Synaesthesia

Karl Petre – 17 November 2006

1 Abstract

The purpose of this project is to create a fully functional and marketable software package for an intelligent lighting processor. The software will serve as an interface between a real-time audio signal and programmable lighting devices, and will use digital signal processing algorithms to detect significant events within the incoming music. The software will be designed to specifically support the decomposition of trance music; user features will be oriented toward use by a performing trance DJ.

2 Introduction

Within the music-oriented realm of intelligent lighting, there exists a fundamental and unforgivable shortcoming. Current lighting devices run pre-programmed patterns, which progress based on vibration pulses that the lighting units detect. Actuation of these lighting devices is limited to the delayed vibration information that is gathered by these sensors. Such a configuration – although quickly conceivable – is only crudely effective. Due to this fact, these lighting systems do not have the ability to truly respond to music, and much possibility for the enhancement of the listener's experience is overlooked.

The purpose of completing this project is to create a more advanced system, using a perspective that is different from that of the common design. Rather than relying on delayed vibration content, this device will have more suitable inputs: it will be hard-wired to the audio signal. Instead of running through pre-programmed visual progressions, this device's processing scheme will use genre-specific decomposition algorithms to analyze what has been played; consequently, it will detect and appropriately respond to motion within the music.

Most directly, this project will have applications in the club lighting scene, and it will therefore be designed specifically for such use. The product to be built during this course will represent only the beginnings of proactive lighting systems, and countless extensions to this work are imaginable.

This proposal presents a brief background on trance music, and provides a thorough technical description of the intelligent lighting device. Major implementation steps are given. Equipment costs are considered, along with realistic design constraints and sustainability issues. Finally, a section on my qualifications is included.

3 Trance

Trance is extremely systematic in nature, making it an ideal starting point for this field of research. There are many characteristics intrinsic to the genre, and these are present in every trance track. Each piece of music consists exclusively of eight-bar phrases that are in common time. The same rhythmic elements are typically found throughout trance: a bass kick on every beat, a snare on the second and fourth beats of each measure, and a high hat on each of the upbeats within a given measure.

The phrases in a trance track can be broken down into three main categories, based on the function of the rhythmic instruments that are present. The first, and most prevalent, is simply when all of the previously listed elements are heard. A *break* occurs when the rhythm drops out, while the melodic instruments remain. A *buildup* is the transition from a *break* back into the standard melodic section.

Every trance track also has an *intro* phase, which is a development of the track to come. During this section, the rhythmic elements are introduced sequentially. Likewise, trance tracks end with *outro* sections that break them down and similarly fade them out. These two sections exist so that the DJ may mix seamlessly from one track to another; the listener will never hear these portions of each song.

The central principle of trance lies in the fact that its music is built of change. Elements of a song that have already been heard once are absorbed and forgotten. This is why, for example, rhythmic patterns are so repetitive – even though the bass hits on every beat, the listener will not pay attention. Rather, the listener will only notice when the bass kicks enter and leave. One who is new to the genre will not immediately grasp this idea; they will never look past the monotony and prevalence of the beat, and consequently miss the concept of the music entirely.

Movement in a trace track is most nearly characterized by change in its bass line; therefore, use of this feature would be quite comprehensive for detection of change in such music. Any given trance track has a one-note bass sequence that exists essentially throughout the track's entirety. The bass line typically moves in eight-bar patterns, and some of the most important changes within a trance occur via key modulation – as created by the bass. The proposed system will capitalize on this characteristic, looking primarily at the low frequencies of audio signals to develop a real-time analysis of their essence.

4 Technical Discussion

4.1 Hardware

The Synaesthesia project makes use of existing hardware, rather than existing as a stand-alone unit. It is being coded on a Linux platform, and uses an M-Audio Delta1010LT soundcard to provide the necessary input and output interfaces. A relatively large number of inputs are needed, because the final system will take three stereo channels of data – the master output from a DJ's mixer, as well as the two unmixed cue channels.

4.2 The JACK Audio Connection Kit

JACK is a low-latency audio server, written for POSIX conformant operating systems such as Linux. It can connect several client applications to an audio device, and allow them to share audio with each other. Clients can run as separate processes like normal applications, or within the JACK server as plug-ins. JACK was designed from the ground up for professional audio work, and its design focuses on two key areas: synchronous execution of all clients, and low latency operation [1].

The Synaesthesia software package is being written using the JACK API. Use of the high-level abstraction layer provided by the API allows all hardware to be managed trivially, shifting the emphasis of the project back towards signal processing. It is desired that the Synaesthesia engine produce an output with a latency of no more than is perceivable by the listener; this is fully supported by the JACK environment.

1. For more information see http://www.jackaudio.org.

4.3 FFTW

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data [2]. It is the Fourier transform library of choice in most cases, and will be used exclusively by this project for such computations.

The PCM-coded audio signals to be processed are read from the buffer of the soundcard. At a sampling rate of 44 kHz, there is more than enough data for the desired processing. The typical buffer size is 1024 data points, although this and several other relevant values are configurable in the JACK interface.

2. For more information see http://www.fftw.org.

4.4 Tempo Detection

Once an FFT of the audio signal may be obtained, the next step will be to implement a routine for detecting the tempo of the audio signal. Taking the FFT over several buffer cycles (or more) of data should yield a spike on the order of 2 to 3 Hz. The frequency of this peak will correspond the tempo of the music.

The estimate of the tempo will be further refined using a Kalman filter, which is a recursive estimator for the state of a dynamic system. Kalman filters are especially useful when the data is incomplete or noisy, which is the case in this particular situation. The correct recognition of the tempo (and the time at which each beat occurs) is highly critical for the accurate operation of the proposed system.

4.5 Event Extraction

Musical signals carry copious amounts of information, which makes them quite complex. Noting this and the extreme diligence used in the creation of music, it is reasonable to conclude that the medium is of value to humans. This is further exemplified by the effort that has been put forth to analyze the cognitive processing in interpreting and responding to music. In particular, the automated extraction of information from audio signals by machines has been researched for the past 20 years, and much of this research effort has recently been focused in the area of melody transcription. These research efforts are quantitative; they attempt to catalog the musical elements in audio streams. The utility of these endeavors has been recognized by current researchers and many applications – ranging from music theory to data compression – have been suggested.

It is my belief that one additional layer of context would prove extremely useful in the realm of music signal processing. Rather than focusing on the precise transcription of audio streams, the overall essence of the music should be considered. In terms of its meaning and effect on the listener, one of the most appropriate mechanisms for music processing is an analysis of the changes that occur within it. For example, a transition from a minor to a major key is much more meaningful than a switch between two pitches in the melody of a particular song. As with the Gestalt idiom that "the whole is greater than the sum of its parts," it is true that music cannot be accurately described by simply producing an inventory of its elements. Across all genres, it is the changes within music that allows it to draw and hold the attention of the listener. Therefore, the recognition and evaluation of these events is critical for successful intelligent processing of music.

There are two distinct approaches that one might take when attempting to extract significant events from music. The first is that an audio processing algorithm might use a complete decomposition of the music to transcribe it, so that it may be examined using an application of existing music theory. However, much research has already been conducted along this line, yet this research has not proven fully successful. While transcription accuracy has improved, the improvement process has been gradual, and the ability to form a complete transcription remains largely elusive.

I propose to research the other methodology, which shifts the effort from a detail-oriented algorithm to one that concentrates on the larger picture. The objective of my work will be to apply signal processing to audio streams for the purpose of finding significant changes within them – such as key modulations, cadences, and noteworthy percussive events. I hypothesize that if such processing techniques are applied to audio signals on a higher level (than that of current transcription algorithms), then it will be possible to establish an efficient system for the extraction of change from music.

Once the software for implementing an FFT has been written, the next step will be to implement a routine for detecting and characterizing the changes that occur in the audio stream. An initial method will be to compare the FFT spectrums obtained from the audio data of each successive beat. By correlating certain frequency ranges of these spectrums, patterns in the music should become apparent, and these may be documented over time to produce a model of the changes that are occurring within the music.

Once this initial algorithm has been employed, it may prove useful to examine the audio signal at higher levels of abstraction. By locating the patterns that occur from one beat to the next, it may be possible to produce an algorithm for locating the downbeat of each measure of the music or the downbeat of each musical phrase. Comparing the spectral contents of measures or phrases could provide additional insight for quantifying motion within the music.

4.6 DMX

Initially, the output of the device will be simulated by printing data to the command line. The lighting output may be modeled by using the parallel port to light LEDs. Once the software has been deemed fully functional, its output will be converted to the DMX lighting standard. The user will have the ability to configure the DMX addresses to be controlled, and will be able to specify the type of lighting unit present at each address.

A DMX signal may be obtained by routing the output stream to the serial port of the computer; hardware is available for transforming an RS-232 serial data stream into the DMX standard. However, such conversion modules are relatively expensive (on the order of $1000), so other approaches may be pursued for creating the DMX signal. A DMX signals is sent over a three-pin XLR-type connector, which transmits a balanced signal using RS-485 voltage levels and cabling practices; therefore, it should be possible to use an RCA to XLR converter to produce a sufficient DMX signal. (The signal could be driven by one of the digital outputs of the soundcard.)

LED arrays are the ideal type of lighting units to be driven because they have the ability to produce a full spectrum of colors. Note that proof of concept is possible without the use of a lighting device.

4.7 User Interface

In order to improve the usability of the Synaesthesia engine, an interface should be created in order for a user to fully control its functionality. It has been suggested that Python is the appropriate language for such an implementation.

5 Project Plan

Note that the following project plan is based on the projected state of completion of this project upon entering the senior design course.

5.1 Task Analysis

5.1.1 Event Extraction

Devise an algorithm for extracting the significant changed from incoming audio streams.

5.1.2 DMX Conversion

Convert the output to the DMX lighting standard.

5.1.3 GUI Development

Develop a user interface for running the Synaesthesia engine.

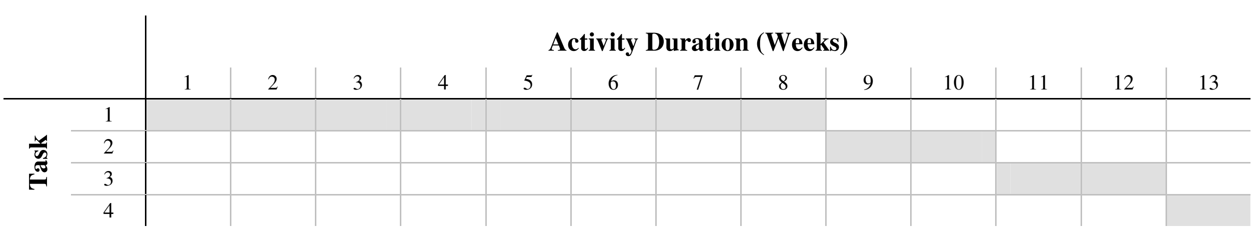
5.1.4 Documentation

Develop a research report and prepare a presentation.

5.2 Critical Path Method

	Task	Duration	Needs	Feeds
1	Event Extraction	8 Weeks		2, 3, 4
2	DMX Conversion	2 Week	1	3, 4
3	GUI Development	2 Weeks	1, 2	4
4	Documentation	1 Week	1, 2, 3	
	Total	13 weeks		

5.3 Gannt Diagram



6 Project Cost

For proof of concept, no additional costs will be incurred.

7 Realistic Design Constraints and Sustainability

Only a minimal amount of equipment was purchased for this project, which included an audio card and assorted audio cables. In addition, an existing computer workstation is being used. Creating each of these electronics components incurred a substantial environmental cost, but they will last long enough that they are practical and sustainable in the context of our society. The computer workstation has been used for several years, and it, along with the audio card and cables, may continue to be used for years to come.

Disposing of the elements of this system after the end of their life cycles will not be trivial. The computer components may be sold back to their manufacturers, so that their parts may be dealt with in an environmentally friendly manner. Manufacturers often reuse certain components, while correctly disposing of the remainder of the parts. Minimizing the environmental cost of producing electronics is the responsibility of the semiconductor industry. Without the use of computers and related parts, there would be no way of completing signal processing projects like this one.

8 Project Qualifications

As a senior engineering student at Swarthmore College, I am certain that my technical background and knowledge will enable me to complete this project. I have much experience with software programming, as well as a thorough understanding of musical concepts that have motivated my desire to imagine such a product. From a design perspective, I have a very thorough concept of the final system that I am aiming to produce, and I feel that its development will directly address an ongoing real-world problem. Moreover, I have spent much time developing this ideas present in this proposal, and believe that its fulfillment will vastly enhance the education that I have received from Swarthmore College.

9 Further Reading

For further reference, see http://engin.swarthmore.edu/~kpetre1/synaesthesia/.