 120). The Whitney brothers, John and James, were seminal in merging art and technology. John was a firm believer that technology was the only solution to creating effective visual music, stating that "Computer graphics systems present an opportunity to realize an art of graphics in motion with potentials that are only now conceivable" (Youngblood, 1970, p. 207). His career included accolades in both artistic and technological fields. Whitney studied photography and musical composition and produced award-winning experimental animated films along with his brother James. John went on to direct films before creating his own company, Motion Graphics, to focus on the invention and use of visual music machines. He built an eight-millimeter optical printer and developed a system of creating sound from the motion of a pendulum. This system allowed the brothers to produce electronic tones that spanned a four-octave range. They then synched the sounds to images of light through stencils, producing a sophisticated example of visual music in which image and sound were intrinsically linked (Brougher et al., 2005). John also turned to World War II era military motion control devices and computer graphics systems, specifically the guidance and control systems for the M-5 and M-7 Antiaircraft guns, to create a large machine capable of producing complex and fluid motion pieces (Youngblood, 1970, p. 208-210). Using such technologies he created the films *Permutations* (1968) and *Arabesque* (Fig. 4) (1975), and influenced his son John Whitney Junior's immersive 3-screen film *Side Phase Drift* (Brougher et al., 2005). Another one of his machines, named *yantra*, ("machine" in Sanskrit), was an analog computerized optical printer. His brother James borrowed the machine to create his film *Lapis*. This film is considered one of the great visual music works, weaving art, science, and spirituality together. James' film *Yantra* [Fig. 5] (1950 -1957), which shares its name with John’s machine, was perhaps the first abstract film of the computer age. It was made up of stencils of thousands of tiny dots created by punching pinholes into cards. The cards were combined in the optical printer to create shifting shapes that merged, split, and reformed. Reducing the image down to its base element, a point of light, made this possible (Brougher et al.).

Figure 4. Stills from *Arabesque.* John Whitney, 1975. [Brougher et al., 2005, p. 146-147].

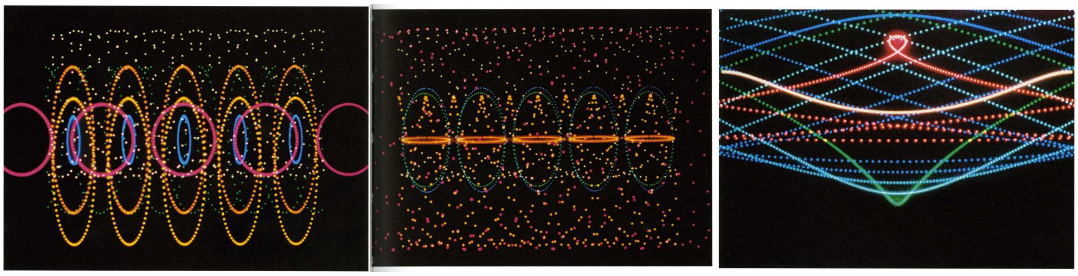
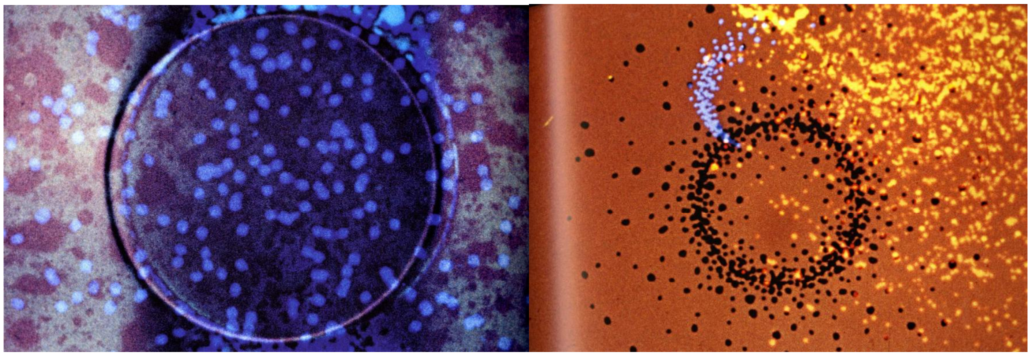


Figure 5. Stills from *Yantra.* James Whitney, 1950-1957. [Brougher et al., 2005, p. 130 – 131].



### *Installations and Light Shows*

Installation art further incorporated the public into visual art. While a painting or a film requires a fixed gaze, sounds can be perceived from any direction, as it exists in and moves throughout space. Installations activate the space occupied, and in their unifying nature approach and augment the physical qualities of sound (Brougher et al., 2005). One of the first examples of installation visual music occurred in San Francisco in 1957. Jordan Belson, formerly a painter, turned to abstract film after seeing the works of Fischinger and the Whitney brothers (Moritz, 1996). Belson used old and new technology, including animation, lasers, and liquid crystals, to create cosmic art based on Eastern metaphysics. Together with Henry Jacobs, an electronic composer, he created an audiovisual collaboration using the projectors and multidirectional sound system at San Francisco's Morrison Planeterium. The performances featured electronic and Afro-Cuban music by a number of artists, accompanied by the films of Belson and James Whitney, along with color projections and the planetarium's star effects. A variety of these images were superimposed over each other to create complex effects. These performances were known as Vortex concerts and quickly became very popular (Brougher et al.). The pioneering Vortex concerts continued from 1957 to 1959 and were even included in the 1958 Brussels World Fair (Moritz, 1996).

Light shows were a contemporary experiment in creating immersive visual music. In the 1960's interest in transcendental spiritualism reemerged. Synesthesia was once again championed as a form of higher consciousness by the American counterculture movement (Dann, 1998). Fueled by the popularity of such hallucinatory drugs as LSD, which induce synesthetic sensations in users, sensory fusion and confusion became the goal of psychedelic art. LSD experiences inspired the light shows that gained prominence during that decade (Dann, 1998). In 1962 Elias Romero, an art student, began to perform light shows in Beat poetry readings, galleries, and coffeehouses. He swirled colored inks, oils, vinegars, and other liquids in a dish and projected the paintings on walls, with musical or poetic accompaniment (Fig. 6). Abstract painter Bill Ham also began to perform light shows combining the liquid projections with film and slides (Brougher et al., 2005). Psychedelic light shows became a popular presence at rock concerts. The Joshua Light Show, a light-show group composed of former Carnegie Institute of Technology students, performed with Frank Zappa and Janis Joplin (Fig. 7). Filmmakers and artists influenced by Fischinger, theWhitneys, and Belson formed Single Wing Turquoise Bird, another light-show group (Brougher et al.). Between 1967 and 1968 they performed at the Shrine Exposition Hall, providing panoramic light shows for rock acts such as the Grateful Dead and the Velvet Underground. Their shows were collaborative improvisations, relying on group interactions to produce unique visuals (Youngblood, 1970).

Figure 6. Stills from light shows. Elias Romero, 1962. [Brougher et al., 2005, p. 158].

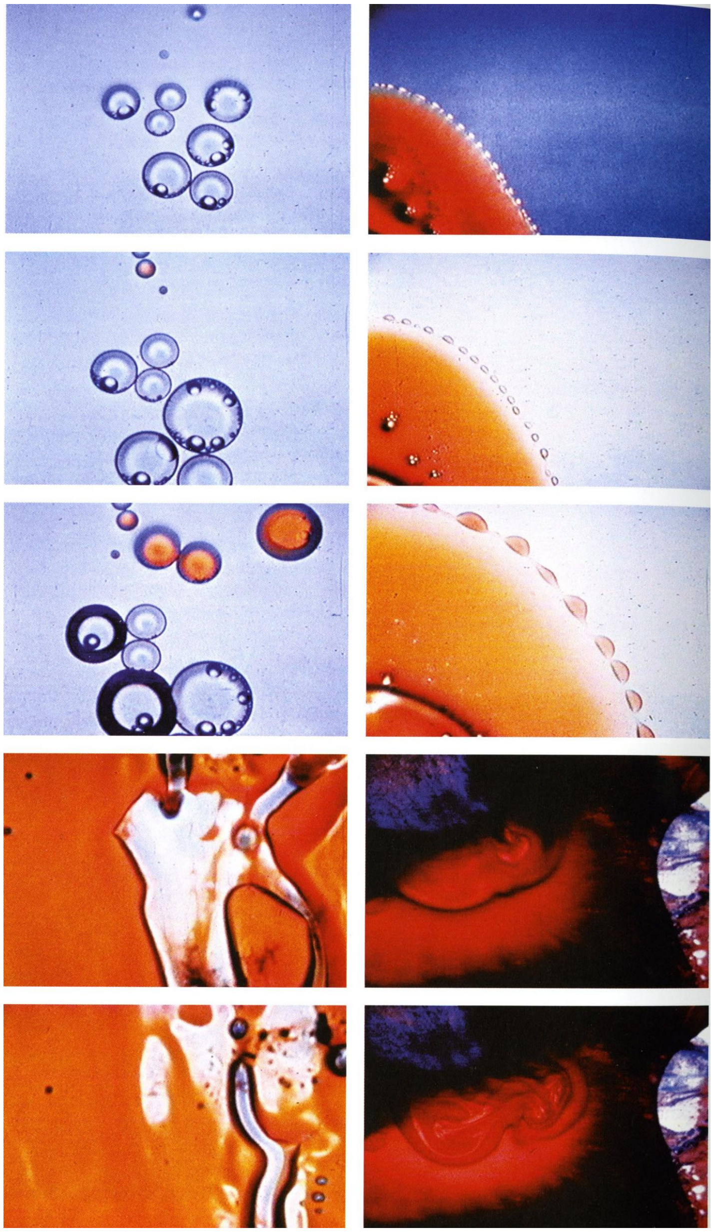
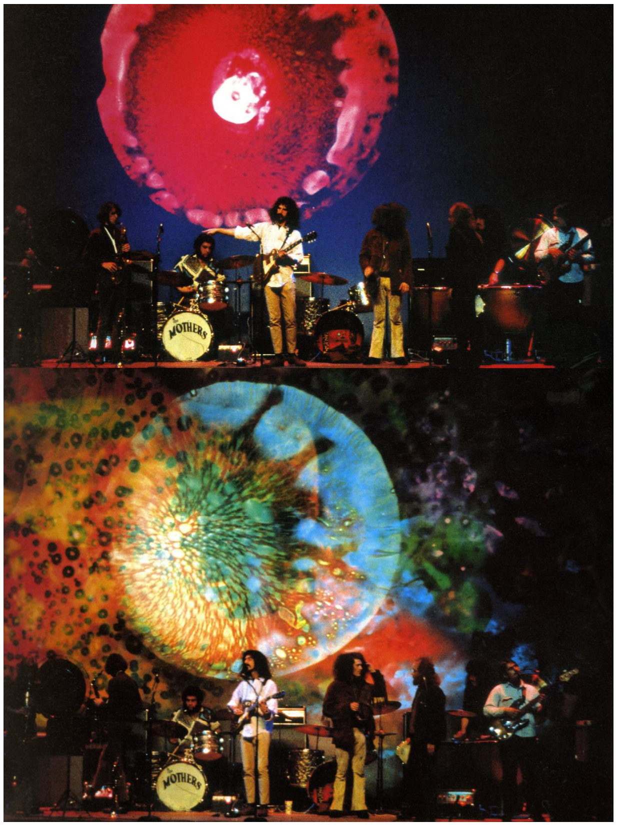


Figure 7. The Joshua Light Show. 1967. [Brougher et al., 2005, p. 163].



The light show as an art form combined high art and popular culture, bringing painting, film, color organs, and music together. As the art became more abstract, it also became more ‘real’ in a physical sense as these installations moved the visual art from a two-dimensional canvas into physical space. The distance between the viewer and the art lessened and the art was allowed to operate directly on the body and senses. Similarly to their Symbolist predecessors, light show artists sought to achieve a new state of consciousness by creating a multi-sensory and super-sensory environment (Brougher et al., 2005). The use of the new technologies of the time allowed these installations and light shows to come closer than previous forms of visual music to evoking synesthesia in viewers.

Unfortunately, light shows proved to be only a trend tied in with the 1960's spirit of a collective mindset. Due to clunky equipment and the fact that it relied on many performers from different backgrounds, the light show fell out of public favor by the 1970's. However, the light show lives on in spaces such as the discotheque and at rave parties. Other works based the idea of light shows have survived by assuming the form of pseudo-real time video works performed on television and computer monitors (Brougher et al., 2005).

### *Automation*

Most of the examples of visual music described previously were constructed in non-real-time. Live light show performances are the exception. The artists created immediate imagery in response to live music, giving the light shows a spontaneous performance aspect not present in the other types of visual music. While seemingly similar, live color organ performances were planned out and composed beforehand. However, color organs and other color instruments introduced the idea of persistent mappings between sound and visual elements. The characteristics of spontaneous performance and reliable sound/visual mappings were both important concepts in the development of real-time automated visual music.

It was perhaps the cumbersome and time-consuming feat of coordinating performers, machines, and tools for light shows and color music performances that lead to a desire for automation. Automated visual music, most readily illustrated by digital visualizers, relies on predefined links between sound and visual elements to automatically generate imagery in response to prerecorded or live music.

### *Digital Visualizers*

Digital media manages to unite music and visual arts because both are created out of bits of electronic data. In the 1990s, interest in visual music reemerged with artists of the digital generation. Widespread availability of powerful and user-friendly personal computers led to the development and resulting popularity of music visualizers, which generate animated imagery based on music. The 1999 Windows Media Player application *Visualizations* created metamorphosing designs as visual representations of any given music played through it (Duffy, 2001). Such applications are now common in other digital media players like Winamp and iTunes. The software system Bomb, developed by Scott Draves, produces visual music using multi-layered cellular automata, reaction diffusion, an icon library, and a number of special graphics programs. This produces a fluid, textured, rhythmic and animated video stream that is generally non-representational (Fig. 8). Bomb is described as “eye-candy software” that, through iteration and recombination, produces novelty from a fixed program that results in a variety of different effects and forms (Draves, 2000, para. 9; 21).

Figure 8. Still image from *Bomb* visualizer. Scott Draves, n.d.. [Draves, n.d.].

Robyn Taylor, Pierre Boulanger, and Daniel Torres from the University of Alberta developed a real-time music visualizer that mapped musical features to three different types of responsive imagery. Their mappings include vocal timbre and piano chords mapped to responsive video, melodic information mapped to the animation of a virtual character, and vocal dynamics mapped to interactive aspects of a virtual space (Taylor, Boulanger, & Torres, 2006). Their visualizer has been used in a live concert to produce a real-time audio-visual piece. The methods used to create the framework for this visualizer will be described later in this paper.

# CHAPTER IV

# METHODOLOGY

As technology evolved and computers became faster, synchronizing musical and visual data became more easily done. The digitization of analog sound allowed computers to represent and analyze music signals and extract musical information from the data. Evolution in visualization made it possible for computers to generate and animate colors, shapes, and forms in both two and three dimensions. Real-time analysis of music is now possible, and corresponding images can be created to almost simultaneously match the sound signal. Visualizers use the stream of musical data to generate and inform the stream of visual data. Elements of the sound data are mapped to image elements which control the characteristics of the created visuals. The process consists of analyzing the musical data for pertinent information, deciding how this musical information will be mapped to the visuals, and then producing the visuals.

## Digital Audio Analysis

Before an analysis of music can be attempted, it is first important to understand how sound works in the real world. These physical properties allow computers to represent and understand sounds. Continuous-time acoustic sounds made by instruments must be transformed into discrete-time digital signals to be stored in computer memory. These digital signals contain information about the sound waveforms that can be analyzed and manipulated for use in music related applications.

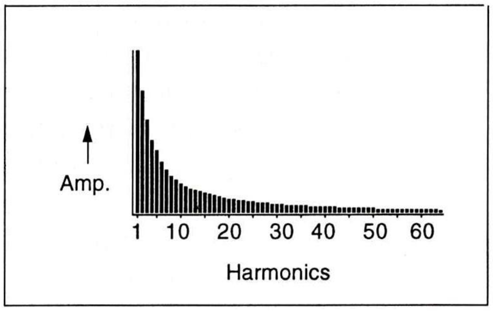
### *Sound Signal Basics*

"Just as an image can be described as a mixture of colors (frequencies in the visible part of the electromagnetic spectrum), a sound object can be described as a blend of elementary acoustic vibrations" (Roads, Strawn, Abbott, Gordon, and Greenspan, 1996, p. 536). In the physical definition, sound is defined as changes in air pressure transmitted as vibrations through the air. These vibrations are represented as compression waves, and so are characterized by the common properties of waves including frequency, period, wavelength, and amplitude (Roads et al., 1996).

If the pressure of the wave varies in a repeating pattern then the sound has a periodic waveform. Sounds with no intelligible pattern are called noise. Most sounds fall somewhere in between these two extremes. A cycle is one repetition of a periodic waveform. The fundamental frequency is the number of these cycles that occur per second. In acoustical terminology it is usually measured in Hertz (Hz), which is equivalent to 'cycles per second'. Frequency corresponds to the perceived pitch of the waveform. Wavelength, or period, is the length of the cycle. As wavelength increases, the frequency in Hz decreases (Roads et al., 1996). Therefore, sounds with longer wavelengths have a lower frequency (i.e. pitch) than sounds with shorter wavelengths.

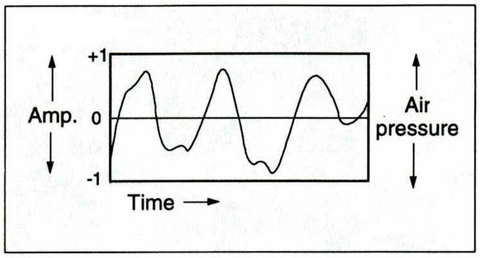
As stated before, most sounds fall somewhere in between pure periodic waves and pure noise. These quasi-periodic or quasi-noisy sounds have many frequencies besides the fundamental frequencies present. The frequency content of a waveform can be graphed as frequency versus the amplitude of each frequency component (Fig. 9). Such frequency-domain or spectrum graphs show the harmonics and partials of the sound. Integer multiples of the fundamental frequency are called harmonics. Partials are any frequency component of a waveform, regardless of whether it is a multiple of the fundamental frequency (Roads et al., 1996).

Figure 9. Frequency-domain spectrum. [Roads et al., 1996, p. 17].



Graphing sound in the form of air pressure versus time yields a time-domain representation graph that illustrates the amplitude of the sound (Fig. 10). Amplitude is measured as the vertical distance between zero pressure and the highest or lowest pressure on the waveform. Amplitude corresponds to the perceived loudness of a sound.

Figure 10. Time-domain graph of a signal. [Roads et al., 1996, p. 16].



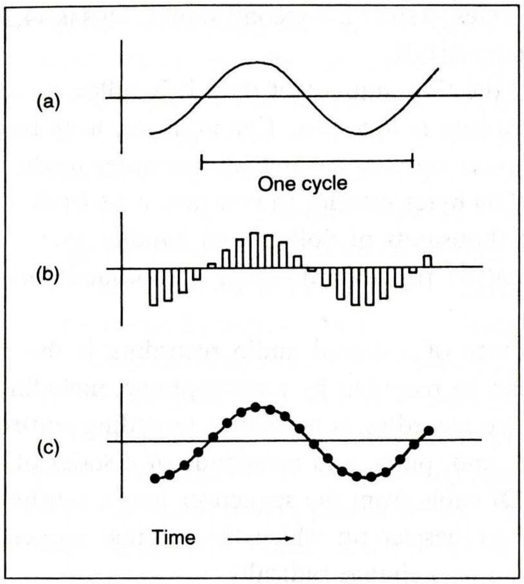
### *Analog Sound to Digital Sound*

The changes in air pressure from a physical sound are analogous to changes in electrical quantity, or voltage. Simply stated, it is possible to modify electrical properties to match air pressure changes (Roads et al., 1996). Digital sound representation involves converting the pressure changes of analog sounds into changes in voltages. Then the continuous-time signals of the analog sound voltages must be converted to discrete-time signals using an analog-to-digital converter (ADC) device. The ADC is driven by a sample clock. At each period of this clock the ADC converts the voltages into a string of numbers and stores them in memory. This process is called sampling. To play back the sound, the numbers are read from memory and sent to a digital-to-analog converter. The DAC, also driven by a sample clock, changes the number stream into a series of voltages. These are filtered into a continuous-time waveform, amplified, and sent out to a loudspeaker (Roads et al., 1996).

### *Sampling*

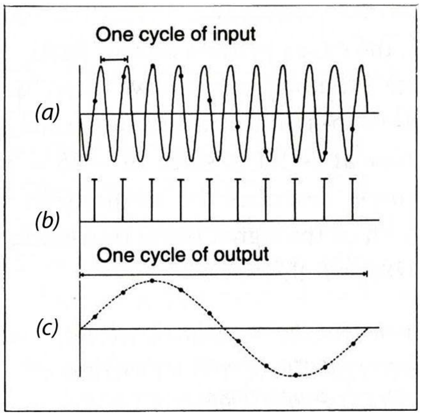
The most important part of the digital audio representation process is sampling. Figure 11 shows how analog signals are converted to digital signals by samples. Fig. 11 (a) shows the original analog signal. Fig. 11 (b) shows the signal represented as samples. The samples are measures of the original signal taken at certain equal intervals of time. They are stored as binary numbers. Fig. 11(c) shows the reconstruction of the sampled waveform from (b). The values of the sampled waveforms are joined by the DAC and a smoothing filter that ‘connects the dots’ between the discrete samples to produce a waveform that looks and sounds like the original signal (Roads et al., 1996).

Figure 11. Analog, sampled, and digital signals. [Roads et al., 1996, p. 25].



The sampling frequency, or the rate at which the samples are taken, can be problematic. Also called the sampling rate, it is measured in samples per second and expressed in Hz. If the sampling rate is too low, distortion called aliasing or foldover can occur. Fig. 12 illustrates what happens when samples are not taken often enough to accurately reflect the signal. The resulting waveform is completely different than the original, so the outputted sound would play at a pitch different from the original (Roads et al., 1996).

Figure 12. An example of aliasing. [Roads et al., 1996, p. 29].



To prevent aliasing from happening, there must be at least two samples per period of the original waveform. In 1928 Harold Nyquist developed the sampling theorem, which describes the relationship between the sampling rate and the maximum frequency of the transmitted signal. Essentially it states that, "In order to be able to reconstruct a signal, the sampling frequency must be at least twice the frequency of the signal being sampled" (Roads et al., 1996, p. 30). The eponymous Nyquist frequency is the highest frequency that can be accurately produced in a digital audio representation, or half the sampling rate. Since the upper range of human hearing is around 20kHz, in many common musical applications the Nyquist frequency is also 20kHz. As such, the sampling frequency must be at least twice as much, or 40kHz. The sampling rate of a compact disc recording is 44.1 kHz, which gives a bit of headroom to account for frequencies above 20kHz (Roads et al., 1996).

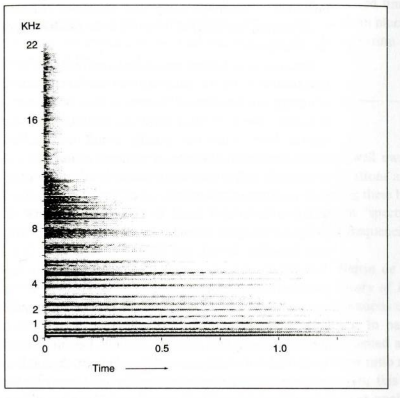
Another potential problem of digital audio representation comes from quantization, or discrete amplitude resolution. The digital values of the sampled signal can only be represented within a certain range and accuracy. The capability of the hardware determines the number of bits that can be used to represent the sample. This in turn affects the possible noise level and amplitude range, which are important factors in the quality of the resulting audio (Roads et al., 1996). The more bits used to describe a sample, the less the quantization noise and the more detailed and accurate the signal resolution. Compact discs use a 16-bit number to represent a sample (Roads et al., 1996).

### *Spectrum Analysis*

By analyzing the sequences of samples and the data contained therein, information can be extracted and plotted on a graph as a spectrum. A general definition of spectrum is “a measure of the distribution of signal energy as a function of frequency” (Roads et al., 1996, p. 536). More simply stated, spectrum is “the combination of the frequencies and their amplitudes that are present in a sound” (Dobrian, 2005, p. 17). Spectrum (or spectral) analysis refers to evaluating the balance among the components of the spectrum. Spectrum analysis is the first step in studying electronic music for applications such as automatic transcription, pitch tracking, rhythm recognition, and sound re-synthesis. “Spectrum analysis reveals the characteristic frequency energy of instrumental and vocal tones, thus helping to identify timbres and separate multiple sources playing at once” (Roads et al., 1996, p. 536 -537).

Strategies for spectrum analysis fall into two basic categories: static, which is like a snapshot of a spectrum, and time-varying, which is akin to a motion-picture film of a spectrum over time (Roads et al., 1996). Since I dealt with a real-time application of such data in the visualizer, here I will only focus on real-time strategies. The most common way to display a time-varying spectrum is to plot a sonogram or spectrogram. Sonograms show the frequency versus time content of a signal (Fig.13). Frequency is the vertical element, while time is the horizontal.

Figure 13. A sonogram. [Roads et al., 1996, p. 543].



*The Fourier Spectrum Analysis Model*

The most common approach to spectral analysis uses Fourier analysis techniques. These are a family of methods that represent sounds as sums of harmonically related sinusoids (Roads et al., 1996). Musical sounds are characterized by harmonic vibrations around a fundamental pitch. In 1822 Jean-Baptiste Joseph, Baron de Fourier, a French engineer, developed a theory of harmonic analysis. In his thesis *Analytical Theory of Heat*, he proposed that complex vibrations could be analyzed as a sum of many simultaneous simple signals. To put it simply, “Joseph Fourier demonstrated that any periodic wave can be expressed as the sum of harmonically related sinusoids, each with its own amplitude and frequency" (Dobrian, 2005, p. 195). Fourier was not researching music, but years later it was found that his theory did indeed apply to it. In 1843 a German scientists named Georg Ohm applied Fourier's theory to acoustical signals. Later another German scientist named H.L.F Helmholtz determined that the harmonic Fourier series of the steady state portion of an instrumental tone largely defined instrumental timbre (Roads et al., 1996).

## Fourier Transforms

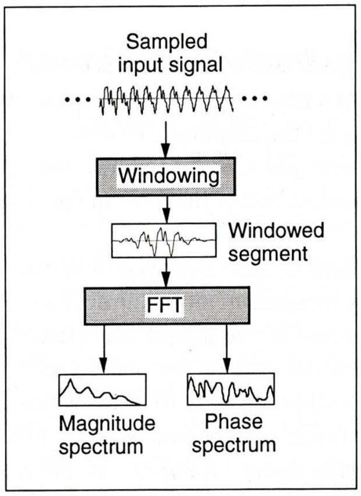
The Fourier Transform (FT) converts analog input signals into corresponding spectrum representation. To expand, the FT "is a mathematical procedure that maps any continuous-time (analog) waveform to a corresponding infinite Fourier series summation of elementary sinusoidal waves, each at a specific amplitude and phase" (Roads et al., 1996, p. 550). Researchers also developed the Discrete Fourier Transform (DFT), "a type of Fourier transform algorithm that can handle discrete-time or sampled signals" (Roads et al., p. 551). According to Dobrian, the DFT "transforms a time-domain representation of a sound wave into a frequency-domain spectrum" (2005, p.195). The Fast Fourier Transform (FFT), developed in 1965 by James Cooley and John Tukey, is simply a more efficient form of the DFT. The Short-Time Fourier Transform (STFT) was later created so that Fourier analysis could be used on sampled, finite-duration, time-varying signals (Roads et al., 1996). Because the STFT can handle such signals, it is an important tool in digital spectrum analysis.

## Windowing

The SFTF makes use of a technique called windowing. It first breaks the input signal into a sequence of ‘short-time’ segments called windows. These windows are usually only 1 millisecond to 1 second long, and can be discrete within the sequence or overlap. Windowing the signal allows each segment to be analyzed separately. This sequence of measurements makes up a time-varying spectrum (Roads et al., 1996, p. 550-551).

The STFT next applies the FFT to each window. Most real-time STFT applications use the FFT on each windowed segment because it is so efficient. The output of the transform, as illustrated in (Fig. 14) is a discrete-frequency spectrum, or “a measure of energy at a set of specific equally spaced frequencies” (Roads et al., 1996, p. 551). Each resulting block of data is called a frame and contains two things: a magnitude spectrum that shows the amplitude of all the analyzed frequency components, and a phase spectrum showing each frequency component’s initial phase value (Roads et al., 1996).

Figure 14. The results of windowing. [Roads et al., 1996, p. 551].



Separating the input signal into windows has three main advantages. First, windowing is useful for the visualization of the spectrum. Limiting the analysis to short windows makes the analysis more accurate because each window analysis plots fewer points than one long analysis of a signal would. Second, using short-time windows is more memory efficient. A long signal would require a large amount of RAM space to hold it while the FFT is calculated for the entire thing. Windowing conserves memory because it is much easier to calculate the FFT on one small segment at a time. Third, windowing results in faster results. If the window is only ten milliseconds long, then the resulting FFT calculation will be returned very quickly even as input is read in. In this way windowing enables applications to perform near real-time spectrum analysis. (Roads et al., 1996).

### *Applications of Fourier Analysis*

A number of studies have used Fourier and other spectral analysis methods to extract musical data for various purposes including annotation, genre classification, and mood detection. Princeton researchers George Tzanetakis and Perry Cook (1999) devised a system that uses the musical features derived from spectral analysis to automatically segment audio based on sound events. Their scheme uses windowing to calculate features every twenty milliseconds. The means and variances of the features are calculated in one second windows. Five features are found: Spectral centroid, spectral rollof, spectral flux, zero crossing, and the root mean-square (RMS). Spectral centroid is the balancing point of the spectrum. Spectral rolloff is a measure of the 'skewness of the spectral shape. Spectral flux is the difference between the magnitudes of two successive frames of the normalized STFT spectrum. Zero crossing correlates to spectral centroid and is the number of time-domain zero-crossings. RMS refers to the measure of the loudness of each window, which may signify new sound events (Tzanetakis & Cook, 1999).

Tzanetakis and Cook, along with Georg Essl, also utilized spectral analysis to automatically classify musical genre. Genre classification is rather subjective and arbitrary, but certain perceptual features dealing with texture, instrumentation, and rhythmic structure can be used to characterize music (Tzanetakis, Cook, & Essl, 2001). This system makes use of STFT to calculate 40 analysis windows of twenty milliseconds each. The means and standard deviation of the features from the windows are calculated to make up a one second 'texture' window. In each of the 20ms analysis windows five features are extracted: centroid, rolloff, flux, zero crossing, and low energy. Centroid here is the mean of the amplitude spectrum. Rollof is a measure of spectral shape. Flux is a measure of spectral change. Zero crossings here refer to the signal's time domain zero crossings. The low energy feature is "the percentage of 'analysis' windows that have energy less than the average energy of the 'analysis' windows over the 'texture' window" (Tzanetakis, et al., 2001, para. 7).

Dan Yang and WonSook Lee (2004) of the University of Ottawa developed a system called Emo to determine the emotional content of a musical piece. They used Fourier methods to find timbral features such as spectral centroid, spectral rollof, spectral flux, and spectral kurtosis (a term that is undefined in the paper) as well as non-Fourier analysis techniques that yielded estimated features such as beats per minute (BPM). While the previously mentioned studies used comparisons of the extracted musical features to classify the sounds, Yang and Lee instead applied their musical features to a mood model to further define musical character.

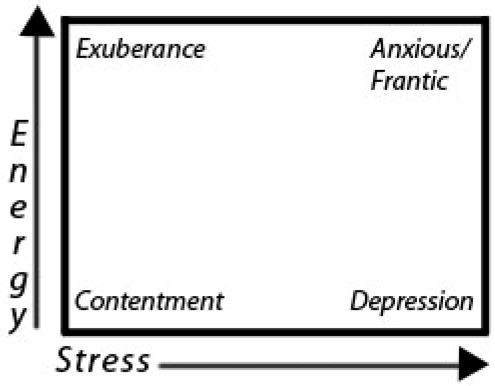
## Mood Models

Musical emotion has most commonly been classified by listener's verbal reports of what feelings a particular piece of music evokes in them. Mood classification is a difficult subject in and of itself. "There is not a standard mood taxonomy system accepted by all currently" (Liu, Lu, & Zhang, 2003, para. 6). Many of the proposed systems use discrete emotional categorization that consists of long and confusing adjective checklists. Such discrete approaches use clusters of adjectives to describe mood, and within these clusters classification can become difficult since words in the same cluster often approximate the same meaning. These checklists also do not take into account mood stimulus, which is an important factor in psychological mood classification of responses (Liu et al., 2003). In other systems emotions are often structured by dimensional ratings, which are single-item test scales (Yang & Lee, 2004). For example, the PANAS-X emotional test scale has a dimensional rating of Positive Affect, so emotions with a perceived positive affect are placed at the top of the scale, while emotions with a low positive effect are placed together near the bottom. The Tellegen, Watson, and Clark mood model uses a pleasant-unpleasant dimension (Yang & Lee, 2004). Dimensional scale ratings have been directly related to musical features in some studies. One such study by Dan Liu, Lie Lu, and Hong-Jiang Zhang, researchers at Microsoft Research Asia, correlated musical features such as tempo, brightness, and loudness to a two-dimensional mood model created by R.E. Thayer.

### *Thayer's Mood Model*

In the late 1990's R. E. Thayer proposed a simple yet specific mood model. Thayer's model is two-dimensional approach based on the theory that mood comes from two factors: Stress, judged on a continuum of happy to anxious; and Energy, judged on a scale of calm to energetic. Mood is divided into 4 clusters, which are Contentment, Depression, Exuberance, and Anxious/Frantic (Fig. 15). Contentment refers to happy and calm, Depression to calm and anxious, Exuberance to happy and energetic, and Anxious/Frantic to anxious and energetic (Liu et al., 2003). This two-dimensional model works well because the clusters are explicit and clearly discrete from one another. These traits, along with the simplicity of the model, make it ideal for computational modeling.

Figure 15. Thayer's mood model. [Liu et al., 2003].



Liu, Lu, and Zhang applied musical features to Thayer's emotional model to determine the music mood. "Music mood describes the inherent emotional meaning of a music clip" (Liu et al., 2003, para. 1). A number of studies have concluded that mode, intensity, timbre, and rhythm are the main elements of use in determining music mood. Because musical mode, or the general musical key of a piece, is very difficult to obtain, Liu and associates only attempted to extract intensity, timbre, and rhythm from their musical data. In their study, intensity corresponds to Thayer's Energy dimension. Both timbre and rhythm correspond to the Stress dimension (Liu et al., 2003).

Liu, Lu, and Zhang downsampled their music clips into a uniform 16000 Hz range, 16 bit mono-channel format. They then divided this into non-overlapping 32ms length frames, and used an octave-scale filter bank to divide the frequencies into 7 sub-bands. From each frame, the means and variances of the timbral and intensity features were calculated across the whole file (Liu et al., 2003). Liu and associates separated rhythm features into 3 aspects: strength, regularity, and tempo. Since bass instruments usually represent rhythm or beat features, they analyzed only the lowest sub-band to extract these features. The spectral information in different sub-bands helped to estimate the timbre of a sound. Liu and associates used both spectral shape features and spectral contrast features. Spectral shape features included centroid, bandwidth, rolloff, and spectral flux, and represent the characteristics of music signals. Octave based spectral contrast features are helpful in music genre recognition, and so here they were used to represent relative spectral distributions (Liu et al., 2003).

In Thayer's model, Energy is easier to compute than Stress. It can be estimated by simple analysis of amplitude, which correlates to intensity. "Intensity is approximated by the signal's root mean-square (RMS) level in decibels" (Liu et al., 2003, para. 3). Liu, Lu, and Zhang calculated the intensity in each sub-band and also the sum of the intensities of the sub-bands. Regardless of the method used to extract it, intensity is essential for mood detection. The Energy measurement for Contentment and Depression is usually much lower than the measurement for Exuberance and Anxiety, so intensity clearly separates the music mood into either the upper or lower half of the mood model.

Once the Energy and Stress of a music clip are established the song can be placed in Thayer's model as a location on the graph of Stress versus Energy. This location determines which of the four mood clusters the song falls into. The emotional classification of the song can then be used to determine artistic characteristics for the generation of visuals.

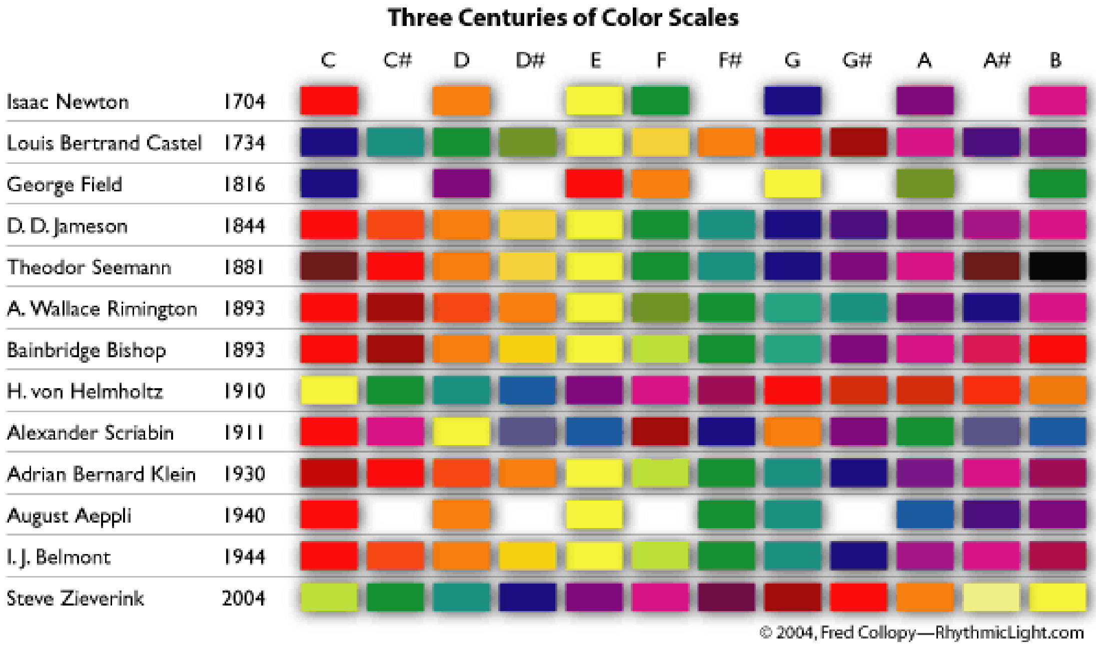
## Mappings

Numerous strategies for mapping sound to visuals have been suggested. In 2005, Randy Jones and Ben Nevile, researchers at the University of Victoria, proposed broad guidelines to assist in the creation of a framework for choosing mappings. Jones and Nevile define mappings as 'transformations' used to convert input parameters to outputs in a different domain" (Jones & Nevile, 2005, p. 55). Mappings can be described based on the number of inputs and outputs as one-to-one, one-to-many, many-to-one, and many-to-many. Since there are so many potential parameters that can be generated from musical data and so many that likewise can be created in visual images, even just the possible one-to-one mappings between them are quite numerous. Therefore it is necessary to develop a framework to determine which mappings are appropriate, how the parameters of the mappings relate to one another, and how the mappings are applied.

Synesthetic mappings are arbitrary and vary from person to person, so they cannot serve as effective guidelines for visual music. However, logical and near-universal mappings based on physics or human perception do exist. For musical frequency (pitch), amplitude (loudness), and timbre (sound color), certain mappings to graphical parameters have a physical basis. For example, smaller objects tend to physically resonate at higher frequencies, so when mapping voice frequency to the size of a corresponding visual object it makes sense to use a scaling factor that maps high notes to small shapes and lower notes to large shapes. Similarly, amplitude of a sound and luminosity of an image map to each other rather naturally because both are measurements of physical intensity (Jones & Nevile, 2005). Other mappings preserve a sense of correlation, but are at the discretion of the visual artist. Musical elements such as tempo or brightness can be correlated to artistic elements like speed or texture. A large number of notes may translate into a large number of shapes.

Jones and Nevile readily admit that color mappings are perhaps the most difficult to determine because color can be quantified in numerous ways that require multiple variables (2005). Through the centuries many scientists, composers, and artists believed in a one-to-one mapping between color and pitch. However, Jones and Nevile counter that "there is no basis for the universality of any one such mapping" (p. 59). Indeed the color-to-note systems developed over the years vary greatly, as shown in Figure 16. Essayist Olivia Mattis agrees with composer Elliot Carter, who proclaimed that the simplicity of a single note to single color analogy is "superficial, if not pointless" (Brougher et al., 2005, p. 213). The meaning of colors can vary widely not only between cultures, but also between individual people within the same culture. Even synesthesia can offer no guidelines here; the mappings of an individual who experiences strong chromesthesia are arbitrary and meaningless to other synesthates (Jones & Nevile, 2005). Jones and Nevile therefore offer no specific guidelines for mapping musical elements to color, but instead stress that whichever mapping is chosen, a consistent internal logic is imperative to convey any sort of meaning. "In general, the choices of cross-modal mappings made by the composer, and their relationships to physically motivated or culturally defined mappings, help define the style of a work of visual music" (2005, p. 57).

Figure 16. *Three Centuries of Color Scales*, Frank Callopy, 2004. [Callopy, 2004].



**Three Centuries of Color Scales**

| | | C | C# | D | D# | E | F | F# | G | G# | A | A# | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Isaac Newton | 1704 | 🟥 | | 🟧 | | 🟨 | 🟩 | | 🟦 | | 🟪 | | 🟪 |
| Louis Bertrand Castel | 1734 | 🟦 | 🟩 | 🟩 | 🟩 | 🟨 | 🟨 | 🟧 | 🟥 | 🟥 | 🟪 | 🟪 | 🟪 |
| George Field | 1816 | 🟦 | | 🟪 | | 🟥 | 🟧 | | 🟨 | | 🟩 | | 🟩 |
| D. D. Jameson | 1844 | 🟥 | 🟧 | 🟧 | 🟨 | 🟨 | 🟩 | 🟩 | 🟦 | 🟪 | 🟪 | 🟪 | 🟪 |
| Theodor Seemann | 1881 | 🟫 | 🟥 | 🟧 | 🟨 | 🟨 | 🟩 | 🟩 | 🟦 | 🟪 | 🟪 | 🟫 | ⬛ |
| A. Wallace Rimington | 1893 | 🟥 | 🟥 | 🟧 | 🟧 | 🟨 | 🟩 | 🟩 | 🟩 | 🟩 | 🟪 | 🟦 | 🟪 |
| Bainbridge Bishop | 1893 | 🟥 | 🟥 | 🟧 | 🟨 | 🟨 | 🟩 | 🟩 | 🟩 | 🟪 | 🟪 | 🟥 | 🟥 |
| H. von Helmholtz | 1910 | 🟨 | 🟩 | 🟩 | 🟦 | 🟪 | 🟪 | 🟪 | 🟥 | 🟥 | 🟥 | 🟥 | 🟧 |
| Alexander Scriabin | 1911 | 🟥 | 🟪 | 🟨 | 🟦 | 🟦 | 🟥 | 🟦 | 🟧 | 🟪 | 🟩 | 🟦 | 🟦 |
| Adrian Bernard Klein | 1930 | 🟥 | 🟥 | 🟧 | 🟧 | 🟨 | 🟩 | 🟩 | 🟩 | 🟦 | 🟪 | 🟪 | 🟪 |
| August Aeppli | 1940 | 🟥 | | 🟧 | | 🟨 | | 🟩 | 🟩 | | 🟦 | 🟪 | 🟪 |
| I. J. Belmont | 1944 | 🟥 | 🟧 | 🟧 | 🟨 | 🟨 | 🟩 | 🟩 | 🟩 | 🟦 | 🟪 | 🟪 | 🟪 |
| Steve Zieverink | 2004 | 🟩 | 🟩 | 🟩 | 🟦 | 🟪 | 🟪 | 🟪 | 🟥 | 🟥 | 🟧 | 🟨 | 🟨 |

© 2004, Fred Collopy—RhythmicLight.com

The previously mentioned Taylor, Boulanger, and Torres used Max/MSP, a sound processing environment, to analyze music and extract pitch, amplitude, vocal timbre, and MIDI chord information. The musical information was then passed into Jitter, ANIMUS, or Virtools, three different visualization environments. In the Jitter environment, chords played on a MIDI piano keyboard directed the color balance of pre-recorded video footage. In this visualization, the chord to color mapping was based on the Circle of Fifths, an important concept in Western musical theory. In this Circle, chords are arranged according to harmonic similarity such that chords that are close together are similar sounding and chords that are distant from one another sound less alike. The chords on the Circle of Fifths are mapped to the colors on the color wheel, and so chords that sound similar musically produce similar colors in the visualization. A further Jitter operation adjusted the color balance of the video to the appropriate colors (Taylor et al, 2006).

Taylor et al. (2006) also mapped vocal timbre to color, again using the Jitter environment. Vocal timbre is defined as “the characteristics of a vocalized sound that make it recognizably different from other vocalizations uttered at the same pitch and loudness” (Taylor et al., p. 4). Differences in the amplitude of each partial frequency measured in the vocalization determine timbre. In this case, the amplitude of the fundamental frequency determined the red value of the color, while amplitude of the second and third partials determined the blue and green components, respectively. The types of sound the vocalist made determined the colors of the visualization. In this case, they applied the changing color values to video footage of a jellyfish (Taylor et al, 2006, p. 4-5).

Instead of attempting to map a color to a musical note or frequency, colors can be related to certain moods. Much study has gone into how colors and their relationships express or can be used to create various emotional effects. Since color is such a complex phenomenon, and one of the most expressive elements of visual imagery, the artistic theories of color deserve special consideration.

## *Color Theory*

Johannes Itten, a leading teacher at the Bauhaus school, developed a detailed theory of color still widely taught in art schools today. His 12-part color circle is the basis for the structure of his theory. Itten's color sphere follows from the color circle and provides a more complete model of color classification.

### The Color Circle and Sphere

The color circle is divided into twelve equal sections as seen in Figure 17. The colors in it follow the sequence of the natural color spectrum seen in a rainbow. The three primary colors—red, yellow, and blue—are configured in an equilateral triangle with three spaces between each. The secondary colors—orange, green, and violet, are also placed in an equilateral triangle that bisects the first so that each secondary color appears in the circle directly in between the two primary colors that compose it. Orange is a combination of red and yellow, green a combination of yellow and blue, and violet a combination of blue and red. These colors must be mixed very carefully so that they appear as pure intermediates and do not lean too heavily toward one primary component or the other. In the spaces left on the color circle are placed the tertiary colors, each of which are the result of mixing a primary with a secondary color. These include yellow-orange, red-orange, red-violet, blue-violet, blue-green, and yellow-green (Itten, 2004). The most important aspect of the color circle is that, according to Itten's color theory, complementary colors are diametrically opposite each other on the circle. The colors are also easily distinct from one another. Using it as a template, the artist can construct works upon logical and objective color principles (Itten, 2004).

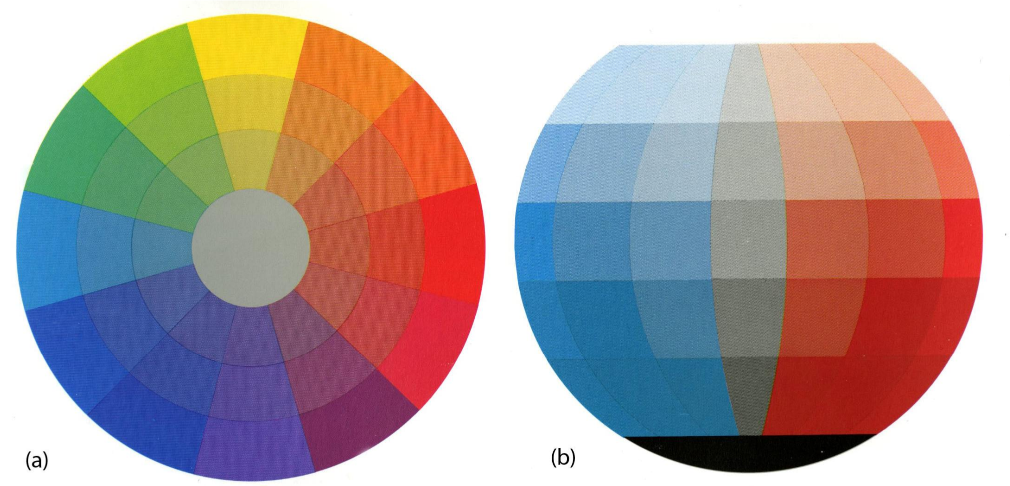
Figure 17. The color circle. [Itten, 2004, p. 35].

The color sphere is the most convenient form for "plotting the characteristic and manifold properties of the color universe" (Itten, 2004, p. 114). Its symmetric form is ideal for visualizing the rule of complementaries, the fundamental relationships between colors, and the interactions between chromatic colors and black and white. All colors possible can be encompassed in the sphere if it is imagined as a transparent solid in which any point corresponds to a color. However for simplicity, Itten's sphere, like his circle, is divided into twelve meridians, and needs only six parallels to adequately classify colors. Figure 18 shows two equatorial views of the sphere from opposite sides. The six parallels create seven zones, with black at the bottom polar zone and white and the top polar zone. The twelve meridians go from pole to pole, and in the middle equatorial zone correspond to the pure colors of the color circle. This makes two zones between each of the poles and the equator section. The zones show the tints, which are colors mixed with an amount of white, and the shades, which are colors mixed with an amount of black. In the two zones between white pole and the equator, two evenly spaced tints of each hue are interpolated, and similarly between the black pole and the equator are two evenly spaced shades of each hue (Itten, 2004).

Figure 18. Equatorial views of the color sphere. [Itten, 2004, p. 116].

Orthographic projections help to visualize the color sphere in two dimensions (Fig. 19). A plan view from the top would show the white pole in the center, the two zones of tints for each hue, and half of the equatorial zone of pure colors. A view from below would show the black pole, two zones of shades for each hue, and the other half of the pure color zone. A horizontal cross-section at the equator [Fig. 19 (a)] shows a central region of neutral grey and an outer ring of pure colors. Between those are two strata of mixed tones created from mixtures of the corresponding complementary colors. A vertical cross-section taken at the red-orange/ blue-green sector, [Fig. 19 (b)], shows that red-orange and blue-green are at their most saturated at the extreme left and right of the equatorial section. Towards the center are mixtures of these complementary colors that meet in neutral grey. The seven equatorial zones tint each of the mixed complements toward white and shade them towards black. As a rule, horizontal sections show the degrees of saturation of hues, and vertical sections show the tints and shades of a given pair of complements in both pure and diluted forms (Itten, 2004).

Figure 19. Orthographic projections of the color sphere. (a) Horizontal cross-section. (b) Vertical cross section. [Itten, 2004, p. 116].

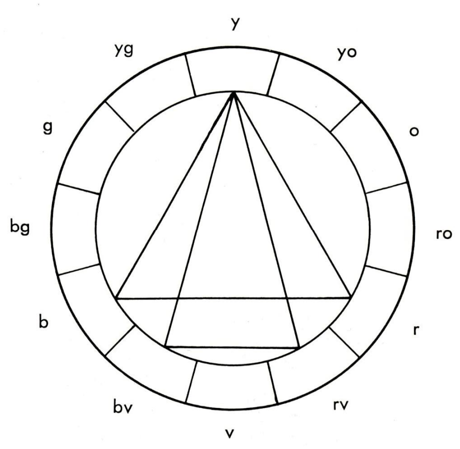


The color sphere is invaluable in Itten’s color theory because not only do opposite points indicate complementary colors as in the color circle, but here opposites indicate complementary degrees of brilliance, or saturation, as well. In addition, paths around or through the color sphere show logical transitions between contrasting colors. Difficulties and possibilities are not destroyed by this simple order. Instead, it serves as a starting point for understanding color (Itten, 2004).

## Color Chords

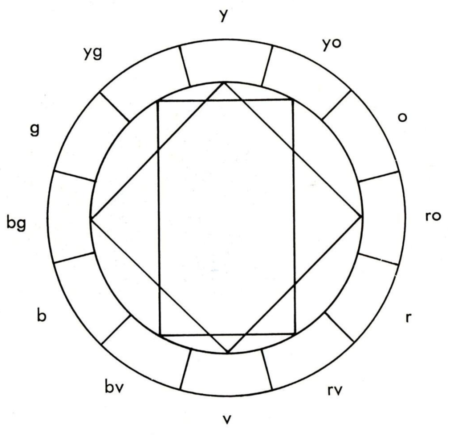
The color circle and sphere are also central in developing systemic color harmony for color composition. Itten states that, "The idea of color harmony is to discover the strongest effects by correct choice of antitheses" (Itten, 2004, p. 119). He therefore used various connections of opposites in the circle and sphere to create his own 'color chords'. His color chords are formed from two or more tones and are termed dyads, triads, tetrads, etc depending on number. Harmonious dyads are composed of two diametrically opposed complementary colors, symmetrical about the center of the color sphere. The sphere yields a large number of dyads. Triads are colors whose positions form an equilateral triangle in the color circle (Fig. 20). Yellow/red/blue is the most powerful of the triads. Triads may also be formed from isosceles triangles. This is done by replacing one color of a dyad with its two neighbors. The triangle also may be placed within the color sphere and rotated in any direction. The colors at the three vertices of the triangle will create a harmonious triad of different chromas. The only special cases are when one of the vertices point to white or black. If one vertex is at white then the other two will point to the first shades of a pair of complements. If the vertex is at black it will point to the first tints of a pair of complements (Itten, 2004).

Figure 20. Color chord triads. [Itten, 2004, p. 118].



Tetrads are formed most simply with squares, which relate two pairs of complements in the color circle (Fig. 21). A rectangle may also be used, along with a trapezoid. The trapezoid produces two adjacent hues, and two opposing hues to the left and right of their complements. As with the case of the triads, inscribing the polygons in the color sphere and rotating them yields countless numbers of tetrad themes. Hexads have six vertices. One way to derive a hexad is to simply make a hexagon on the color circle, which will give three pairs of complementary colors. There are only two such hexads on the color circle, though the hexagon can be rotated in the color sphere to create different hexads of tints and shades. The other method of hexad generation is to place a square in the equatorial zone of the color sphere to create a tetrad, then join each vertex of the square to white and black. A rectangle may be used instead of a square. Other chords such as pentads can also be produced by combining a triad triangle with white and black (Itten, 2004).

Figure 21. Color chord tetrads. [Itten, 2004, p. 118].



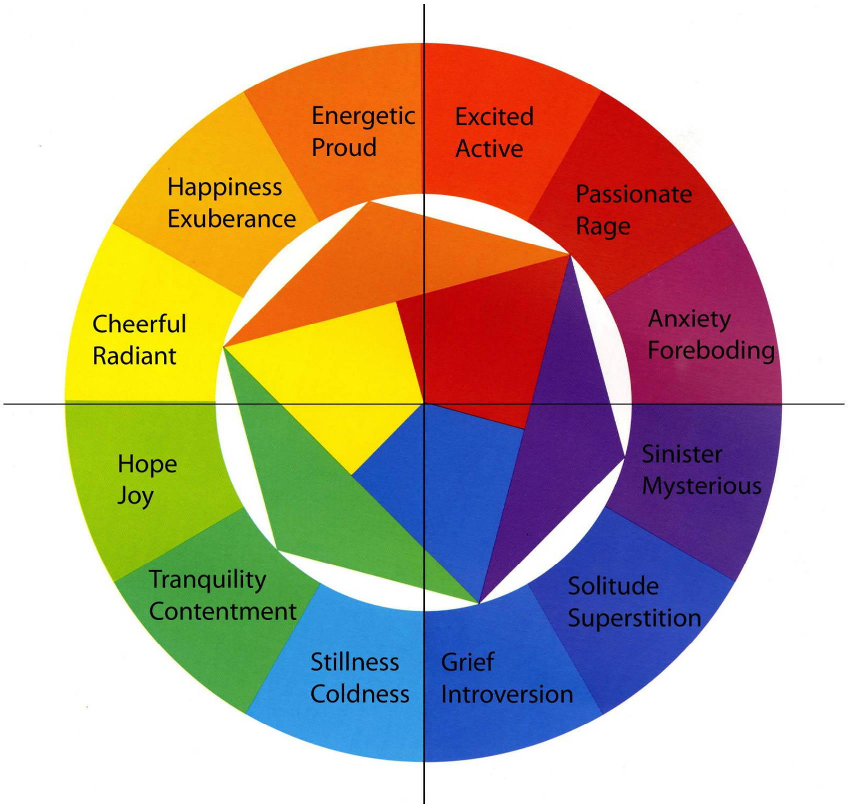
After a discussion of these harmonies, however, Itten is quick to emphasize that the chords chosen for the basis of a composition cannot be chosen arbitrarily. They must be governed by subject matter, theme, and intended execution (Itten, 2004).

## Color Expression

Johannes Itten also developed a theory of color expression. He characterized each of the pure primary and secondary colors on the color circle with certain emotions, based on their hue temperature and energy. As a general rule, Itten states that, "all tints represent the brighter aspects of life, whereas shades symbolize the dark and negative forces" (Itten, 2004, p. 137). Yellow, the brightest of the pure colors, symbolizes understanding, knowledge, heaven, and truth. It is especially radiant and cheerful when contrasted with dark colors. However when darkened or desaturated it becomes sickly. Red is a radiant and very flexible hue. Itten believed that it has the ability to express all emotions between the infernal and the sublime. He likens it to blood, passion, evil, warmth, and love. Red-orange is the warm-pole of the color circle. It has the most energy of all the colors. It signifies fiery strength, feverishness, belligerence, and passion. Red is always warm. Blue, on the other hand, is always cold. Itten states that it is a very spiritual color, introverted and infinite. When dimmed, its expression also darkens into fear, grief, and perdition (Itten, 2004).

Green's emotional effect changes depending on whether it leans more towards blue or yellow. Towards yellow, it conveys youth, nature, hope, and joy. Towards blue it becomes coldly aggressive. Blue-green has the coldest temperature and least energy. It is the cold-pole of the color circle. A pure green communicates fruitfulness, contentment, tranquility, and hope. If green is dulled then it also descends into sadness and decay. Orange is the maximum in radiant activity. It expresses pride and external ostentation. When tinted towards white it loses its character, and when shaded towards black it withers to a dull and taciturn character. However, these dull shades can be warmed to soothing browns and beiges. Violet is characterized as the color of the unconscious. It is a mysterious color and can convey menace or encouragement. It has an impressive and imposing nature, and can be oppressive. When darkened or dulled it becomes superstition. A dark violet can be terrifying, but a light violet becomes delicate, lovely, and enchanting (Itten, 2004). The color expressions are labeled on the color circle in Figure 22. Rotation around the circle follows a certain logic; as the colors become darker and cooler so to do the emotions. As they gradually return back to warmth and radiance their expressions do too. This flow enables Itten's color expression model to be mapped directly onto Thayer's aforementioned mood model.

Figure 22. Color expressions mapped onto the color circle.



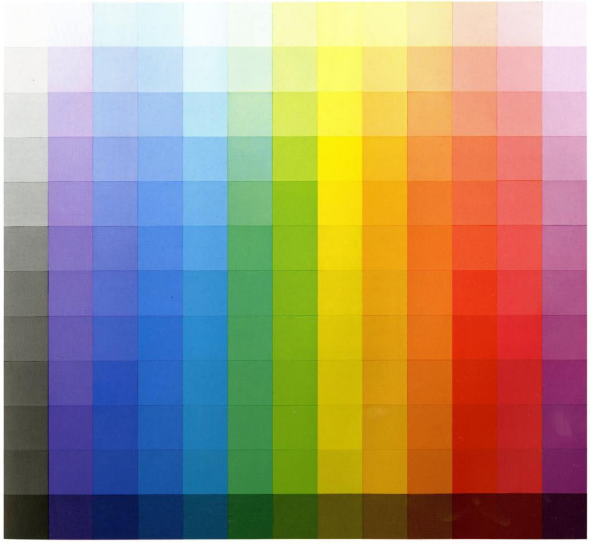
<u>Color Expression and Thayer's Mood Model</u>

The four mood clusters in Thayer's model are analogous to the four quadrants of the Cartesian coordinate system in two-dimensions. The twelve hues of Johannes Itten's color circle can also be grouped into four quadrants. Since each of these twelve hues has a general emotion or mood associated with it, the color emotions will also follow Thayer's mood model.

Colors in the circle can be divided into two equal groups based on color temperature. The Energy dimension of Thayer's model corresponds to this split. The warm colors— yellow, yellow-orange, orange, red-orange, red, and red-violet —are higher in perceptual temperature and so more energetic than the cool colors yellow-green, green, blue-green, blue, blue-violet, and violet (Itten, 2004). The warm colors relate to the high-Energy mood clusters and therefore the yellow to red-violet half of Itten's color circle is analogous to the upper half of Thayer's model, or quadrants 1 and 2. The yellow-green to violet half of the color circle is equivalent to the low-Energy half of Thayer's model, or quadrants 3 and 4.

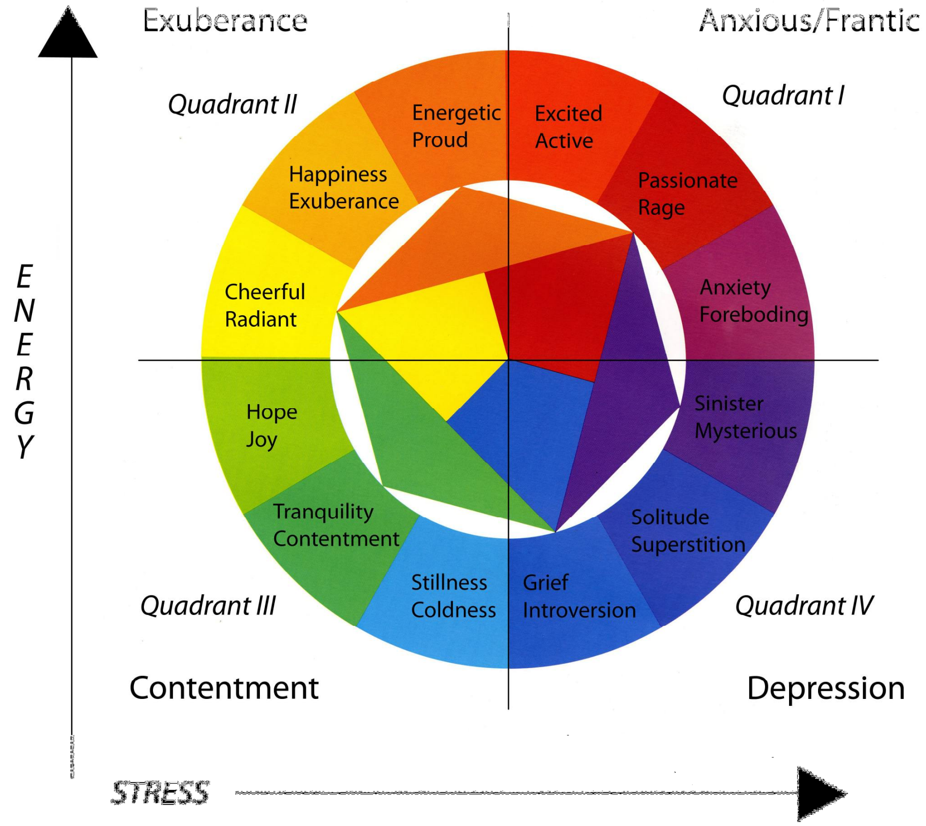
Ittens' color circle can be divided into a relatively lighter and a relatively darker group of colors. Thayer's Stress dimension corresponds to this chromatic light-dark split. The circle assumes that all of the colors are at their purest brilliance. When the hues of the color circle are tinted or shaded to match the brilliance of twelve equidistant steps from white to black, each hue is at a different step (Fig. 23) (Itten, 2004). Pure red (step 8), violet (step 10), and blue (step 9) are at low levels, with only a few steps on the way to black. Yellow (step 4), green (step 6), and orange (step 6) are at high levels, close to white. Yellow assumes a light overall character while red or blue evoke a character of overall darkness. From this the color wheel can be split along its center vertical line into a light orange-yellow-green half that is equivalent to quadrants 1 and 4, and a dark red-violet-blue half that is equal to quadrants 2 and 3.

Figure 23. Tints and shades of the color circle. [Itten, 2004, p. 55].



Inscribing the color circle into the square of Thayer's mood model illustrates the mood/color connections. The hues in quadrant I all fall under Thayer's Exuberance cluster, as do their corresponding moods; the hues and moods in quadrant II fall under Anxious/Frantic; hues/moods in quadrant III are contained in the Depression cluster; hues/moods in quadrant IV all fall under the cluster heading of Contentment (Fig. 24). Because these two systems correspond so well, polar coordinates can be used to map a position in Thayer's model onto a hue in Itten's model to establish the general mood and mood color of the music. When the mood of a song is plotted on Thayer's model, polar coordinate formulas can be used to find that position's distance and angle from the center of the model. Finding that same angle on the color expression circle will correlate the mood of the music to a general mood color. These two classifications can so be used to provide the basis for the general look and actions of the generated visuals.

Figure 24. Color expression circle inscribed onto Thayer's mood model.



# CHAPTER V

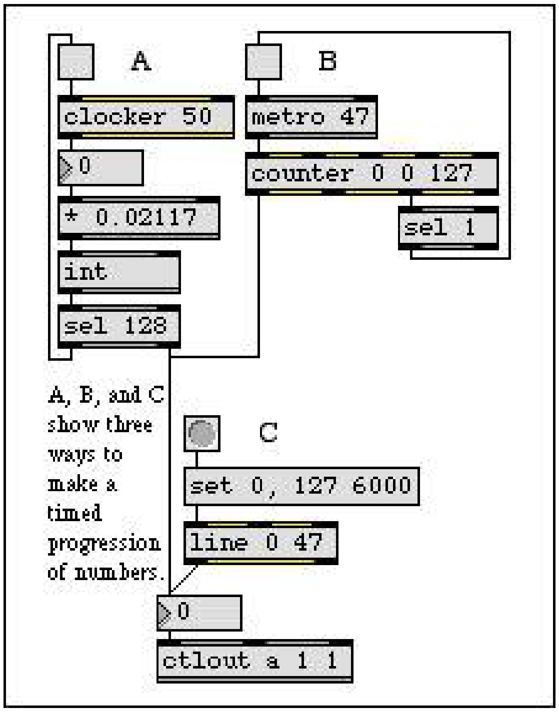
# IMPLEMENTATION

## Max/MSP/Jitter

The Cycling ’74 software package *Max/MSP/Jitter* is a three-tiered visual programming environment especially suited to musical processing and multi-media applications. Each of the three applications serves a specific purpose. Cycling ‘74’s website describes *Max* as “ a graphical programming environment that provides user interface, timing, communications, and MIDI support”, states that the purpose of *MSP* is “for real-time audio synthesis and DSP (Digital Signal Processing)”, and further says that *Jitter* is “for video and matrix data processing” (Cycling ’74, n.d.). Data is easily passed between the three modules such that information from one can be used to control processes in another. I decided to use Max/MSP/Jitter because it offered all of the functionality I needed and greatly simplified the process of passing data from an audio space into a visual space.

Max, as it will be called from this point, contains a large number of *objects* that perform various functions. These objects appear to the user as boxes with texts or graphics in them. The first word of text in an object box identifies the task the object will perform. Objects pass information to one other by the use of *patch cords*, which connect the objects. Object representations have small black bars at the top and bottom. The bars at the top are inlets for receiving information, and the bars at the bottom are outlets for sending information (Zicarelli & Taylor, 2006). Max programs, called *patches* or *patchers*, contain groups of connected objects that together perform a desired task. Figure 25 illustrates a simple Max patch. Several patches can also be connected to form larger, more complex Max programs. A large Max patch might also contain *subpatches*, which are patches within a patch that execute additional calculations or analysis. Subpatchers are saved within and open with their parent patch. *Abstractions*, on the other hand, are patches within a patch that are saved separately and can run independently of the containing patch. Max programs can contain a number of patches, subpatches, and abstractions to carry out their complex tasks.

Figure 25. A simple Max patch. [Cycling '74, 2007].



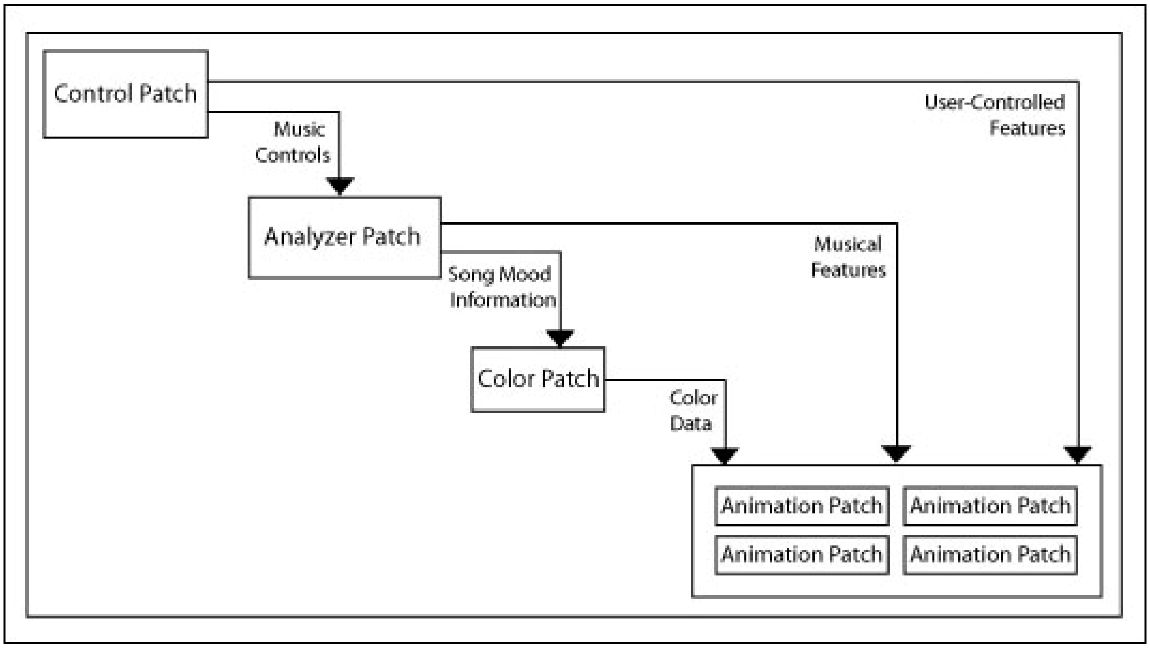
## Visualizer Patches

There were four main tasks required in the creation of the visualizer: analysis of musical data, color determination, image creation, and interactivity control. To facilitate these tasks, I created seven Max patches that pass information between each other. The Control patch handles user interactivity, the Analysis patch performs the musical data analysis, and the Color patch determines song-appropriate colors. There are four separate Animation patches, each of which creates a unique animation scheme controlled by the music.

The Control patch is the top-most patch. It is the only patch that the end user is able to engage. It includes the user interface and a number of subpatches that handle receiving and sending user commands. These commands specify which song file to play, which animation scheme to visualize, and optionally which colors to use in the visualization. There are additional commands specific the animation schemes. The music-related data is sent into the Analysis patch, which is contained as an abstraction within the Control patch. The Analysis patch analyzes the musical signal to extract features such as note attacks and volumes, as well as the overall mood of the song file. It too has a number of subpatches, and contains the Color and Animation patches as abstractions. The Analysis patch sends information into the Color patch. With the help of many subpatches, the Color patch determines the appropriate colors based on the song mood.

The Control patch, the Analysis patch, and the Color patch all send information to the four separate Animation patches. The Animation patch scheme selected by the user receives and then visualizes the data as a directed animation. Specifically, the Analyzer patch sends the musical features into the Animation patches to control their visual elements such as motion, timing, and scale. The Color patch sends the colors it determines from the mood analysis and color expression model into the Animation patches to control the colors used in the animations. The Animation patches each have a specific way of applying all of this received data to parameters of shape, form, color, and configuration to produce four different animation schemes. The general information pipeline of the visualizer is shown in Figure 26.

Figure 26. The Max pipeline of the visualizer program.



### *Control Patch*

The Control patch is the entrance to the visualizer. Its main function is to serve as an intermediary between the visualization program and the user. The patch directs user input to the appropriate patches. While a variety of calculations occur in this patch, only the user controls are visible to the end-user.

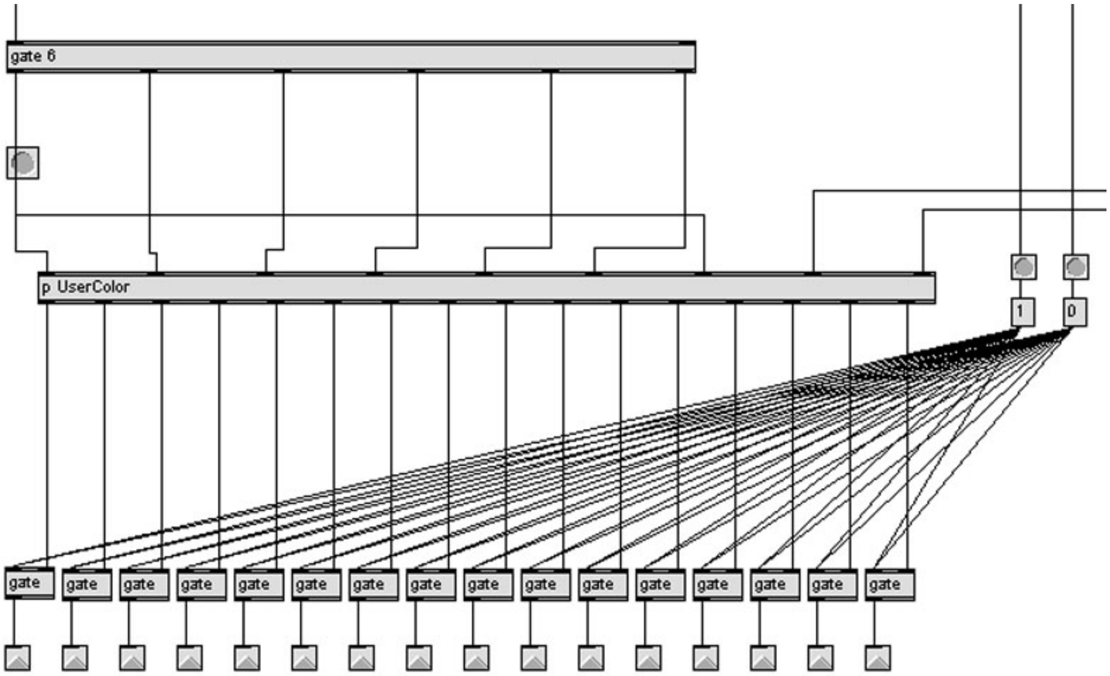
When the program is opened, a *metro* object within the Control patch turns on. *Metro* sends an alert, or a *bang*, every two milliseconds that tells various objects to perform their operations. Generally, the *metro* object is used to ensure that certain operations, like window redrawing or sample clock ticks, are constantly working. The *metro* output is sent into the Analyzer and Animation patches. Also immediately upon opening a *bang* is sent to the Analyzer that enables all of the signal analysis processes.

To start the program, the user selects a song file from a list. When the user presses the start button, the song file name is sent into the Analyzer patch. Playback commands such as stop, pause, and volume, are also included in the visible portion of the Control patch. These commands are sent to the Analyzer patch as well. The Control patch also has a progress bar to report the elapsed time of the song to the user. The song duration and elapsed time are received from the Analyzer patch.

Color picker controls let the viewer choose the colors used in the visualization. When the color picker is activated by the user, a message is sent to a *gate* object. The *gate* object is a traffic controller that can be used to direct whether and where information is sent. When the color controls are turned on, *gate* allows the color data from the color picker controls to be sent on to the Color patch, as shown in Figure 27.

This data overrides the color information in the Color patch so that the user's desired colors are used in the visualization instead. When the color picker controls are turned off, *gate* blocks the user color information and all of the colors come from the automatic calculations in the Color patch.

Figure 27. *Gate* objects in the Control patch. The boxes with triangles are *outlet* objects that send the color data directly into the Color patch.



The subpatch *UserColor* performs a number of calculations to create a palette with sixteen colors for use in the animation schemes. *UserColor* generates variations of each of the user's chosen colors. The methods used within the subpatch are very similar to the color generating methods located in the Color patch and its subpatches. These calculations will be discussed within the sections describing those patches.

The Control patch handles passing all of the data into the appropriate Animation patch. The user is able to choose from a drop down list one of four different animation schemes to be displayed. This allows the viewer to select the many different ways that the music may be manifested as images by this program. These four schemes are based on flocking simulations, particle simulations, spring physics, and spline calculations. The schemes will be further described in the Animation Patches section. Certain parameters affecting the images in the animation schemes are also subject to user control. The user may choose the shape or drawing type displayed in the flocking and particle scheme visualizations. In addition, the user can also select pictures from a predefined folder to serve as textures on the shapes of the flocking and springy schemes.

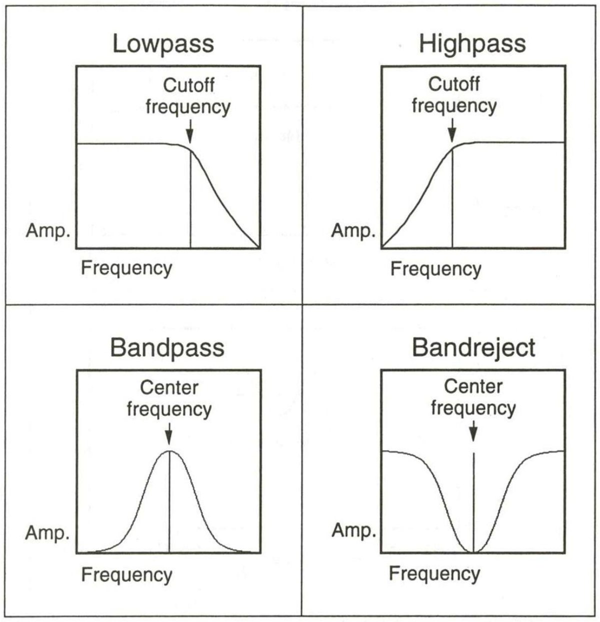
### *Analyzer Patch*

The Analyzer patch generally functions to play back and perform processing on the audio signal of the chosen song file. Many of the objects in this patch have a tilde following their name. The tilde simply designates these as MSP objects that deal with digital audio signals. Yellow patch cords indicate an audio signal being passed between objects.

When the Control patch is opened, it sends a *bang* to an object within the Analyzer patch called *dac~* that immediately turns on Max's built in digital-to-analog converter (DAC). When the Analyzer patch receives a song file name from the Control patch, the *sfplay~* object reads in the song file's digital audio information. It sends the data to the DAC, which converts the digital song file into an analog signal and plays the analog audio through the computer's speakers.

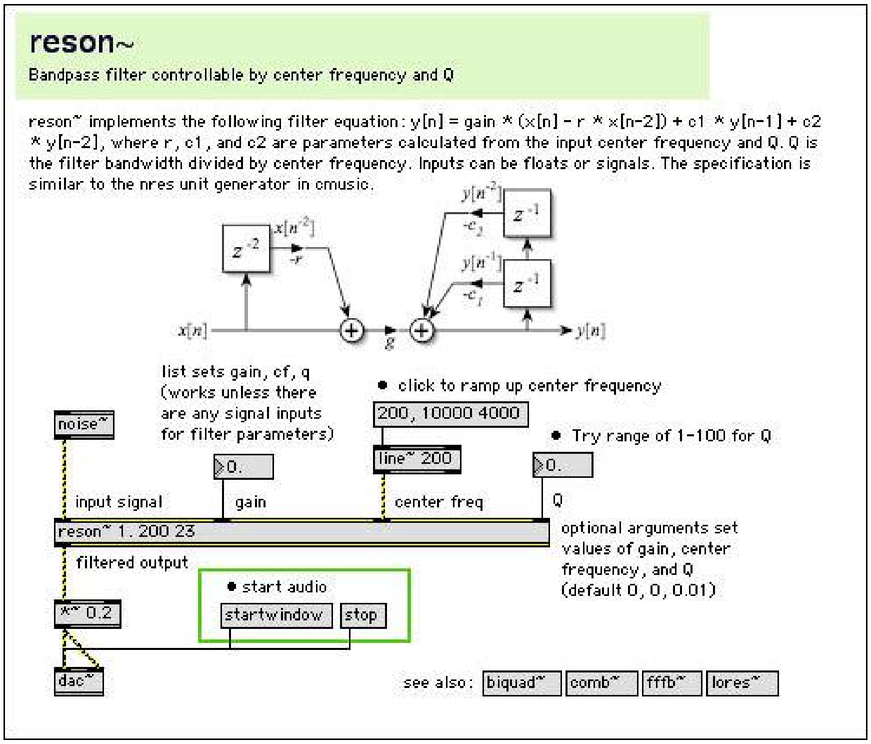
The *sfplay~* object also sends the audio signal out to five separate *reson~* objects. *Reson~* is one of a number of filter objects included in Max. In sound analysis, filters are operations on a signal that boost or reduce selected regions of the frequency spectrum (Roads et al., 1996). The frequency response curve of a filter shows amplitude-versus-frequency. The peaks of the curve represent areas where the signal is boosted, and troughs represent areas of the signal that are reduced (Roads et al.). Figure 28 shows frequency response curves for four common types of filters.

Figure 28. Filter frequency response curves. [Roads et al., 1996, p. 188].



*Reson~* is a bandpass filter with by two parameters, center frequency and Q. Figure 29 shows the *reson~* help file.

Figure 29. The *reson~* object. [Cycling ’74, 2007].



reson~

Bandpass filter controllable by center frequency and Q

reson~ implements the following filter equation: y[n] = gain * (x[n] - r * x[n-2]) + c1 * y[n-1] + c2 * y[n-2], where r, c1, and c2 are parameters calculated from the input center frequency and Q. Q is the filter bandwidth divided by center frequency. Inputs can be floats or signals. The specification is similar to the nres unit generator in cmusic.

list sets gain, cf, q (works unless there are any signal inputs for filter parameters)

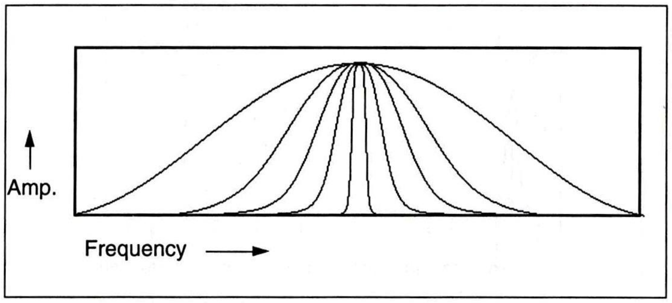
● click to ramp up center frequency

● Try range of 1-100 for Q

optional arguments set values of gain, center frequency, and Q (default 0, 0, 0.01)

In a bandpass filter, the center frequency is the point of maximum amplitude and determines which particular frequency is boosted. The bandwidth is the area of the response graph between the maximum amplitude point and the low cutoff frequency. Q is “the filter bandwidth divided by the center frequency” (Roads et al., 1996). When the center frequency is constant, adjusting the Q is the same as adjusting the bandwidth of the filter. A high Q value focuses the frequency response sharply around the center frequency. Lower Q values increase the bandwidth of the response graph so that more frequencies are boosted (Roads et al.). Figure 30 shows a bandpass filter with various Q values. *Reson~* also has another parameter, gain. Gain is simply the amount of boost the frequency band receives.

Figure 30. A bandpass filter with various Q values. Lower Q values create a wider curve. Higher Q values create a narrower curve. [Roads, 1996, p. 191].



Rather than dealing with the entire audio signal spectrum, I divided it into ranges that could each be analyzed separately. The five *reson~* filtering objects each isolate a different area of the signal. According to the 'Mixing Engineer's Handbook', there are six distinct ranges of the audio bandwidth. The Sub-Bass ranges from 16 – 60 Hz. The Bass range is from 60 – 250 Hz and contains the fundamental notes of the rhythm section. The Low-Mids, which are the low-order instruments, range from 250 – 2000 Hz. The High-Mids range from 2000 – 4000 Hz and encompass the usual vocal range. Presence is located between 4000 and 6000 Hz and provides the clarity and definition of voices and instruments. Brilliance is between 6000 and 16000 Hz and contains sibilance and very high pitches (Owsinski, 2006).

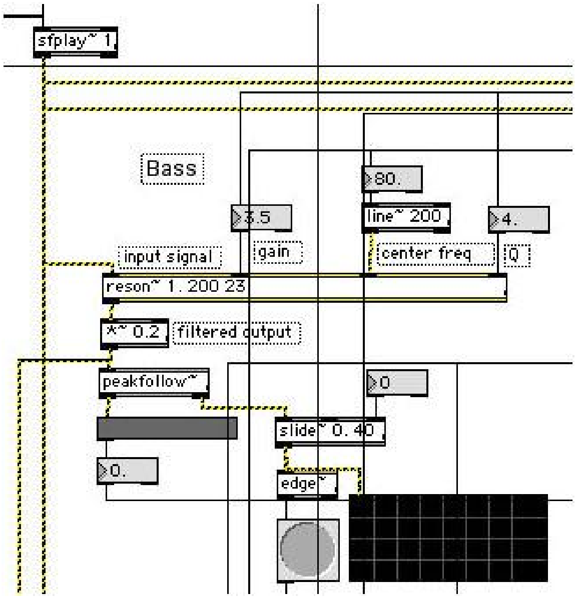
For simplicity, the Bass and Sub-Bass ranges were combined into one Bass range of 16 –250 Hz. Experimentation showed that the most effective center frequency for this overall Bass range was 80 Hz, and the most effective Q value for isolating bass frequencies was 4. To isolate the Low-Mid range, the *reson~* filter worked best with a center frequency of 1500 Hz and a Q of 7.5. The High-Mid range was accentuated most effectively with *reson~* at a center frequency of 3150 Hz and a Q of 7.5. Emphasis of Presence used *reson~* with a center frequency of 5500 Hz and a Q of 5. Brilliance emphasis worked best when *reson~* used a center frequency of 8000 Hz and a Q of 8.

The five filtered signals from the *reson~* objects are each amplified by a signal multiplier and then passed into *peakfollow~* objects. *Peakfollow~* is one of a set of outside objects, called externals, written by third parties. *Peakfollow~* was developed by an audio-visual artist who simply goes by the name Jasch. It, along with a number of other externals, is available at the website http://www.jasch.ch/. *Peakfollow~* tracks the audio signal and illustrates it as a continuously moving graph of amplitude versus time (Jasch, 2008).

When it detects a peak, it interpolates a straight line from its current amplitude value up to the peak amplitude value, then decays exponentially until another peak in the signal is detected. *Peakfollow*~ sends this modified signal to its left output. The numerical values of the *peakfollow*~ signal are in a range from zero to one. These values represent a relative volume for each of the five peaks and are sent into the Animation patches to help control the generated visuals.

The right output of *peakfollow*~ is zero if the signal is falling, and is one if the signal is rising. This data is sent through a *slide*~ object that makes a smooth logarithmic transition between the one and zero values. *Slide*~ smoothes out the signal to prevent it from reporting fast transitions that might result from vibrato instruments. The modified signal is then sent through an *edge*~ object that detects changes from zero to non-zero in the signal. Each time a change is detected, *edge*~ sends out a *bang* message. The edge detection provides a simple but effective method of note attack detection. *Bang* messages from the Bass range indicates rhythmic activity, and as such can be used as a beat tracker. Figure 31 shows the audio signal being passed out of the *sfplay*~ object and through the pipeline of analysis objects from *reson*~ through *edge*~ to detect beat attacks. The graphical objects pictured beside *edge*~ and under *peakfollow*~ are an oscilloscope and a signal level meter, and are only there for observation.

Figure 31. The beat attack detection pipeline. Here the *reson~* object isolates the Bass frequency range.

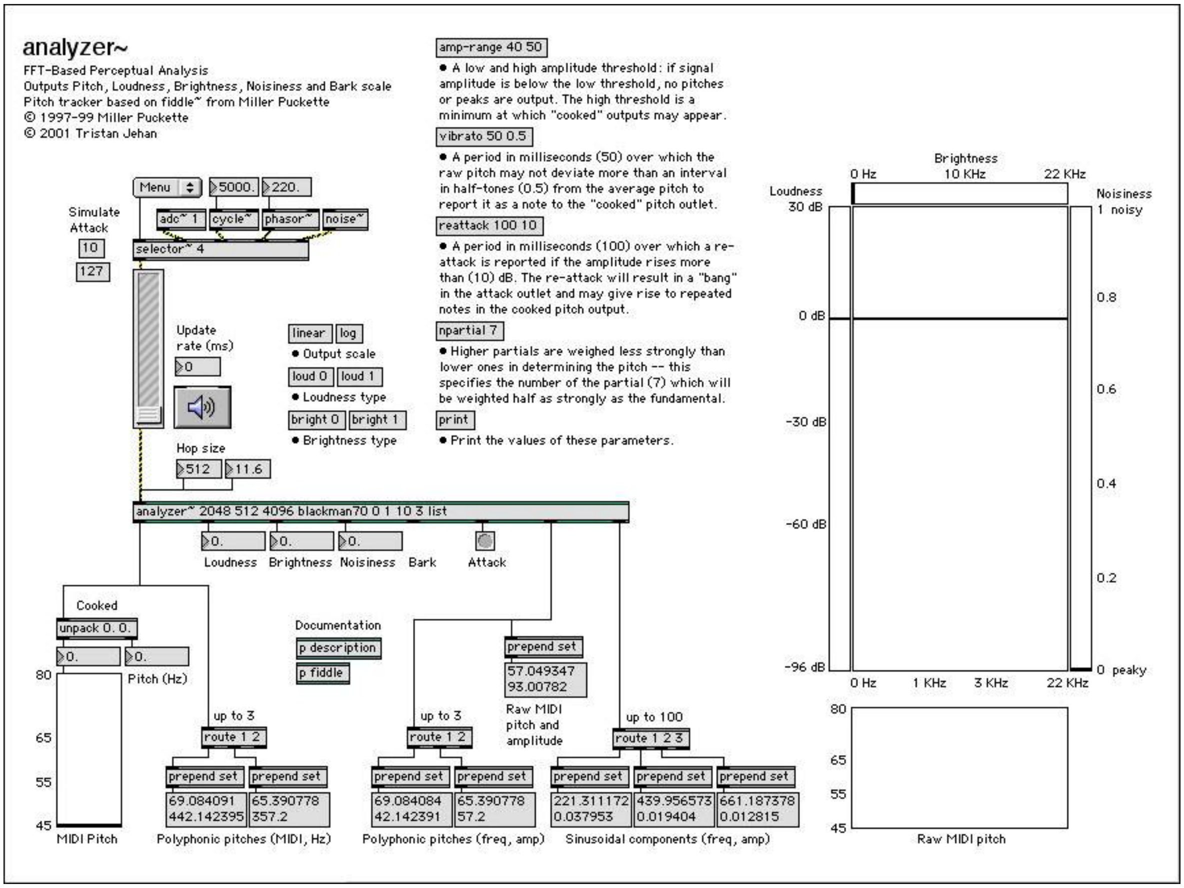


The *edge~* object's attack detection in the bass range is used to determine a rhythmic rate, or tempo for the song. Every four seconds, the beat attacks that occurred within that time interval are counted up and averaged with the note hit counts from the previous twelve seconds. This average gives an approximate song speed. Based on observation of the slowest and fastest song speeds, scaling was performed to bring the song speed into a range suitable for use with the other song data.

The song speed determines the color chord type associated with a song such that slower songs have smaller chords with fewer colors, like dyads or triads, and faster songs have larger color chords like tetrads or hexads that result in more colors. Song speed was scaled to a range of -100 to 100 so that it could also be used to assist in determining the Stress measurement of a song.

The main calculations to extract the Stress and Energy measurements of a song are applied to the unaltered original audio signal. It is sent from the *sfplay*~ object to *analyzer*~ for evaluation. The *analyzer*~ object, (Fig. 32), is an external created by Tristan Jehan. It and other helpful externals are available at his website http://web.media.mit.edu/~tristan/. *Analyzer*~ performs a number of analysis functions based on the Fast Fourier Transform (FFT). It outputs pitch, loudness, brightness, noisiness, a Bark scale, note onset detection, raw pitch with overall amplitude, and sinusoidal decomposition. The pitch tracking outputs and sinusoidal decomposition were not used and will not be described. The note onset detection was not as effective as the *peakfollow*~ and *edge*~ detection method described earlier, and so it was also not used.

Figure 32. Tristan Jehan’s *analyzer~* object. [Jehan, 2001].



analyzer~
FFT-Based Perceptual Analysis
Outputs Pitch, Loudness, Brightness, Noisiness and Bark scale
Pitch tracker based on fiddle~ from Miller Puckette
© 1997-99 Miller Puckette
© 2001 Tristan Jehan

amp-range 40 50
- A low and high amplitude threshold: if signal amplitude is below the low threshold, no pitches or peaks are output. The high threshold is a minimum at which "cooked" outputs may appear.

vibrato 50 0.5
- A period in milliseconds (50) over which the raw pitch may not deviate more than an interval in half-tones (0.5) from the average pitch to report it as a note to the "cooked" pitch outlet.

reattack 100 10
- A period in milliseconds (100) over which a re-attack is reported if the amplitude rises more than (10) dB. The re-attack will result in a "bang" in the attack outlet and may give rise to repeated notes in the cooked pitch output.

npartial 7
- Higher partials are weighed less strongly than lower ones in determining the pitch -- this specifies the number of the partial (7) which will be weighted half as strongly as the fundamental.

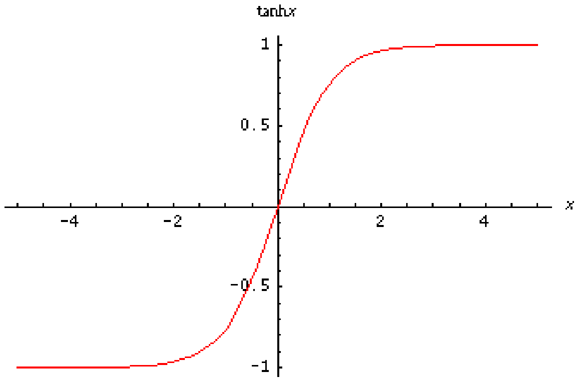
print
- Print the values of these parameters.

In *analyzer~*, brightness is a measure of spectral centroid, or which area of the frequency spectrum has the greatest concentration of high amplitudes. Loudness is a measure of spectral energy, and noisiness is spectral flatness measure (SFM) based on the Bark scale (Jehan, 2001). The Bark scale is a psychoacoustical-based scale of the sound spectrum (Smith & Abel, 1999). Psychoacoustics refers to the psychology of sound and how sound is actually perceived by the human ear and mind (Roads et al., 1996). The Bark scale is based on numerous psychoacoustical experiments that determined the critical bands, or ranges, of human hearing. In the scale, each of the bands has a width of one Bark. Measuring spectral energy (loudness) over the bark scale gives a close correspondence to what the human ear actually hears in the audio signal spectrum (Smith & Abel, 1999).

The *analyzer~* object reports loudness in a range of -96 to 30 dB. Brightness, which corresponds to the frequency of the overall timbre, is measured within a range of 0 to 22050 Hz. I observed averages of the loudness and brightness levels for each of the songs in an experimental song bank. For both measurements the songs stayed within a middle range without actually approaching the extremes of the allowed range. The loudness levels for the actual songs stayed within a range of -30 to 15 db. The brightness levels for the songs stayed within a range of 0 to 4500 Hz.

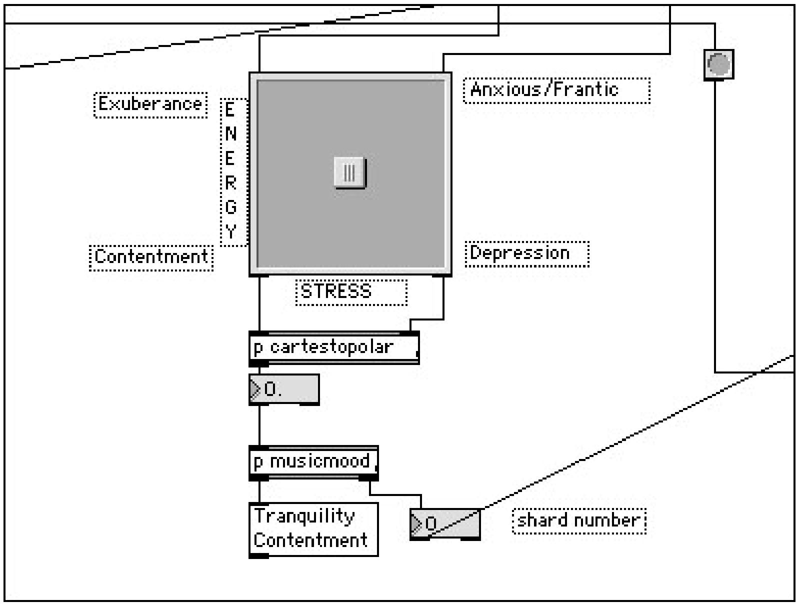
For each song the overall averages of both loudness and brightness were scaled and moved into suitable ranges for calculation. The actual range of song brightness averages was scaled from 0 through 4500 Hz to 0 to 4.5 Hz, then changed to the range of -2.25 through 2.25 Hz so that the median point between the two extremes was zero. This allowed the brightness measurement to be weighted against a hyperbolic tangent function. As illustrated in Figure 33, in a hyperbolic tangent the differences between values nearer to zero are much larger than numbers farther from zero. This means that brightness values near the center of the range are more heavily emphasized than those towards the outside of the range. This ensures that songs are more likely to fall to one side of the graph or the other rather than staying in the center. The actual range of song loudness averages was changed from -30 through 15 db to -2.25 through 2.25 db, then also weighted against a hyperbolic tangent function such that values in the middle areas of the range were amplified more than those at the extremes. Both of these numbers were then scaled to be in the range of -100 to 100. The brightness result was further averaged with the approximate song speed value in a 1:1 ratio to fit with Liu, Lu, and Zhang's assertion that Stress is a function of brightness and speed. The resulting loudness measurement determines Energy, and the resulting brightness and song speed measurements determine Stress.

Figure 33. The hyperbolic tangent function. [Weisstein, 2008].



Once Stress and Energy are determined, they are plotted on a two-dimensional graph with Stress as the horizontal axis and Energy as the vertical axis to produce an (x, y) pair (Fig. 34). The pair is then sent into a subpatch called *cartestopolar* that converts the (x, y) values into polar coordinates; or a radius and an angle value in degrees. The angle value is then sent into another subpatcher called *musicmood* that determines, based on the color expression circle, the musical mood. During the course of the song, the current mood is returned to and displayed in the Analysis patch. The angle value also determines color. The twelve shards of Itten's color circle are each assigned a number, and the *musicmood* patch reports the current color shard.

Figure 34. Energy and Stress measurements plotted on a two-dimensional graph..



The Analyzer patch sends four pieces of musical information to the Animation abstractions embedded within it. These are: the bass frequency beat *bang*, the volumes of the five frequency ranges, the chord type, and the overall volume of the audio signal. These all have some control over parameters in the four different Animation schemes. The Analyzer patch also sends the shard number and the color chord type directly into the Color patch. In addition, the Analyzer patch returns the current time and duration of the song to the Control patch. These two pieces of information are used to control the song progress bar in the Control patch.

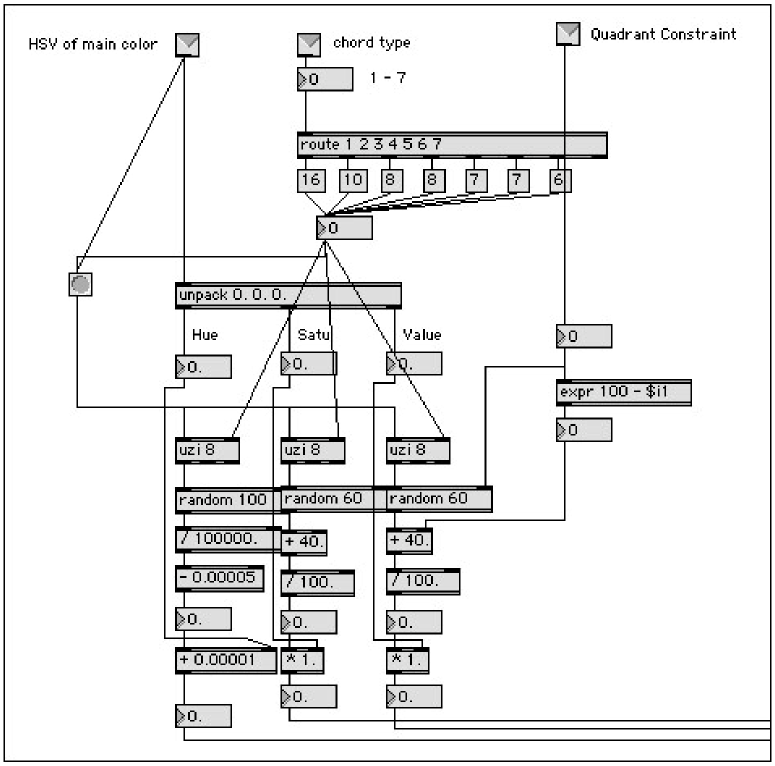
### *Color Patch*

The Color patch produces a color palette based on the music mood. The color generation method of the Color patch only needs two pieces of information, the shard number and the color chord type, to calculate appropriate colors. The shard number maps the song mood to its corresponding color and so specifies the main color of the visualization. The color chord type indicates how many additional colors the visualization will have. It also determines the proportions of the different colors to be used. Using shard number and chord type, the Color patch determines from one to six colors that are appropriate for the visualization. Further calculations produce additional variations of each color. In all, 16 colors are generated and returned to the Analyzer patch as a palette for use in the visualization.

When a shard number is determined, it is sent into an abstraction called *ColorSup*. Since Itten's color circle only contains twelve shards, the Hue-Saturation-Value (HSV) values of each shard color were hard-coded into *ColorSup*. HSV representation of the colors was used instead of the more common Red-Blue-Green (RGB) representation so that I could adjust saturation and value to better fit the perceived song mood. A new shard number causes the corresponding shard HSV values to be returned to the main Color patch. These HSV values provide the main color for the resulting visualization.

The main color returned to the Color patch is sent into the *maincolor* subpatch, shown in Figure 35. This subpatch generates variations of this color to create visual interest in the visualization. The variations ensure that objects in the visualizer do not blend into one another because they are all the exact same color. Small values are added to the initial Hue, Saturation, and Value values to produce sixteen slightly different variations of the main color. The random values added to Hue range from -0.0005 to 0.0005. This ensures that the random hue only deviates very slightly from its initial value, keeping the resulting colors from being perceived as too different from the original. The randomness applied to the saturation and value numbers is constrained so that the resulting colors are not too dark, too light, or too grey. The randomness amount is determined by the music’s mood quadrant. In the Depression and Contentment quadrants, randomness is constrained so that saturation and value additions make the colors duller and darker. In the Exuberance and Anxiety/Frantic quadrants, the constraints allow the colors to be brighter and lighter.

Figure 35. The *maincolor* patch. It receives the HSV values, chord type, and constraint and produces random variations of the color. The HSV color variations are converted to RGB values and sent back to the Color patch.



The sixteen modified HSV values are each sent into an *hsvtorgb* object, used to convert the HSV values into RGB values. The sixteen variations of the main color are returned to the main Color patch.

Once the main color is determined, additional colors depend upon the chord type. There are seven different chord types, each corresponding to a chord number: 1- monochromatic, 2- dyads, 3 - equilateral triads, 4 - isosceles triads, 5 - square tetrads, 6 - rectangular tetrads, and 7 - hexads. The monochromatic chord only uses the main color. For the other types, colors are determined by position in relation to the main color. For example, for a square tetrad the three additional colors must form a square with the main color on the color circle. The shard number of the main color plus three gives the position of the second shard, the main number plus six gives the third shard, and the main number plus nine gives the fourth shard. These three shard numbers are each sent to the *ColorSup* abstraction to determine their corresponding HSV values.

There are seven chord subpatchers, one each to handle the seven different chord types. They each work similarly to the *maincolor* subpatch. For dyads, the second color's HSV values are sent into the *dyad* subpatch, randomized and converted into RGB values six separate times, then returned to the main patch. For triads, the two subsequent colors are sent into the *equitriangle* or *isoctriangle* subpatches, randomized and converted to RGB four separate times, then returned. The same applies to the *rect, squarerect,* and *hexad* patches, with three modified colors returned for the tetrad methods and two modified colors returned for the hexagonal method.

The color chord determines how many variations of each chord color are used in the palette. To keep the main color the most prominent in the visualization, the palette always contains more variations of the main color than variations of the additional chord colors. For a monochromatic color scheme, all sixteen colors are variations of the main color. For a dyad there are only two colors, so the palette contains ten variations of the main color and six of the second color. For triads, there are eight variations of the main color and four each of the other two. For tetrads, there are nine variations of the main color and three each of the other three colors. A hexad produces six variations of the main color, and two each of the other five colors. The Color patch returns the 16 RGB colors and the Quadrant number back to the Analysis patch.

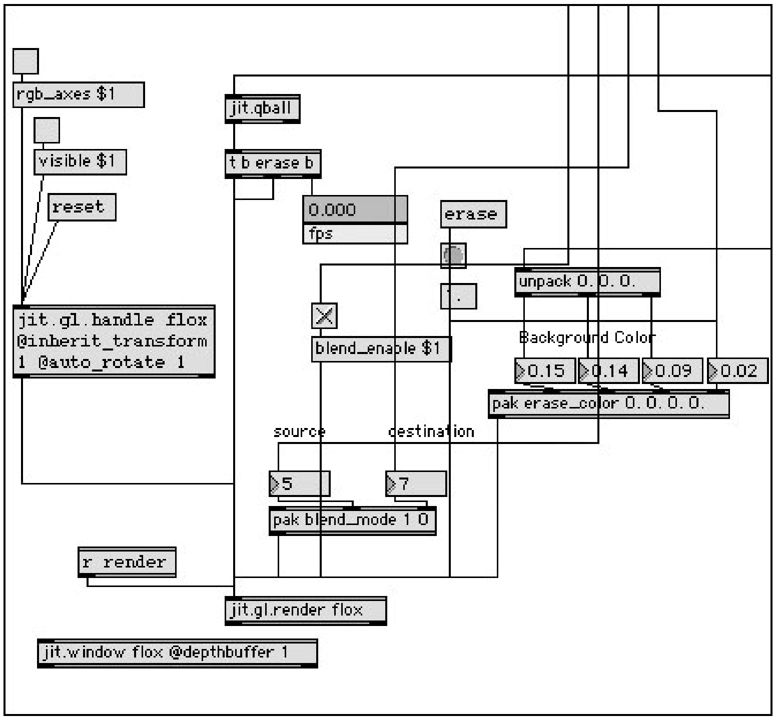
### *Animation Patches*

Within the Analyzer patch are four abstractions that control four different animation schemes. The schemes each create their own windows and have different methods of visualizing musical data. The first scheme, labeled Flocks in the Control patch, animates shape primitives using a simple flocking simulation. The second, called Particles, animates shape points using a simple particle simulation scheme. The Cloudy scheme animates Nurbs shapes according to simple spring mesh physics. The Curve scheme animates Bezier curves. Only one scheme is active at a time.

## Jitter Image Creation

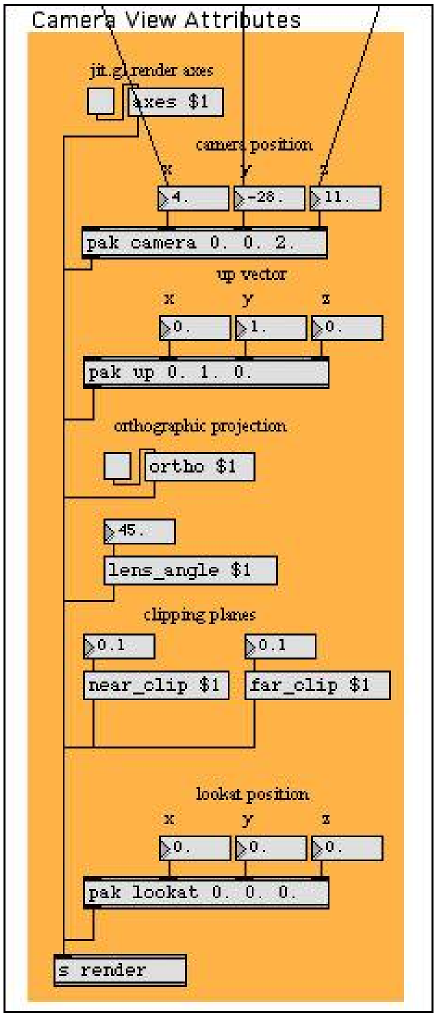
Jitter is capable of using the OpenGL cross-platform standard to draw two-dimensional and three-dimensional shapes (*Jitter Tutorial*, 2007). Jitter contains a number of OpenGL-based drawing objects for image creation. These include shape primitives such as circle, plane, sphere, cube, open cube, cylinder, open cylinder, and torus. For Jitter to draw OpenGL objects, a drawing context must be created. A drawing context consists of a named *jit.gl.render* object and a destination window with the same name. Figure 36 shows the drawing context for the Flocking animation scheme. The drawing context for this scheme has the name Flox. This creates a window, also named Flox, where the visualization will be drawn. Every shape that is to be visible in the Flox window must be drawn to the render object Flox. All of the Jitter OpenGL objects also share the name Flox.

Figure 36. The Jitter drawing context for the Flocking scheme.



The *jit.gl.render* object is also used to control camera attributes within the scene. Attributes such as camera position, camera look-at position, lens angle, and near and far clipping planes (Fig. 37) can all be sent to the render object to control the motion of the camera and look of the scene.

Figure 37. Camera view attributes for the drawing context. [*Jitter Tutorial*, 2007].



The background color and background erase color of the window are also associated with the camera. The background color and erase color are the same. However, different alpha values of the erase color can produce interesting ghosting effects in the window. The alpha value ranges from zero to one. If the alpha value is high, objects in the scene can appear to have slight motion trails. The lower the value, the more previous frames of the animation persist. If the value is very close to zero, previous views do not erase for a long while and the scene appears to have many overlapped objects in it.

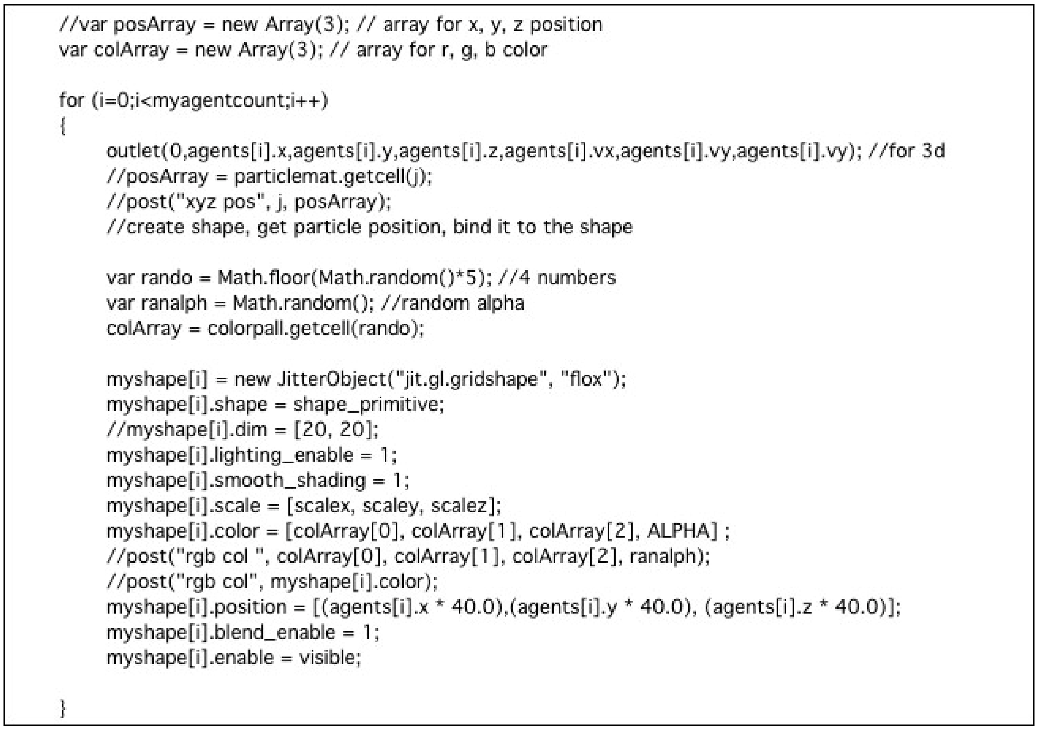
Each of the animation scheme abstractions has a drawing context and window associated with it. Four image windows are always visible but only one will actually be active. The others remain blank or frozen. For example, when the Flocks option is chosen, the musical, control, and color information is sent to the Flocking abstraction and the Flocking window Flox displays animation. The windows associated with the other schemes do not respond.

## Flocking Scheme

The animation for this scheme is based on a flocking program by Craig Reynolds that simulates the coordinated movement of birds or animals in a group. The members of the group are called agents, or ‘boids’. The agents each have a distinct position and velocity. A set of predefined rules act upon the velocity (Reynolds, 2001). Parameters such as separation, alignment, and cohesion affect the rules and change the flocking behavior. Varying the values of the parameters produces a variety of interesting movements and configurations. Reynolds’s site, http://www.red3d.com/cwr/boids, explains the procedure and history of his boids simulation in more detail.

The scheme is written in Javascript, and was included with Jitter as an example. The *js* object in Max allows procedural code written in Javascript to be used in Max. *Js* implements the core Javascript language, and contains a number of objects and functions specific to Max and Jitter (*Jitter Tutorial*, 2007). The code snippet in Figure 38 shows *jit.gl.gridshape* objects being initialized and defined in Javascript. This object creates simple geometric shapes in Jitter. Each *jit.gl.gridshape* object generated is hard-coded to be drawn in to the ‘flox’ drawing context.

Figure 38. The Flocking Javascript generating *jit.gl.gridshape* objects.



```
//var posArray = new Array(3); // array for x, y, z position
var colArray = new Array(3); // array for r, g, b color

for (i=0;i<myagentcount;i++)
{
    outlet(0,agents[i].x,agents[i].y,agents[i].z,agents[i].vx,agents[i].vy,agents[i].vy); //for 3d
    //posArray = particlemat.getcell(j);
    //post("xyz pos", j, posArray);
    //create shape, get particle position, bind it to the shape

    var rando = Math.floor(Math.random()*5); //4 numbers
    var ranalph = Math.random(); //random alpha
    colArray = colorpall.getcell(rando);

    myshape[i] = new JitterObject("jit.gl.gridshape", "flox");
    myshape[i].shape = shape_primitive;
    //myshape[i].dim = [20, 20];
    myshape[i].lighting_enable = 1;
    myshape[i].smooth_shading = 1;
    myshape[i].scale = [scalex, scaley, scalez];
    myshape[i].color = [colArray[0], colArray[1], colArray[2], ALPHA] ;
    //post("rgb col ", colArray[0], colArray[1], colArray[2], ranalph);
    //post("rgb col", myshape[i].color);
    myshape[i].position = [(agents[i].x * 40.0),(agents[i].y * 40.0), (agents[i].z * 40.0)];
    myshape[i].blend_enable = 1;
    myshape[i].enable = visible;

}
```

Within the code, other functions handle information being fed into the *js* object within the Flocking abstraction patch, including characteristics such as shape type, scale, and color. Position is calculated within the code. The original code returned two-dimensional position data that was used to draw two-dimensional circles and line segments using MSP’s simple drawing primitives. I modified the code to calculate three-dimensional position data and to draw primitive forms into Jitter's three-dimensional space. The Flox drawing context then renders the shapes in the Flox window.

The Flocking abstraction moves the camera to a random x, y, z position every time a bass note is detected. The camera always looks to the center of the scene, where the flocks are located. This results in an effect akin to camera cuts on each beat, creating an obvious correlation between image and sound. The background color and background erase color of the window are received from the Color patch. Every time new colors are placed in the color palette, the render object receives one of the variations of the main color to serve as the background color of the animation. Flocking also receives the volumes of the five frequency ranges from the Analyzer patch. Each of these constantly changing values determines the scale of the objects in the five different flocks. Each of the flocks also receives four colors from the color palette. Each time an object is generated it is randomly assigned one of these colors. Each of the five flocks is also composed of one of the eight different OpenGL primitive shapes: sphere, torus, cylinder, open cylinder, cube, open cube, flat plane, and flat circle. If picture files are received as textures from the Control patch, then the pictures are applied to the faces of the shapes. The pictures are tinted the color of assigned to the shape by the Flocking Javascript. The resulting imagery is colorful with or without textures and populated by a number of different shapes.

## Particle Scheme

The Particle patch uses a drawing context called Parts and creates a window of the same name. The camera motion of the Particle patch is similar to that of the Flocking patch, moving in a random direction each time a bass beat is detected. The method of retrieving a background and erase color is the same as well.

The animation of the Particle scheme is also based on a preexisting Javascript that was included with Jitter as an example file. This script performs three-dimensional particle generation with simple gravity. The number of particles, the speed at which the particles move, the number of particle generation points, and the decay of the particles over time are all parameters that can be adjusted in the Particle patch.

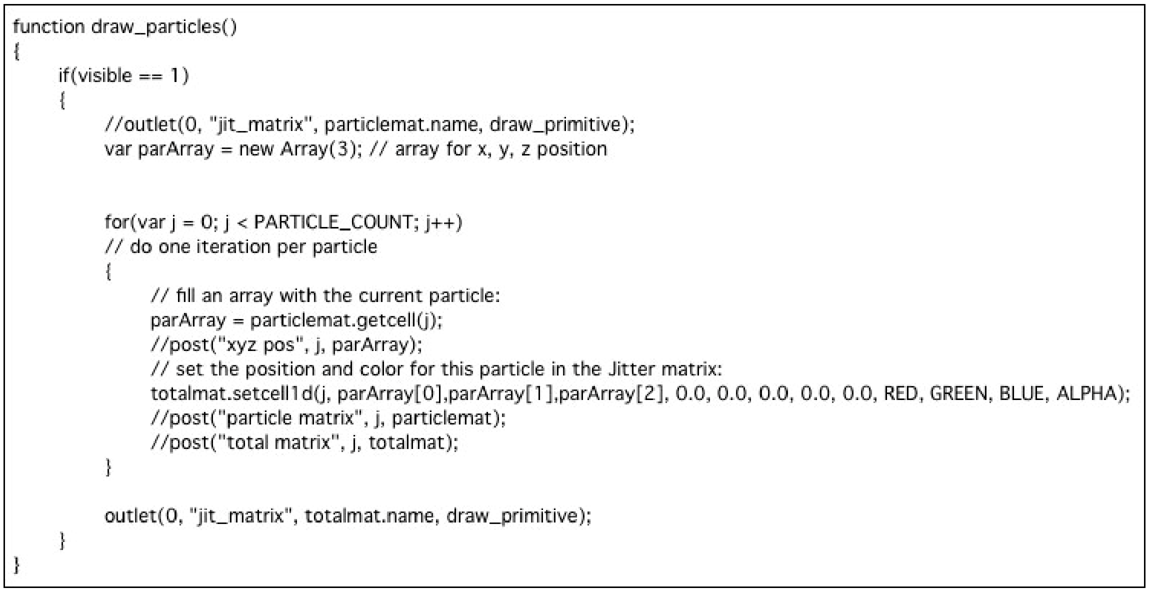
In the Javascript, the calculated particle positions are fed into a jitter matrix object, then sent out to the *jit.gl.render* object to be drawn to the window. When the *jit.gl.render* object receives a matrix, it draws the values of the first three planes of matrix data as the x, y, z positions of vertices of a single OpenGL object. This object can be drawn with any of the OpenGL drawing primitive types. Jitter has ten different drawing primitives to choose from, which are vertex points, lines, line strips, line loops, triangles, triangle strips, triangle fans, quad strips, or polygons. The original code only allowed Jitter to draw the shape with drawing primitives. I modified the code so that the different shape primitives, such as cubes, spheres, and toruses, could be drawn at the position of each vertex. The code snippets in Figure 39 show the function that draws the data with drawing primitives and the function that renders the data as shape primitives.

Rendering shape primitives slows the visualization, but results in more textured imagery.

In addition, I made it possible for the particle group to be rendered using an input color.

Figure 39. The Particle scheme Javascript. (a) Code to generate the particles with drawing primitives. (b) Code to generate the particles with shape primitives.

(a)



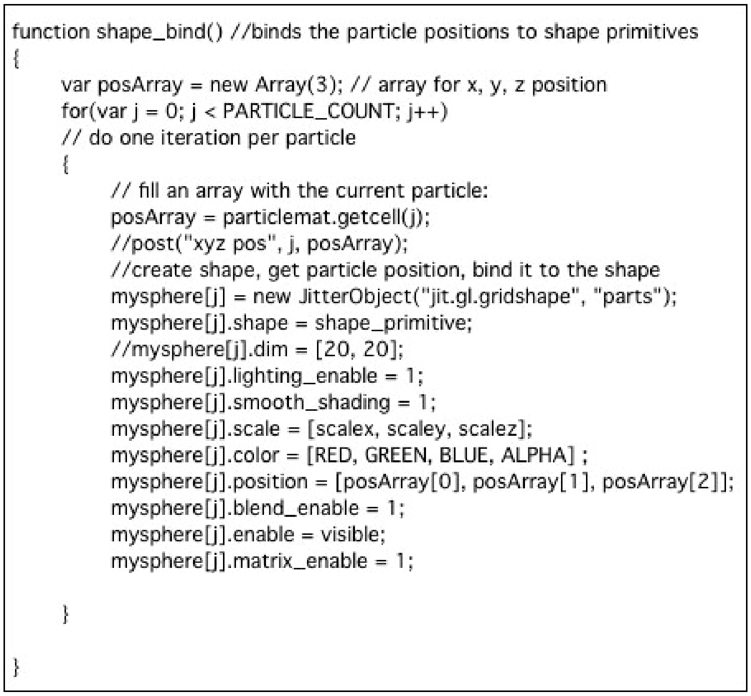
```
function draw_particles()
{
    if(visible == 1)
    {
        //outlet(0, "jit_matrix", particlemat.name, draw_primitive);
        var parArray = new Array(3); // array for x, y, z position


        for(var j = 0; j < PARTICLE_COUNT; j++)
        // do one iteration per particle
        {
            // fill an array with the current particle:
            parArray = particlemat.getcell(j);
            //post("xyz pos", j, parArray);
            // set the position and color for this particle in the Jitter matrix:
            totalmat.setcell1d(j, parArray[0],parArray[1],parArray[2], 0.0, 0.0, 0.0, 0.0, 0.0, RED, GREEN, BLUE, ALPHA);
            //post("particle matrix", j, particlemat);
            //post("total matrix", j, totalmat);
        }

        outlet(0, "jit_matrix", totalmat.name, draw_primitive);
    }
}
```

Figure 39 continued.

(b)



```
function shape_bind() //binds the particle positions to shape primitives
{
      var posArray = new Array(3); // array for x, y, z position
      for(var j = 0; j < PARTICLE_COUNT; j++)
      // do one iteration per particle
      {
            // fill an array with the current particle:
            posArray = particlemat.getcell(j);
            //post("xyz pos", j, posArray);
            //create shape, get particle position, bind it to the shape
            mysphere[j] = new JitterObject("jit.gl.gridshape", "parts");
            mysphere[j].shape = shape_primitive;
            //mysphere[j].dim = [20, 20];
            mysphere[j].lighting_enable = 1;
            mysphere[j].smooth_shading = 1;
            mysphere[j].scale = [scalex, scaley, scalez];
            mysphere[j].color = [RED, GREEN, BLUE, ALPHA] ;
            mysphere[j].position = [posArray[0], posArray[1], posArray[2]];
            mysphere[j].blend_enable = 1;
            mysphere[j].enable = visible;
            mysphere[j].matrix_enable = 1;

      }

}
```

The scheme contains five different particle groups, each distinct and unaffected by the others. The Particle patch receives the volumes of the five frequency ranges from the Analyzer patch. These values determine the size of the particles or shapes of the objects in the five different particle groups. The abstraction also receives the color chord type from the Analyzer patch. This defines how many attractors each particle group will have. The overall volume of the audio signal affects the alpha values of all the particles. The colors of the five groups are received in from the Color patch.

## Cloudy Scheme

The Cloudy patch creates a window called Polo from its drawing context. The camera motion of the Cloudy patch is the same as the Flocking and Particle patches, moving in a random direction each time a bass beat is detected. The method of retrieving a background and erase color is the same as well.

The animation of the Cloudy scheme is also based on a preexisting Javascript that was included with Jitter as an example file. This script performs a simple spring simulation that is physically motivated but inaccurate. The code creates a Nurbs surface from a grid of points in three-dimensions. The spring-connected grid points stretch, contract, and bounce away from one another, causing the Nurbs form to billow, wave, and fold.

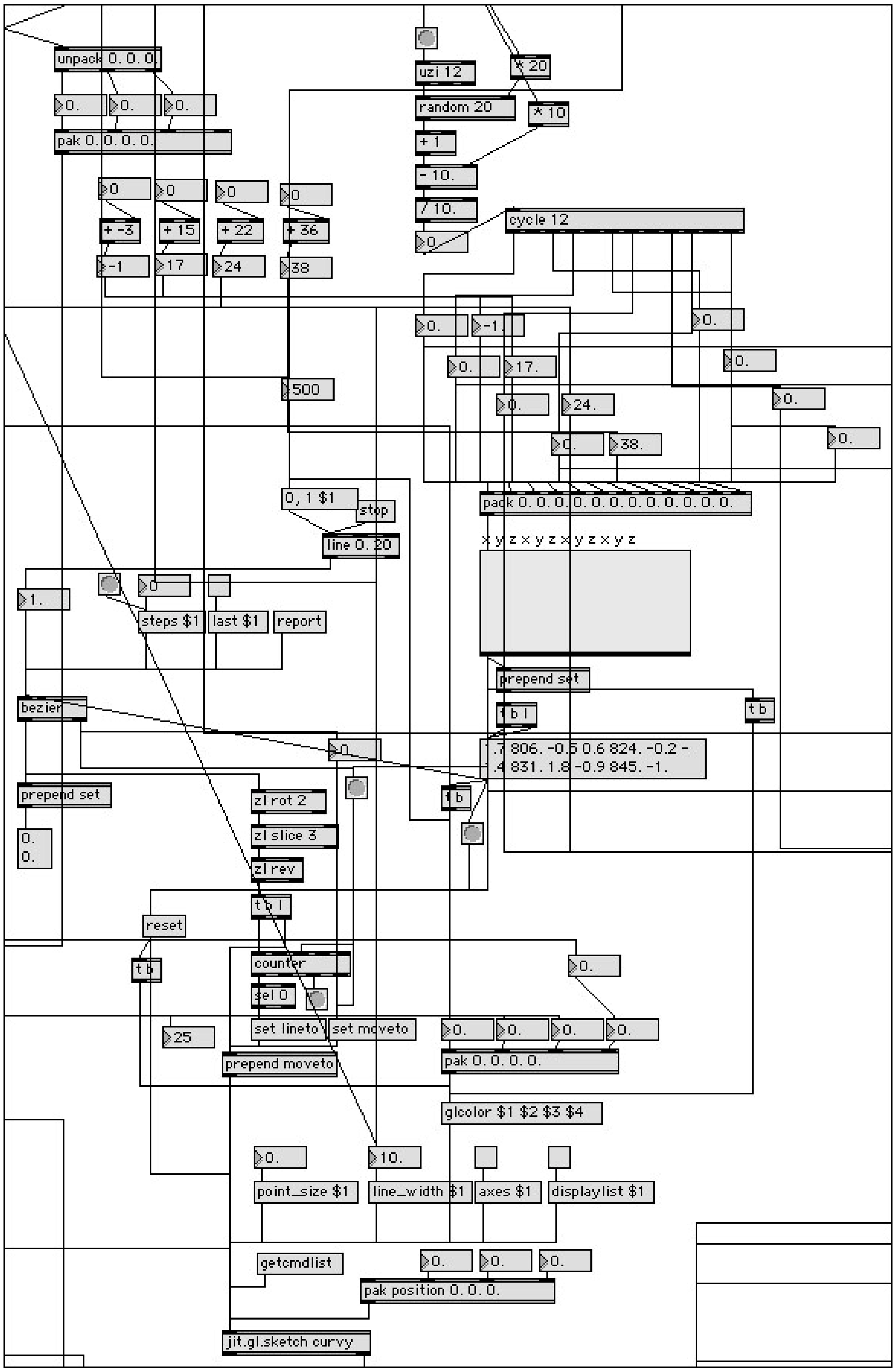
Parameters such as spring strength, friction, and inertia affect the behavior of the spring grid. These are fed into the Javascript within the Cloudy patch.Varying the values of the parameters can produce a variety of interesting Nurbs surfaces and movements, but even the default spring simulation generates cloud-like motion. The Cloud abstraction patch draws five separate springy Nurbs surfaces. It receives the different volumes of the five frequency ranges from the Analyzer patch. These volume values determine the alpha of the objects in five different flocks. The colors of the five Nurbs forms are received from the Color patch. As in the Flocking scheme, if picture files are received as textures from the Control patch then they are applied to the Nurbs surfaces. The textures are tinted the color of assigned to the shape by the Javascript.

## Curve Scheme

The Curve animation scheme uses a different camera animation method than the other three schemes. Here, the camera continues forward in the z direction for the duration of the song, so that the view is always progressing forward through the scene. Each time a bass beat is detected, the camera moves a small random value in the x and y directions, but continues to move forward. This gives the camera a slightly meandering motion. The camera always looks in the z direction. Similarly to the other animation scheme abstractions, the background and erase colors are received from the Color patch.

The curve objects in this scheme are not controlled by a Javascript, but are generated and controlled directly within the Curve abstraction patch. For every beat detected, a pipeline of Max objects creates three curved lines. Figure 40 shows the connected objects that create one curve. At the beat, twelve values are packed into a list to form four (x,y,z) triplets. Each of the triplets defines a vertex in three-dimensional space. The four vertices are the control points of a Bezier curve. The x and y values for each vertex are random, but the z value is dependent on the camera's position. The first z value is slightly behind the z value of the camera, the second is slightly ahead, the third a bit further, and the last far out in front of the camera.

Figure 40. The pipeline of objects that creates Bezier curves.



The list of vertices is sent into an object called *bezier* that generates a series of small line segments that make up the curve. These segments are sent to a *jit.gl.sketch* object. *Jit.gl.sketch* performs simple OpenGL commands that draw the line segments. Each line segment is drawn consecutively so that the Bezier curve looks as though it draws itself into the scene. The z positions of the Bezier points ensure that the lines appear to come from behind the camera into the space in front of it.

The *jit.gl.sketch* object receives the colors for each curve from the Color patch. The Curve abstraction also receives the five volumes from the Analyzer patch. The Bass, High Mid, and Brilliance volumes control the alpha values of the three curves.

# CHAPTER VI

# RESULTS

## Song Classification

A bank of training song files were used during initial development and testing of the mood classification method. After the method produced accurate mood extraction results for this small set of training files, I compiled a new set for testing. Testing against a new set ensured that the results were not biased towards the methodology. The testing set contained twenty-six songs that can all be broadly categorized as modern popular music. They range in date from the 1950's to the present day, performed in English or Japanese. The genres of the various songs include R&B, Rap, Pop, New Age, and Alternative Rock.

I categorized each song in the testing set into Contentment, Depression, Anxious/Frantic, or Exuberance groups according to my predictions of the song's mood classification. Table 1 shows the song list, the predicted mood classification, and the actual mood classification produced in the program.

Table 1
Predicted and actual mood cluster classification of selected songs.

| **Song Title** | **Artist** | **Predicted Mood Cluster** | **Calculated Mood Cluster** | **Accurate** |
|---|---|---|---|---|
| A Wolf at the Door | Radiohead | Exuberance | Exuberance | Yes |
| Pink Maggot | Deftones | Exuberance | Contentment--Exuberance | Yes |
| Transylvanian Concubine | Rasputina | Exuberance | Exuberance | Yes |
| I Put a Spell on You | Screamin' Jaw Hawkins | Exuberance | Contentment--Depression | No |
| Black is the Color of My True Love's Hair | Nina Simone | Exuberance | Exuberance | Yes |
| Will It Go 'Round In Circles | Billy Preston | Exuberance | Anxious/Frantic--Exuberance | Yes |
| Paper Planes | M.I.A | Exuberance | Exuberance | Yes |
| Rose | A Perfect Circle | Anxious/Frantic | Depression--Anxious/Frantic | Yes |
| Girl Anachronism | Dresden Dolls | Anxious/Frantic | Depression--Anxious/Frantic | Yes |
| Cult of Personality | Living Colour | Anxious/Frantic | Anxious/Frantic | Yes |
| Knife Party | Deftones | Anxious/Frantic | Exuberance--Anxious/Frantic | Yes |
| Merciless Cult | Dir en grey | Anxious/Frantic | Anxious/Frantic | Yes |
| Bombs Over Bagdad | Outkast/Rage Against the Machine | Anxious/Frantic | Anxious/Frantic | Yes |
| Girl Boy Song | Aphex Twin | Anxious/Frantic | Anxious/Frantic--Exuberance | Yes |
| Garden of Everything | Sakamoto Maaya & Steve Conte | Contentment | Contentment | Yes |
| Orange Moon | Erykah Badu | Contentment | Depression--Contentment | Yes |
| I Got a Woman | Ray Charles | Contentment | Contentment | Yes |
| I Can't Go For That | Daryl Hall & John Oates | Contentment | Contentment | Yes |
| Everyday People | Arrested Development | Contentment | Contentment | Yes |
| Time of Gold Flowing | Please Save My Earth OST | Contentment | Contentment | Yes |
| Orestes | A Perfect Circle | Depression | Depression--Anxious/Frantic | Yes |
| My Sweet Prince | Placebo | Depression | Contentment | No |
| Under the Milky Way | The Church | Depression | Depression | Yes |
| Last Night I Dreamt That Somebody Loved Me | The Smiths | Depression | Contentment--Depression | Yes |
| Eleanor Rigby | The Beatles | Depression | Contentment | No |
| Miss Misery | Elliot Smith | Depression | Contentment--Exuberance | No |

The ‘Calculated Classification’ column sometimes shows multiple classifications. In many of the songs, the tempo, volume, and/or overall tone of the music changes within the song. Perceptually, this often changed the mood of the song. The changes affected the program’s calculations, resulting in a song moving into a new mood quadrant. For example, the song Pink Maggot by the Deftones has an extended slow introduction section that lasts almost three minutes. It suddenly becomes much louder and more energetic when the main section of the song starts. The mood classification program accurately changes from Contentment to Exuberance, reflecting the transition. The changes were only considered accurate if they corresponded to a perceived change in the music that matched the new classification. Such changes ultimately result in a more thorough sense of the affect of the song.

Out of the twenty-six songs, four songs, or 15%, were classified incorrectly. The method therefore exhibits an 85% success rate. Of the songs that were misjudged, three predicted Depression songs were classified as Contentment instead. I believe that the difference between the two Quadrants is often a matter of whether the song is in a major or minor key. This will be explored further in the Future Work section.

## Colors

A stated goal of the visualizer was to ensure that the colors of the visualization matched the musical mood. While this is a subjective judgment, steps were taken to ensure that low Energy moods (Contentment and Depression) manifested duller colors and high Energy moods (Exuberance, Anxious/Frantic) exhibited brighter colors. Similarly, low Stress moods (Contentment, Exuberance) were made to exhibit lighter colors and high Stress moods (Depression, Anxious/Frantic) were made to have darker colors. In addition, the low Stress Contentment and Exuberance moods were designed to produce color chords with fewer colors, whereas the high-Stress Depression and

Anxious/Frantic moods were designed to produce high-number color chords. Figure 41 shows a variety of the color palettes produced for each of the mood types.

Figure 41. Color palettes for different mood and chord types. (a) Contentment monochrome. (b) Contentment dyad. (c) Exuberance equilateral triad. (d) Exuberance isosceles triad. (e) Depression isosceles triad. (f) Depression square tetrad. (g) Anxious/Frantic rectangular triad. (h) Anxious/Frantic hexagonal.

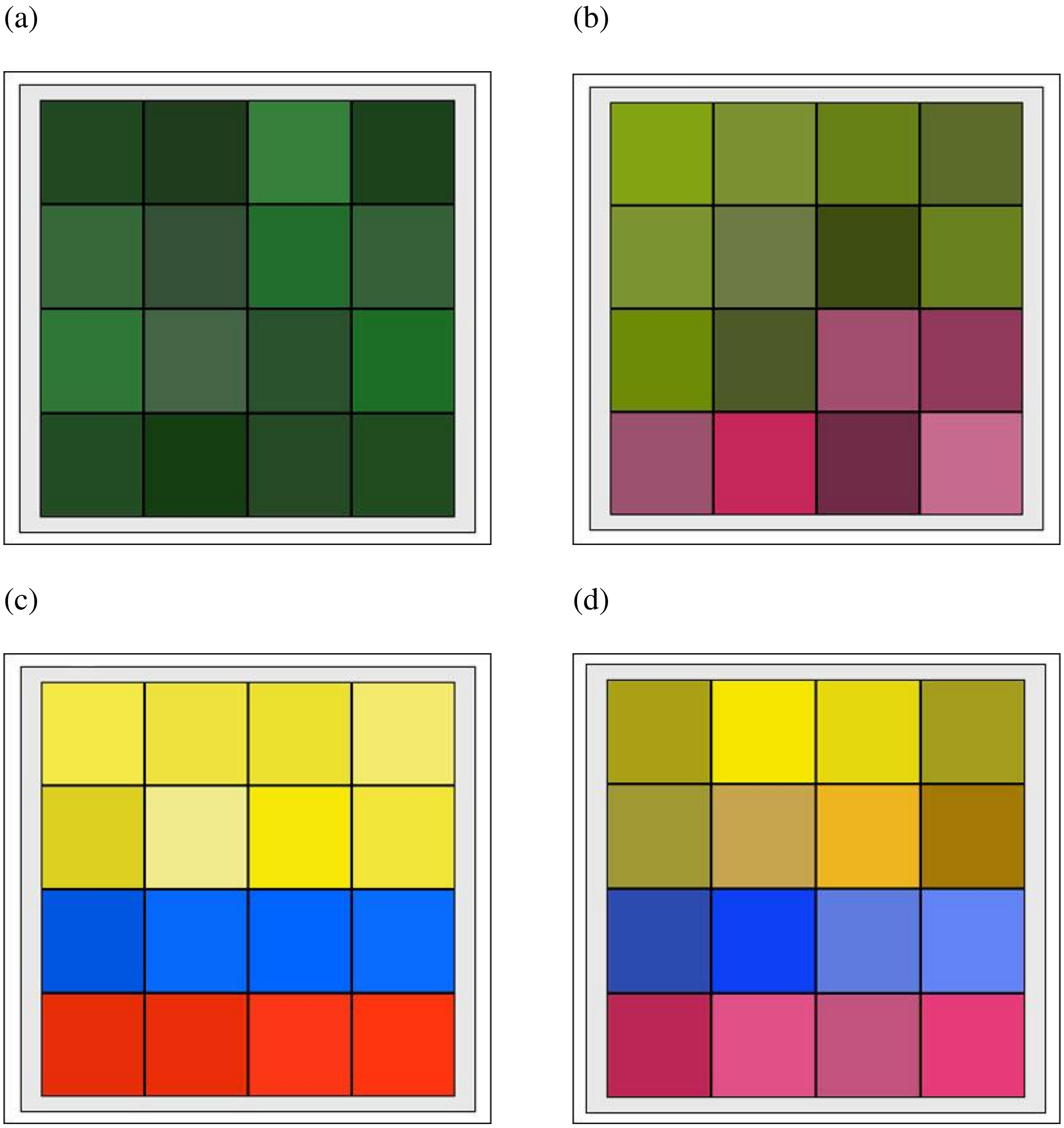
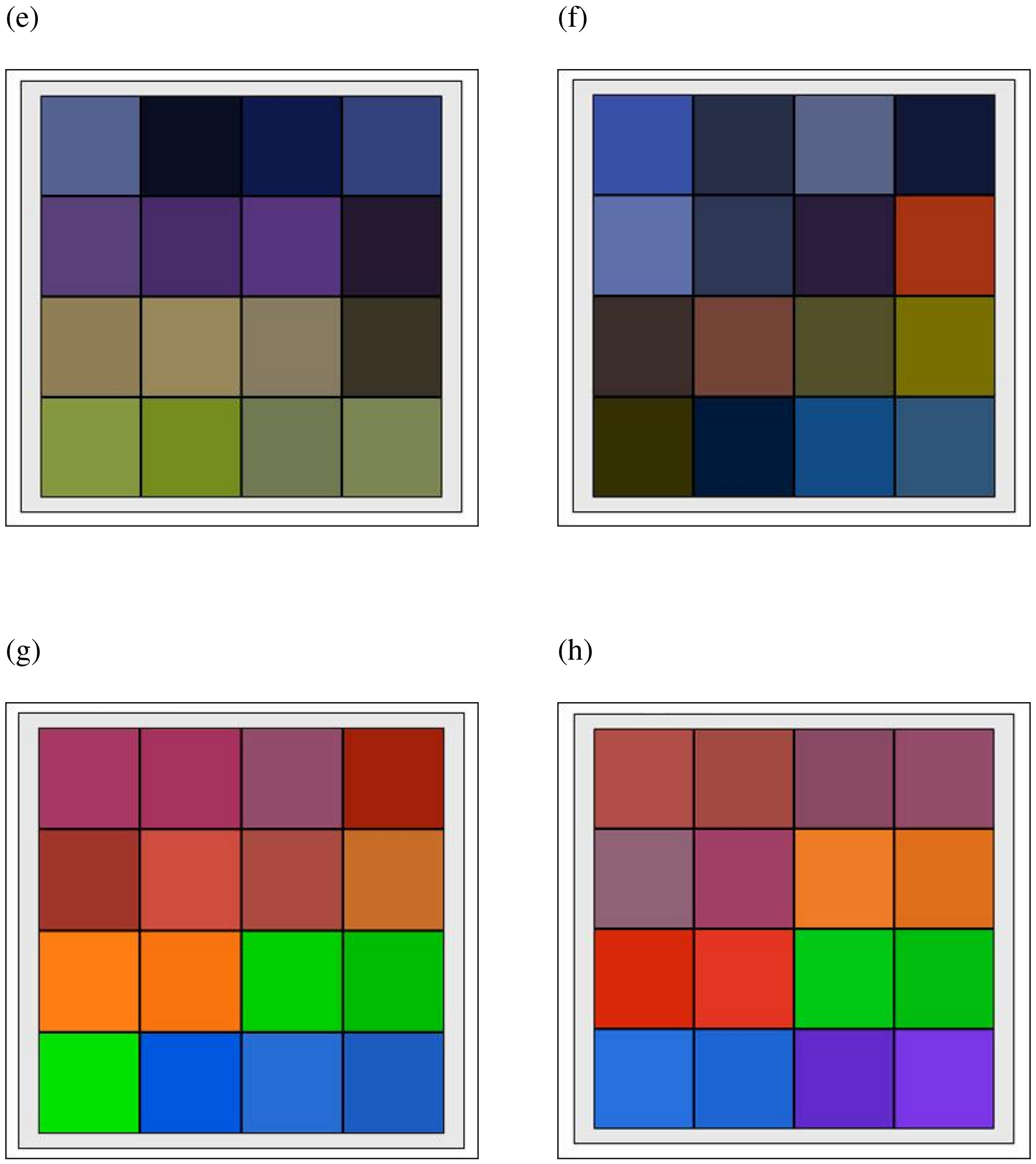


Figure 41 continued.



(e)

(f)

(g)

(h)

Contentment produced only monochrome or dyad chord types, resulting in single or two-color palettes. This matched Contentment quadrant's low Stress classification. The main color in the monochrome palette in Figure 41 (a) is the green of tranquility and contentment. The low energy and cool temperature of the green colors match the low Energy and low Stress of the Contentment quadrant. The palette is made up of colors that are generally duller than the original, and many are also lighter. This fits the assertion that colors in a Contentment palette should be both dull and light. In Figure 41 (b) the Contentment dyad is composed of variations of yellow-green and red-violet. Both are constrained to more subdued versions of the original color. Both also have lighter variations of the original. However, in this case the some dark variations of red-violet and yellow-green are present. The randomness of the values added to the original colors did sometimes result in colors going out of their perceived bounds.

Even with the random values constrained, dyads sometimes produced color combinations that were somewhat jarring or discordant. This occurred most often with Exuberant palettes. Such combinations would seem better suited to high Stress mood quadrants.

Exuberant triads generally produced bright and light palettes, matching the high Energy and low Stress mood. Bright and energetic main colors, even when paired with cooler colors such as blue, retain a sense of happiness and radiance. As seen in Figure 41 (d), even when the color variations result in some duller or darker colors, the brighter colors make the palette retain its Exuberant nature.

Depression generates palettes that are dull and dark in nature. Even with an occasional bright color, the low Energy, high Stress character of Depression is maintained. In Figure 41 (f), a single bright color stands out from the rest. However, it does not affect the general dull, depressed feeling of the palette.

Anxious/Frantic produces a high number of colors with much variation. These palettes contain bright, dark colors as they were meant to. However, in hindsight I believe the feeling of Anxious/Frantic may have been more effectively manifest as a discordant dyad, with two polar opposite hues of high saturation and dark color.

While the random color method did produce some variations in the palettes that did not fit a specific mood quadrant, in general I found the moods evoked by the palettes matched the moods of the songs that produced them.

## Visuals

Visualizing clear correspondences between musical elements and visual elements was another stated goal. In the animations, the beat information provides the most obvious link between sound and image. In each animation, a beat causes a change in camera direction or placement of objects. Third parties that viewed the visualization were able to identify and relate to the easily recognizable connection. Parameters such as transparency and scale provide other correlations between the sound and the music. Viewer response and reaction indicated that these further allowed viewers to relate to the visuals. While judgment of the results is also subjective, I believe that the correspondences are clear enough that anyone able to perceive beats or volume changes in music will recognize them.

The actual visuals illustrate the characteristics of their mood type. In general the visualizations of Exuberant songs display few colors but they are bright and opaque. Anxious/Frantic songs display many colors and many different shape and sizes. Their animations exhibit fast motions and quick camera cuts. Contentment songs are visualized with few colors, fewer objects, and slower, smoother camera motion. Depressed songs also display slower and more even camera motion. They have a higher number of colors than Contentment.

The Flocking scheme is most successful when visualizing Anxious/Frantic songs. It creates a large number of shapes that never stop moving. The colors of each of the shapes are constantly changing so that the forms seem to vibrate. This additional motion lends the shapes an agitated quality. This fits with the frenetic character of the Anxious/Frantic quadrant. The quick camera cuts also add to the energy and stress of the visualization.

Figure 42 shows a number of stills from the Flocking scheme that illustrate visualization possibilities for the different mood types. Fewer shapes are generated for the low Energy quadrants of Contentment and Expression, as shown by Figure 42 (a) and (f). Exuberance and Anxious/Frantic generate such a large number of shapes that they quickly begin to overlap. This results in the creation of interesting textural images, as shown in Figure 42 (b) through Figure 42 (e).

Figure 42. Stills from the Flocking visualization. (a) Depression. (b) Anxious/Frantic. (c) Anxious/Frantic. (d) Exuberance. (e) Exuberance. (f) Contentment.

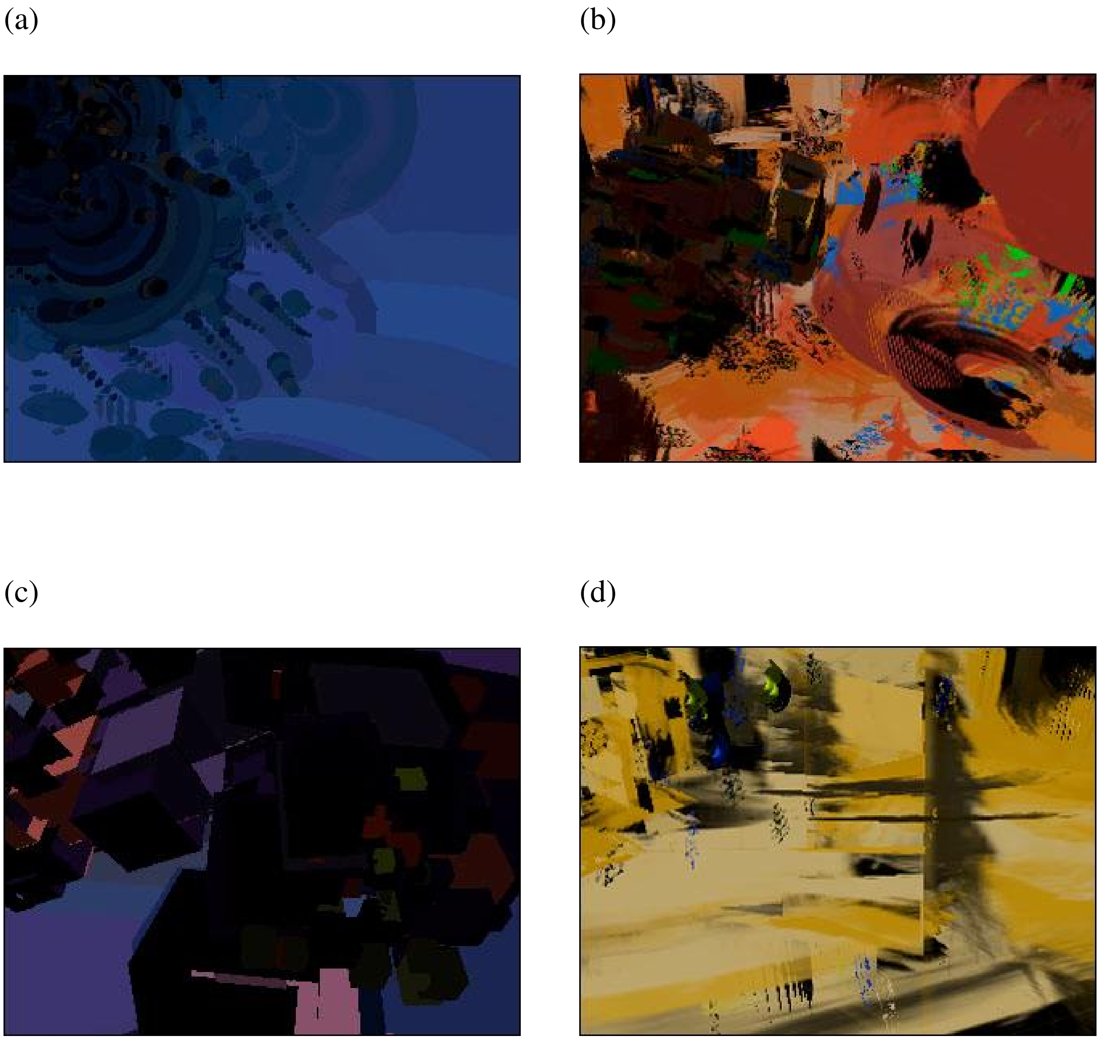
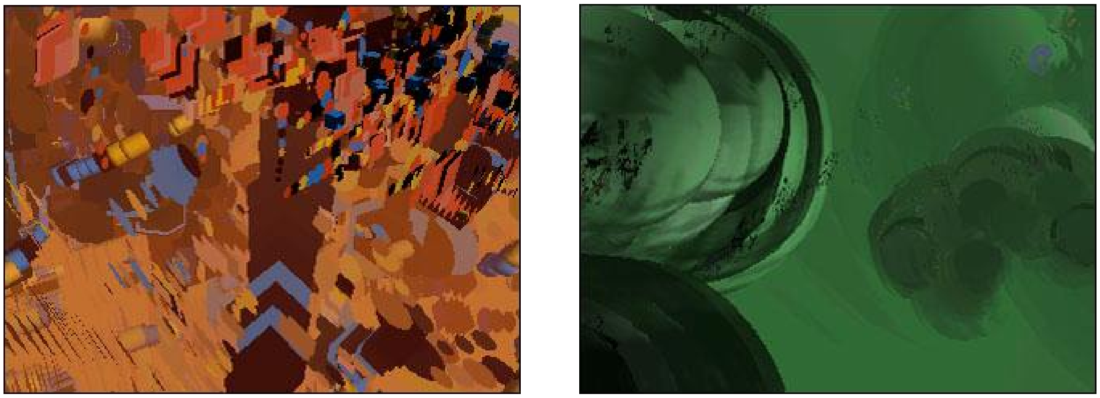


Figure 42 continued.



(e)

(f)

Figure 43 shows examples from the Particle animation scheme. The Particle scheme works well with Exuberant and Anxious/Frantic Songs. For the low Energy mood types the number of particles in the visualization is low and the particles are smaller in size. Because there are fewer particles moving at once, the particle cloud seems less dense. Figure 43(a) and (d) illustrate this. High Energy songs are crowded with particles that are constantly moving and drawing over themselves. The rapid motion of the particles gives the visualization a sense of energy. The dense central crowding of overlapping particles adds stress to the visualization, creating a sense of anxiety.

Figure 43. Stills from the Particle visualization. (a) Contentment. (b) Exuberance. (c) Exuberance. (d) Depression. (e) Anxious/Frantic. (f) Anxious/Frantic.

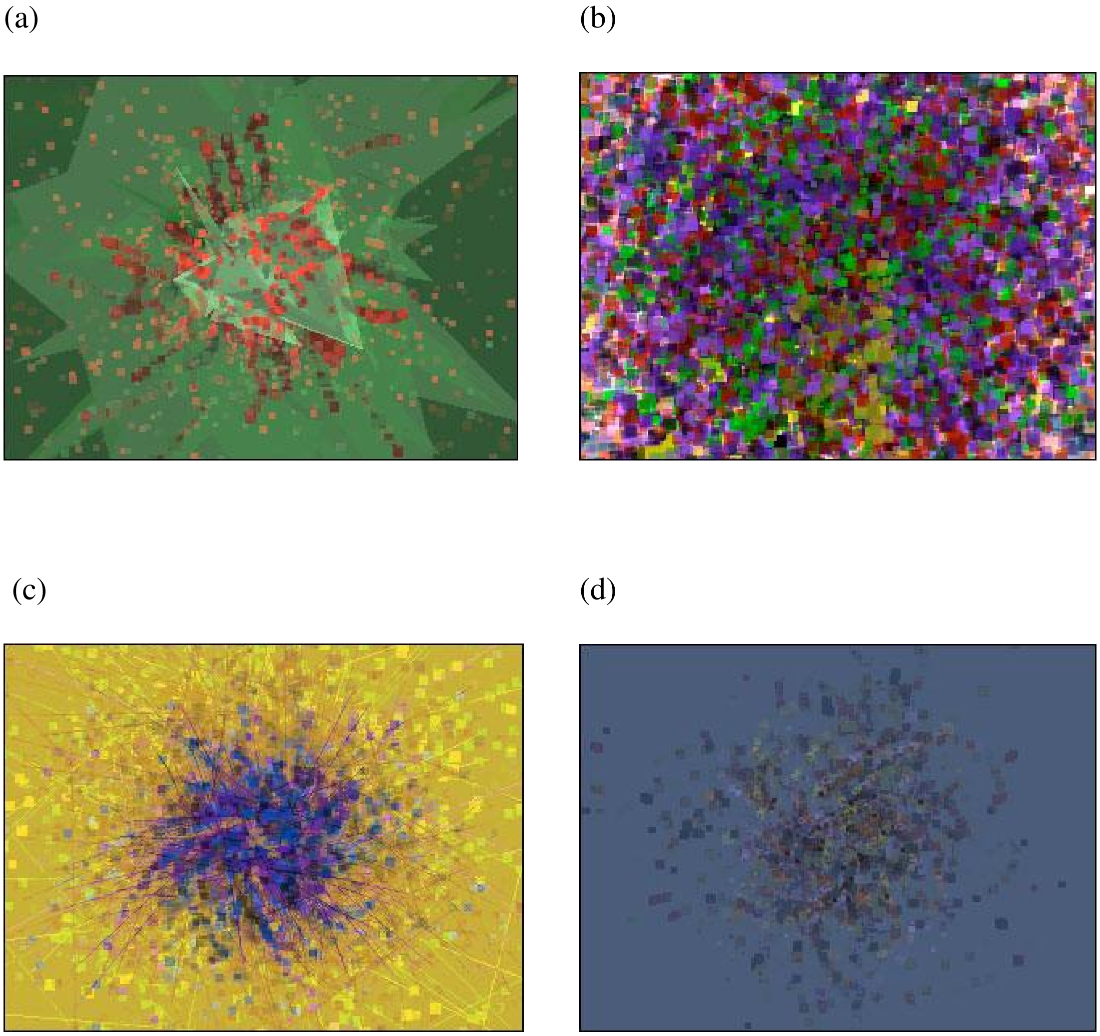
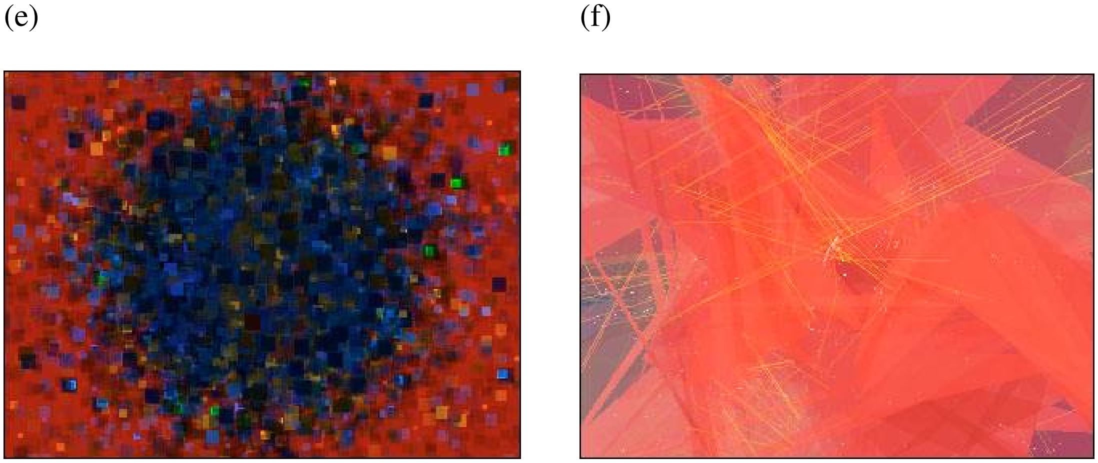
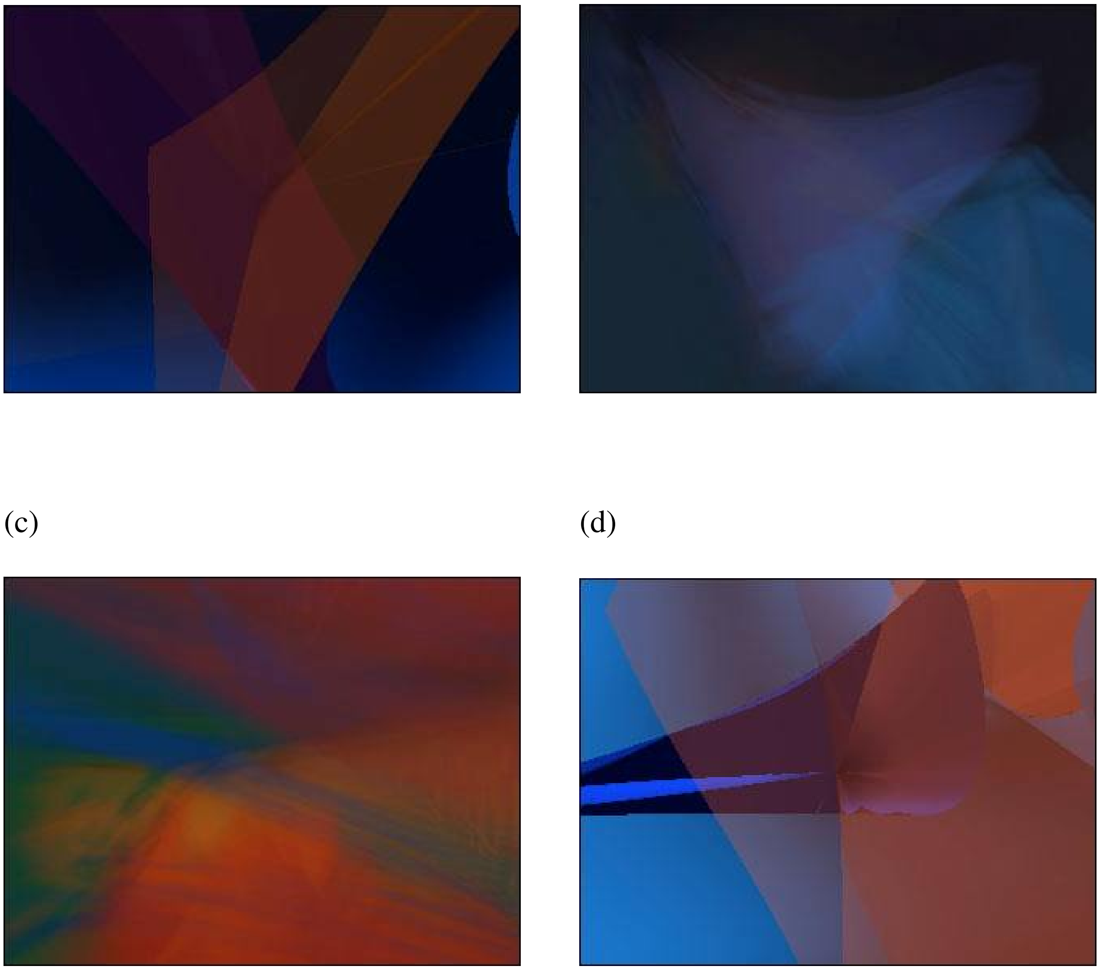


Figure 43 continued.


(e) (f)

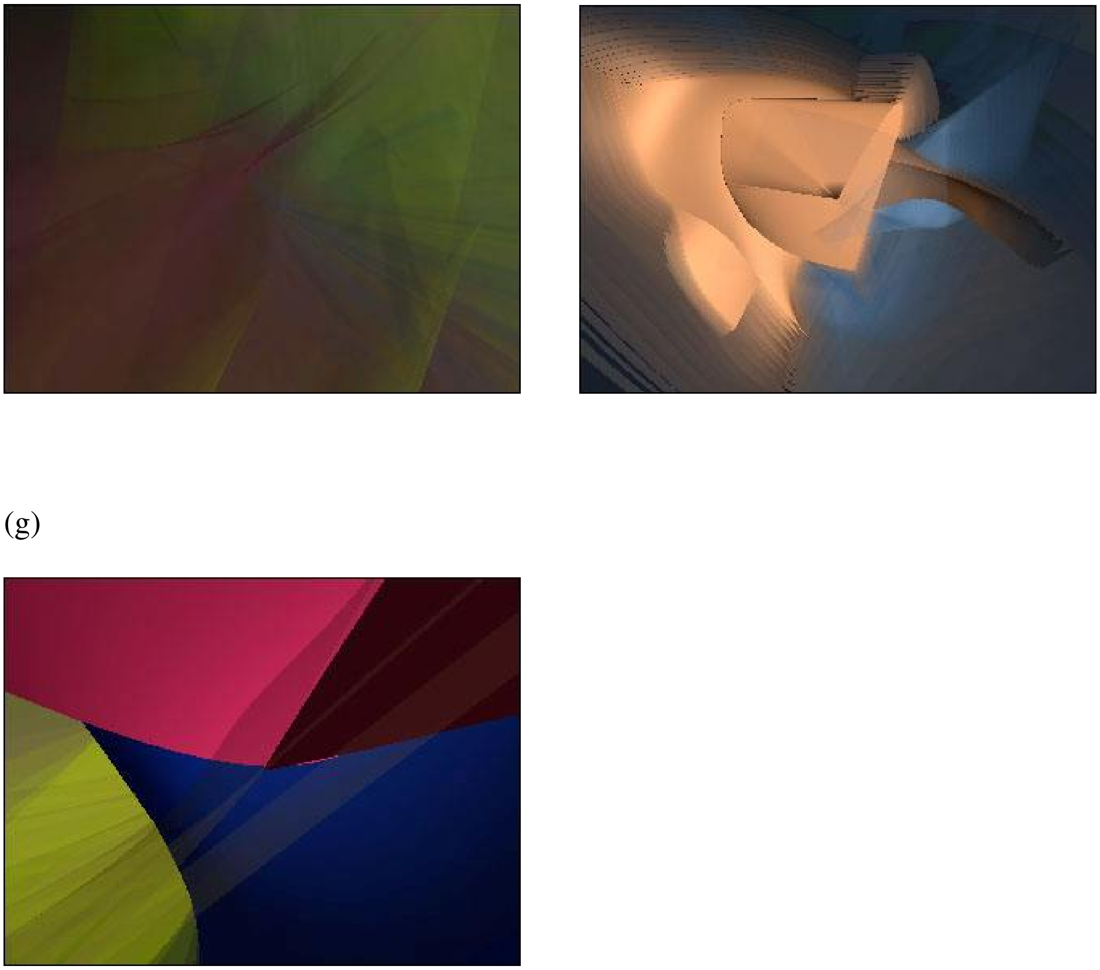
The Nurbs objects of the Cloudy animation scheme are shown in Figure 44. The Cloudy scheme is most successful at visualizing Contentment. The billowing Nurbs with their rounded shapes give a sense of and low energy and low agitation, as shown in Figure 44 (e) and (f). Depression works well too, especially when textures are applied to the Nurbs surfaces. The textures give the illusion that the colors of the different Nurbs shapes are bleeding into one another. With depressions dark colors, this bleed becomes stressful and almost insidious, as illustrated in Figure 44 (b). The multiple colors of the Anxious/Frantic quadrant produce interesting color bleeds as in Figure 44 (c), but the high energy of the quadrant causes the camera to cut around the scene a rapidly. High Energy Exuberance songs also cause fast camera motion. For Anxious/Frantic and Exuberance, the softly billowing motion of the Nurbs objects is lost to the camera cuts. However, the cuts provide quick glimpses of interesting shape configurations, as in Figure 44 (d) and (g).

Figure 44. Stills from the Cloudy visualization. (a) Depression. (b) Depression. (c) Anxious/Frantic. (d) Anxious/Frantic. (e) Contentment. (f) Contentment. (g) Exuberance.


(a) (b)

(c) (d)

Figure 44 continued.



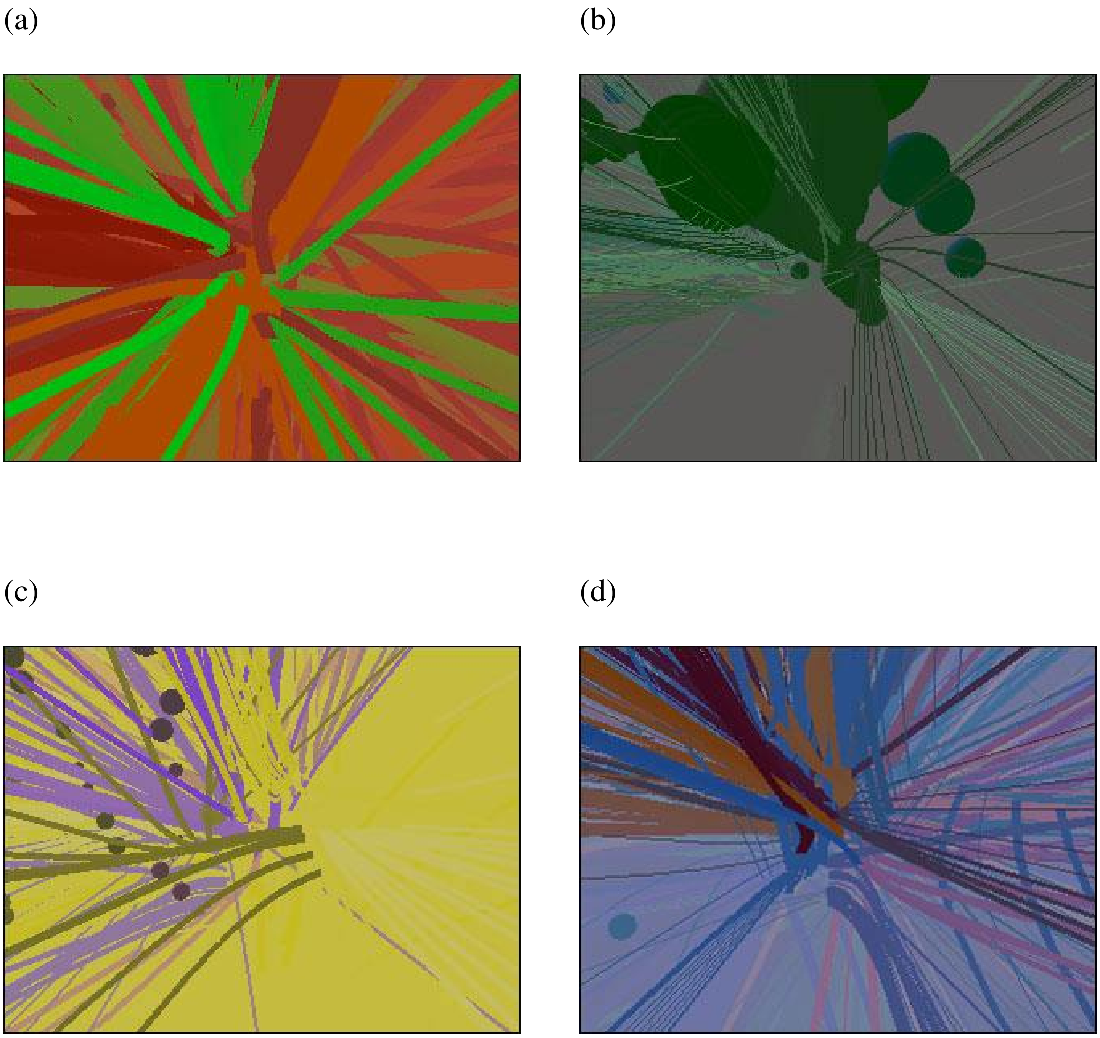
(e)

(f)

(g)

Figure 45 shows stills from the curve sketch animation scheme. In general the high Stress quadrants, which have more rhythm information, produce a higher number of curves. Figure 45(a) and (d) show a large number of curves in the scenes. The texture of the scenes tends to be denser and thicker, since the curves are drawing over other curves more often. Low Energy quadrants tend to have thinner curves that leave more of the

Figure 45. Stills from the Curve visualization. (a) Anxious/Frantic. (b) Contentment. (c) Exuberance. (d) Depression.



background visible, similar to Figure 45 (b). In these quadrants the visuals seem less energetic to match the quadrant's low Energy character.

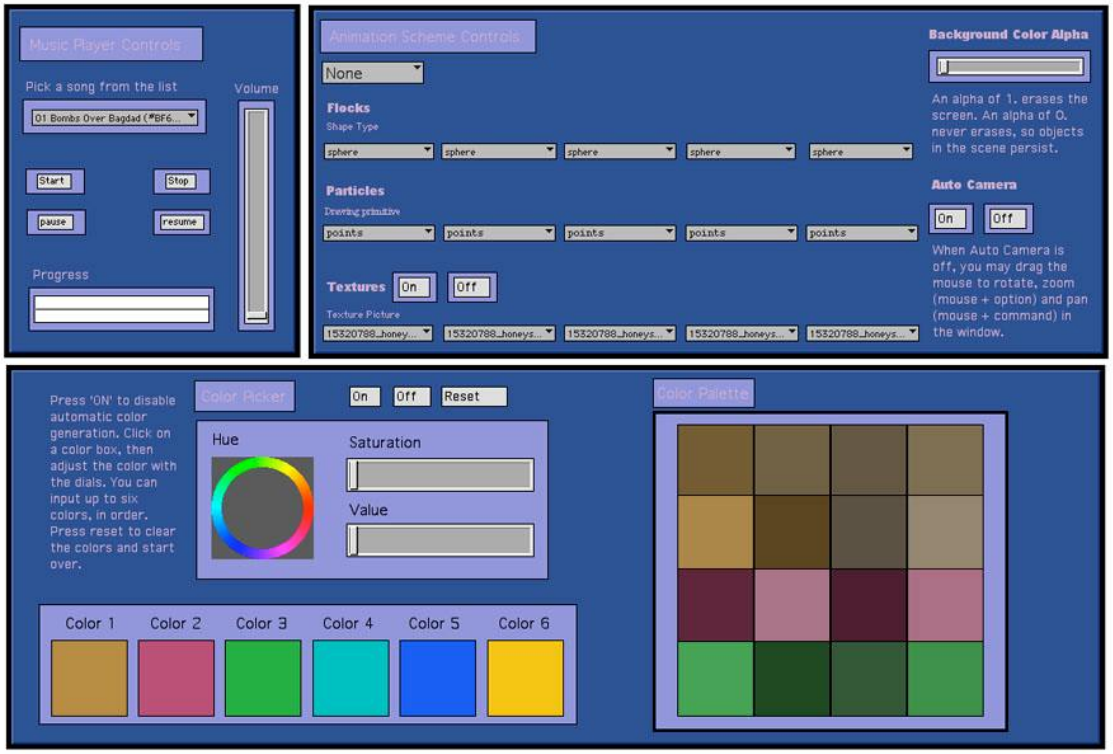
## Interactivity

Interactivity is achieved using a simple interface that allows the user to control certain parameters of the visualization. For two reasons only a few parameters are available for user modification. First, presenting the viewer with too many options would potentially confuse the user or distract him from the actual visualization. Second, including too many user controls reduced the run-time performance speed of the application. Indeed, usage of the color controls and of the texture controls slowed the application's run-time speed down so that the visualization lagged slightly. However, once the colors or textures were set, the visualization returned to normal run-time speed.

The interface was designed to be simple and easy to understand. It includes written instructions that make it very straightforward, as shown in Figure 46.

Figure 46. The Control interface. (a) The entire interface panel. (b) Music player controls. (c) Animation scheme controls. (d) Color picker.

(a)



An alpha of 1. erases the screen. An alpha of 0. never erases, so objects in the scene persist.

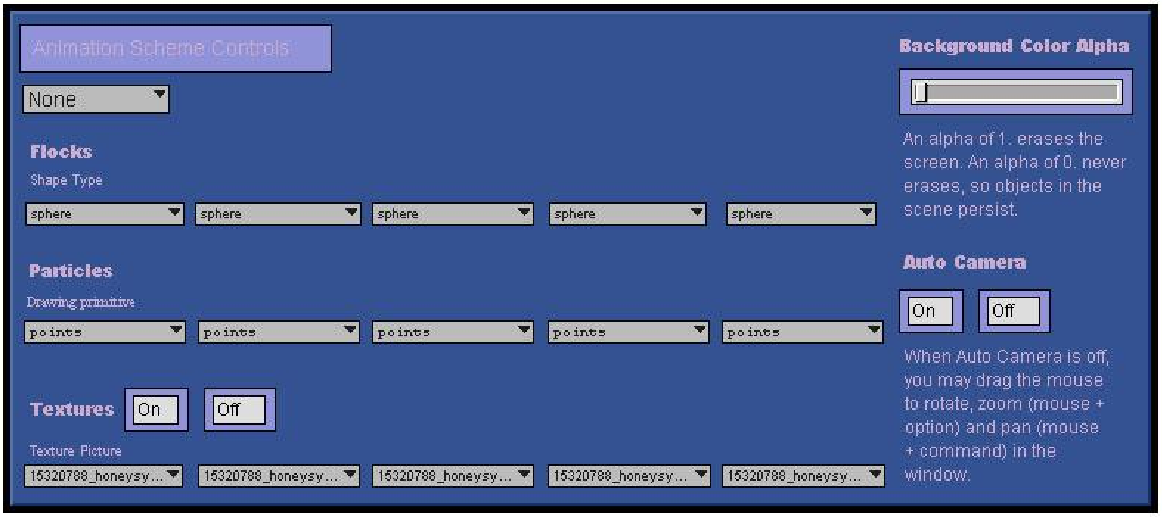
When Auto Camera is off, you may drag the mouse to rotate, zoom (mouse + option) and pan (mouse + command) in the window.

Press 'ON' to disable automatic color generation. Click on a color box, then adjust the color with the dials. You can input up to six colors, in order. Press reset to clear the colors and start over.

(b)

Figure 46 continued.

(c)



None

sphere | sphere | sphere | sphere | sphere

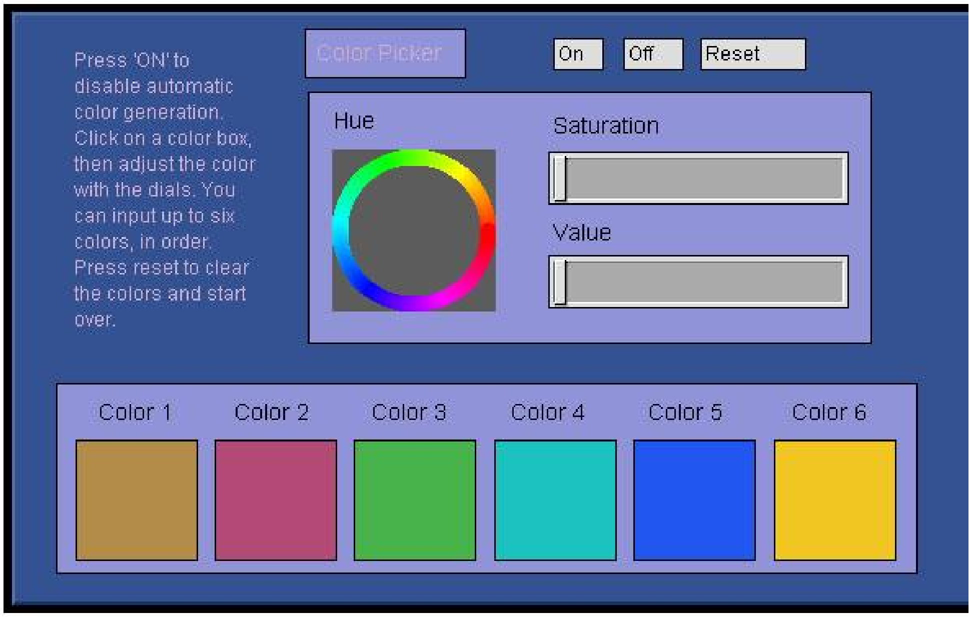
points | points | points | points | points

15320788_honeysy… | 15320788_honeysy… | 15320788_honeysy… | 15320788_honeysy… | 15320788_honeysy…

An alpha of 1. erases the screen. An alpha of 0. never erases, so objects in the scene persist.

When Auto Camera is off, you may drag the mouse to rotate, zoom (mouse + option) and pan (mouse + command) in the window.

(d)



Press 'ON' to disable automatic color generation. Click on a color box, then adjust the color with the dials. You can input up to six colors, in order. Press reset to clear the colors and start over.

The Music Player controls allow the user to choose which song to visualize. A user can read in any AIFF song file to be visualized. The start, stop, pause, and resume controls affect the playback of the song files. The volume bar controls playback volume, and the progress bar shows how much time has elapsed in the song.

The Animation Scheme controls allow the user to pick which of the four animation schemes is visualized and to affect certain parameters of the schemes. The user can control what shape types appear in the Flocking animation and what drawing primitive types appear in the Particle animation. The background color alpha slider lets the user change the transparency of the background color. The lower the value, the less the previous frames of the animation are erased. A low alpha causes old images to persist for a while, giving the shapes in the visualization a ghosting effect. The automatic camera motion in each animation scheme can also be disabled. This allows the viewer to manually control the camera with the mouse. Dragging the mouse in the visualization window rotates the camera, dragging while holding down the *option* key causes the camera to zoom in or out, and dragging while holding the *command* key pans the camera.

The Color Picker controls let the viewer override the automatic color selection and choose the colors used in the visualization. The user first turns the color picker on. She then clicks on a color box, then uses the hue circle and the saturation and value sliders to determine the color in that box. For a second color, the user clicks on the color box, then adjust the controllers to find the desired color. The number of color boxes clicked determines the number of colors in the animation. The user must click the color boxes in order, but she can go back and adjust colors on previously clicked boxes. If the colors are unsatisfactory or the user wants to begin again, clicking *reset* will clear the colors so new ones can be chosen. The current picked colors are displayed in the Color Palette area to the right of the color picker controls.

Overall the interactivity controls seemed successfully simple and user-friendly. The labeled drop down menus, sliders, and buttons of the graphical user interface are familiar tools that invite the viewer to interact with the Control patch. These user interface elements make it easy to quickly change elements of the visualization. Viewers were eager to play with the commands and create their own visuals. With the combination of different animation schemes, shape primitive types, drawing primitive types, colors, textures, and background alpha values, users have many parameters experiment with to tailor the visuals to their own vision of how the music should look.

# CHAPTER VII

# CONCLUSION

The framework for visualization proved satisfactory. Relevant musical information retrieval was achieved, as indicated by the mood extraction method's accuracy. The mood classification method generally produced color palettes than matched the current mood of the visuals according to Itten's theories. From a subjective standpoint, these palettes were highly successful. The resulting visuals produced a suitable degree of visual interest. In the future, more complicated animations could be produced so as not to lose viewer interest.

However, computer processor speed proved to be an important issue. The first iterations of the visualizer were unable to run in real-time because the processor was overtaxed with calculations. Due to this reason, the visual were kept rather simple. In previous iterations I included texture and lighting generation, but they slowed the animations to a crawl. As their affects did not drastically alter the images, they were removed. In these initial stages, the animation schemes were created as Max patches. Converting the ideas of those schemes in Javascript programs and streamlining the musical analysis calculations helped the speed considerably. However, some of the animations may lag slightly if other calculations, such as user input, are occurring at the same time as the animation.

## Future Work

Future work will begin with the necessary task of improving the visualizer's efficiency by further streamlining calculations. While the visualizer does work in real-time, at certain points it lags behind. Ensuring that it is efficient enough to avoid this will make the visualizer performance ready.

Improving efficiency will also make the visualizer capable of being used for live performances. The theoretical framework has important implications for live performance. Since mood classification, beat recognition, and volume range information are all extracted automatically, a musical performer could use this system to generate imagery without having to worry about constantly needing to control the parameters. Different animation schemes that fit the performer's view of his or her music could be developed. When run they would follow the mood, rhythm, and intensity of the musician's unique performance without needing inputted cues. In a similar fashion, the performer might alter his performance to ensure the creation of specific desired visuals.

With a faster system it should also be possible to create more complex imagery. Due to concerns for processing efficiency and speed, the visualizer currently only uses the shape primitives found within the OpenGL repertoire, such as cubes, spheres, and toruses, and simple Nurbs objects. In future iterations I would like to include more complex three-dimensional models built in other applications and imported into Jitter. Similarly, it should also be possible to export the musical analysis data from Max into another graphics application, such as the open-source application vvvv or even into an

Autodesk Maya Mel-scipt. The musical data would then be able to control more complex animations than simple rotation, translation, and scaling.

# REFERENCES

Berman, G. 1999. “Synesthesia and the Arts.” *Leonardo* 32(1): 15 – 22.

Brougher, K., J. Strick, A. Wiseman, J. Zilcer, & O. Mattis. 2005. *Visual Music: Synaesthesia in Art and Music Since 1900*. New York: Thames & Hudson.

Collopy, F. 2004, October 19. *Color Scales?* Retrieved August 3, 2008, from http://rhythmiclight.com/archives/ideas/colorscales.html

Cycling ’74. 2007. Max/MSP/Jitter. [Computer Software]. Retrieved October 17, 2007, from http://www.cycling74.com/downloads/max4

Cycling ’74. n.d.. *Max/MSP/Jitter Product Overview*. Retrieved September 3, 2007, from http://www.cycling74.com/products/mmjoverview

Cytowic, R. 2002. *Synesthesia: A Union of the Senses*. Cambridge: The MIT Press.

Dann, K. 1998. *Bright Colors Falsely Seen: Synaesthesia and the Search for Transcendental Knowledge*. New Haven, CT: Yale University Press.

Dobrian, C. 2005. *MSP Tutorials and Topics*. Retrieved June 20, 2007, from http://www.cycling74.com/twiki/bin/view/ProductDocumentation/WebHome

Draves, S. 2000. “Metaprogramming Emergent Graphics.” *YLEM,* 20(6). Retrieved August 2, 2007, from http://draves.org/meg/

Draves, S. n.d.. *Bomb Dribble Index One*. Retrieved August 2, 2007, from http://draves.org/bomb/1/

Duffy, P. 2001. *Blue Cats and Chartreuse Kittens: How Synesthetes Color Their Worlds*. New York: Times Books Henry Holt and Company, LLC.

Evans, B. 2005. “Foundations of a Visual Music.” *Computer Music Journal,* 29(4): 11 – 24.

Gooding, M. 2001. *Movements in Modern Art: Abstract Art*. Cambridge: Cambridge University Press.

Hertz, P. 1999. “Synesthetic Art—An Imaginary Number?” *Leonardo,* 32(5): 399 – 404.

Itten, J. 2004. *The Art of Color*. (Ernst van Haagen, Trans.). New York: John Wiley & Sons, Inc. (Original work published 1961).

Jasch. 2008. Objects for MaxMSP. [Computer Software]. Retrieved December 12, 2007, from http://www.jasch.ch/dl/default.htm

Jehan, T. 2001. MSP Externals. [Computer Software]. Retrieved November 8, 2007, from http://web.media.mit.edu/~tristan/

*Jitter Tutorial*. 2007. Retrieved May 14, 2007, from http://www.cycling74.com/twiki/bin/view/ProductDocumentation/WebHome

Jones, R. and B. Nevile. 2005. "Creating Visual Music in Jitter: Approaches and Techniques." *Computer Music Journal* 29(4): 55 – 70.

Liu, D., L. Lu, and H.J. Zhang. 2003. "Automatic Mood Detection from Acoustic Music Data." *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) 2003*. Baltimore, MD: The John Hopkins University, pp. 81 – 87.

Moritz, W. 1989. "Abstract Films of the 1920s." *Intenational Experimental Film Congress.* Toronto: Art Gallery of Ontario: 52 – 57.

Moritz, W. 1996. "Jordan Belson – Biography." In Jean-Michel Bouhours (Ed.), *L'art du Mouvement*. Paris: Cinema du Musee National d'Art Moderne, Centre Pompidou. Retrieved September 10, 2008, from http://www.centerforvisualmusic.org/Belsonbio.htm

Owsinski, B. 2006. *The Mixing Engineer's Handbook*. Boston: Thomson Course Technology PTR.

Reynolds, C. 2001, September 6. *Boids Background and Update*. Retrieved August 20, 2008, from http://www.red3d.com/cwr/boids/

Roads, C., with J. Strawn, C. Abbott, J. Gordon, and P. Greenspan. 1996. *The Computer Music Tutorial*. Cambridge: The MIT Press.

Shaw-Miller, S. 2002. *Visible Deeds of Music: Art and Music from Wagner to Cage*. New Haven, CT: Yale University Press.

Smith, J., and J. Abel. 1999. "Bark and ERB Bilinear Transforms." *IEEE Transactions on Speech and Audio Processing* 7(6): 697 – 708.

Taylor, R., P. Boulanger, and D. Torres. 2006. “Real-Time Music Visualization Using Responsive Imagery.” *Proceedings of the 8th International Conference on Virtual Reality*. Laval, France: Laval Virtual, pp. 62 – 69.

Tzanetakis, G., and P. Cook. 1999. “Multifeature Audio Segmentation for Browsing and Annotation.” *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* New Paltz, NY: IEEE, pp. W99-1 – W99-4.

Tzanetakis, G., G. Essl, and P. Cook. 2001. “Automatic Musical Genre Classification of Audio Signals.” *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) 2001*. Bloomington, IN: Indiana University, pp. 205 – 210.

Weisstein, E.W. 2008. *Hyperbolic Tangent*. Retrieved September 3, 2008, from http://mathworld.wolfram.com/HyperbolicTangent.html

Yang, D., and W. Lee. 2004. “Disambiguating Music Emotions Using Software Agents.” *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) 2004*. Barcelona, Spain: Universitat Pompeu Fabra, pp. 52 – 58.

Youngblood, G. 1970. *Expanded Cinema*. [Electronic version]. Retrieved September 10, 2008 from http://www.vasulka.org/Kitchen/PDF_ExpandedCinema/ExpandedCinema.html

Zicarelli, D., and G. Taylor. 2006. *Max Getting Started*. Retrieved June 20, 2007, from http://www.cycling74.com/twiki/bin/view/ProductDocumentation/WebHome

# VITA

Name: Karessa Natee Bowens

Address: Texas A&M University, C108 Langford Center, 3137 TAMU,
College Station, TX 77843-3137

Email Address: knbowens@gmail.com

Education: B.F.A., Design, The University of Texas at Austin, 2004
M.S., Visualization Sciences, Texas A&M University, 2008