University of BRISTOL

DEPARTMENT OF COMPUTER SCIENCE

An Automated Lighting Control System Based on Advanced Music Information Retrieval Techniques

Ian Calverley Lawson

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering

May 2012 | CSMENG-12

Declaration

A dissertation submitted to the University of Bristol in
accordance with the requirements of the degree of Master of Engineering in
the Faculty of Engineering. It has not been submitted for any other
degree or diploma of any examining body. Except where
specifically acknowledged, it is all the work of the Author.

Ian Calverley Lawson, May 2012

Abstract

This project is based on creating an immersive, synchronised, automated lighting control system from an audio input. The aim of the project is to use advanced music information retrieval techniques to extract features from an audio input. These collected features will be manipulated and combined to construct a segmented mood classification algorithm and to identify significant points in the audio where lighting cues can be placed. A live-feed from a microphone will be integrated into the system to allow it to react to captured environmental noise. The results of the lighting system will then be output to a suitable environment where they can be tested and demonstrated.

In this project I will explain the latest music information retrieval techniques and their current applications. From the technical definitions I will show how these techniques can be adapted to create a piece of software that automates a difficult manual process. The lighting system will need to be accurately synchronised to the provided audio input, and reflect the change of mood throughout the song. It will also need to work on a wide range of audio tracks to prove its success.

The project was chosen to address a common problem that I have identified in many music venues over the last few years. Complex lighting systems are present in almost all music venues and are never automated to the music that is playing. Accordingly, this results in an uncoordinated light display or hiring the expertise of a trained stage lighting professional. This project aims to discover whether this process can be automated and how much useful information can be extracted from a piece of music to create a lighting display.

The project has required a strong understanding of both signal processing and musical theory. Undertaking this project has allowed me to combine my passion for music with the technical skills I have acquired during the course of my degree.

The following list highlights the main achievements of this project:

- I have written just over 1000 lines of code, with the majority contained in the core implementation (in C++) and a fraction in producing the output (in MEL).
- I have combined a selection of advanced music information retrieval techniques in an innovative way to create a lighting control system.
- Within this system I have implemented an intelligent approach to mood classification of an audio track - breaking the audio track down into relevant segments and classifying those segments.
- I have spent around 80 hours finding appropriate music information retrieval algorithms and understanding the Vamp Audio Plugin system.

Contents

1 Abbreviations

- MIR - Music Information Retrieval.

- C4DM - Centre for Digital Music, Queen Mary, University of London.

2 Introduction

2.1 Background

Automated lighting refers to stage lighting that possesses some form of intelligence. In this field, the intelligent behaviour is produced by sophisticated engineering of the lighting hardware. These intelligent lighting systems combine elements from a wide range of technologies including mechanics, electronics and robotics. They can produce eye-catching effects such as colour shifts, pans, tilts and prism rotations, all of which are enabled by their advanced mechanical design.

The concept of automated lighting can be traced back to at least 1906, when Sohlberg was issued a patent for his novel light-projecting apparatus [1]. Sohlberg's unmechanical form of automated lighting controlled a carbon-arc bulb with a number of cords that controlled pan, tilt and zoom. The invention aimed to enable the operator to remotely control the light fixture in a concealed location using a simple set of controls. Since then, technological advancements have been made, with cords being replaced by electronic controllers and complex mechanisms integrated into the light fittings to increase the number of possible effects. From 1981 to the present day automated lighting has gained significant recognition and popularity in the entertainment and "architainment" lighting industries [2].

However, for these intelligent lighting effects to be produced they need to be controlled by an intelligent lighting programmer. Typically, sequences of lighting effects are pre-processed and played back using a controller - rarely are lights controlled live by an individual.

2.2 Aims

This project is based on gathering state of the art music information retrieval techniques in order to produce an automated lighting control system. The final lighting system needs to be immersive and synchronous with any given audio input.

The project combines research from a number of fields, including the psychology of moods and their relationship with colours, to digital signal processing techniques in the complex domain. The first area that needs to be investigated is feature extraction. There are a wide range of MIR techniques that can gather features such as beat locations, changes in pitch and chord patterns. I will need to examine which of these features are required to build the lighting system and locate appropriate MIR techniques that can be used on my development platform.

Once the information has been extracted it will then need to be analysed and filtered to generate useful input for my lighting system. I will need to investigate how the selected MIR techniques work, and with an understanding of the algorithms, optimise parameters to produce the best results.

Throughout the process of creating this system, I will also examine the methods of mood classification of an audio signal, and what effect mood has on a lighting system. Mood is a high-level feature which can be conveyed by both audio and visual stimuli, and the relationship between the two is an interesting concept. In this project I will look at a novel method of mood classification based on splitting a musical piece into a number of structurally different sections and classifying these sections individually. The motivation for this proposal is based on the fact that no musical piece ever stays the same, and that there is variation between sections such as the chorus and verse. By identifying and understanding the mood in each of these sections a lighting system can change and adapt throughout a piece of music, offering a more immersive display.

I will also look into the integration of a microphone to my system, which will act as another source of information that can be used by the lighting system. By capturing environmental noise in the same location as the lighting display, the system can gain an understanding of its environment. This will allow the lighting display to adapt in real-time and increase the performance of the overall system.

The final problem will involve finding a suitable output for my lighting system. The output must be able to demonstrate every element of the lighting control system. To ensure I am not constrained by the output hardware, the lighting will be shown using a suitable piece of software.

At the end of the project I aim to have produced a working system that can produce a lighting display for any input. The system will need to identify strong note onsets and beat locations in the audio signal to produce a set of accurate cues. The mood classification system will be tested on a set of audio samples to ensure that it identifies the mood correctly.

2.3 Motivation

Lighting displays are used worldwide in a number of different locations to provide a visual stimulus alongside some other form of media. The most common location for these light displays is at live music venues where they are used to highlight bands playing on stage, or illuminate a dance floor. Lighting is an important element in these environments and is sometimes overlooked, with focus put on providing high quality audio. However, lighting is perhaps the most striking aspect of a venue; lighting transforms a room, shifts the mood and engages the visual senses to create a memorable and immersive atmosphere.

In the UK alone there are a total number of 1675[1] listed nightclubs, all of which have lighting setups of varying size and complexity. The majority of these systems are either operated randomly by a computer or manually by an individual. With the advancements of music information retrieval, it is now a possibility to automate this process and in doing so guarantee a much higher level of performance.

Currently, high quality lighting displays are produced manually by professionals in real-time. The tools for producing these displays vary, but it is usually done using a MIDI controller, with a set of stored light sequences. With enough training and practice, video performance artists are extremely capable of producing masterful lighting displays, but as with any skill, it differs from person to person. Extreme care and attention needs to be paid to keep the lights completely synchronised with the music; the method is also susceptible to human error.

Proprietary software has been created that attempts to automate this process, but in practice it is extremely limited. From my research, it seems that the leading piece of technology is produced by a software company based in New York called Light-O-Rama.[2] Their software suite supports both MIDI and audio input data and provides the user with an interface to select a track and view its waveform. They then approach the task of creating an automated lighting display using a small number of features. They utilise a beat detection algorithm and perform some amplitude analysis to generate a set of lighting cues. These are then presented to the user, who can then choose to add lighting effects at these given points in time. The software can randomly add these effects if the user does not have time, but this randomness and lack of innovation results in the software producing odd and unsynchronised lighting shows. If any music-dependent light show is to be created from this software, it demands that hours of time be spent by the user to create it themselves.

MIDI is an electronic musical instrument industry specification that symbolically describes the electronic steps required to generate sounds. Because of its representation, it is easy to break down and understand, but MIDI tracks cannot contain vocal content or produce realistic sounds. As the aim of my project is to produce a system that can be used at live music venues, I have decided not to look at using symbolic inputs and instead have focused on the harder task of analysing polyphonic audio signals.

An alternative to Light-O-Rama is the open source lighting software Vixen.[3] Vixen contains

[1] yell.com

[2] lightorama.com

[3] vixenlights.com

even less technical functionality than Light-O-Rama but encourages it's community of users to develop and share plugins. The main problem with the software solutions mentioned is their lack of focus to a certain market. Both Vixen and Light-O-Rama have attempted to make automated lighting software that can be used easily in many different ways. But because of this mentality they have chosen to implement a small number of MIR algorithms; always leaving the user to do something manually to finish the display.

The aim of my project is to solve the deficiencies of the current software options by fully automating this process. By using as many relevant music information retrieval techniques as possible, I aim to create a piece of software that can create a realistic and immersive light show. This technology can be integrated into a software package and sold to owners of entertainment venues or even to enthusiastic home users. It can save these entertainment venues large amounts of money by removing the need for them to hire someone to produce a lighting display. If executed correctly, it should also perform better than existing software solutions and at a high consistent standard that could rival a professional lighting engineer. The technology can also be expanded to a number of other environments at a later stage, e.g. it could be used to trigger fireworks at public events and ceremonies.

The development of this automated lighting control system will also be a starting point for other researchers and academics. As music information retrieval techniques continue to improve, extracted high-level features can be integrated into my system to improve its performance.

I also hope to provide a new method of mood classification that can be used in a number of other applications. Mood classification of audio is currently used in music classifiers and recommendation systems as well as other applications. Moodagent[4] is a recently published application available on a number of mobile, desktop and web platforms that is based solely on mood classification of music tracks. It can either generate a playlist of songs given a mood such as tender or angry, or generate a playlist based on the mood of a given track. As music databases grow, more efficient mood classification methods are needed. Software of this nature is based on accurate mood classification algorithms; and I hope that by analysing the mood of a track using a divide and conquer approach I can offer a competitive alternative.

No real research has been devoted to automated lighting, instead, the focus has been on the underlying techniques of MIR. These feature extraction mechanisms are currently used in a number of other applications including recommender systems, automatic score creation of audio, automatic categorisation of audio and music generation. These MIR techniques extract low-level

[4] moodagent.com

and mid-level features from an audio signal, such as note onsets, amplitude peaks and tempo. I plan on using these to generate a high-level feature - mood. I will use and extend existing methods of mood classification but apply these in a different way to an audio track. Previous research into the effect of colours on mood will be used to help portray my mood classification results in the output lighting display.

2.4 Challenges

One of the main challenges of this project is finding a set of reliable MIR algorithms. It is outside the scope of this project to develop a collection of MIR techniques myself to accompany the lighting system, so I need to ensure that the selected MIR algorithms work on my development platform and produce accurate results. If the results produced by the MIR algorithms are inconsistent, they cannot be used, as this inaccuracy will directly impact the results of my mood classification and lighting cues. Once located, the performance of any MIR algorithm will need to be tested and analysed before it can be used in my system. If any MIR algorithm can be modified slightly to increase performance, it will be investigated. It is necessary for these MIR algorithms to work in real-time if they are to be combined with the captured environmental sound.

Another challenge of this project is understanding how to map a set of extracted numerical values from an audio input to a single long lasting emotional state. Furthermore, it needs to be identified which extracted values hold more importance, meaning research will have to be carried out to examine the effect of these features on mood states.

The main task in this project lies in using well known musical features and applying them to a real-world application. But given the future possible applications of my system, the central challenge is to make this lighting system perform well on a wide range of musical tracks from a selection of different genres. This means that my implementation must be tolerant to changes in input and produce consistent results every time.

Performance of the system will be measured by the synchronicity of the lights to the provided audio track and by the classification of the mood. A synchronous light show that correctly identifies the mood of the track will be considered as a positive result. But only with consistency over a large sample set can the project be deemed as a success and considered as a complete solution to automated lighting control from and audio input.

The main aims and objectives of this project are:

- Research and understand current MIR methods.

- Use relevant MIR techniques to create a lighting control system from an audio input that performs well over a large sample of inputs.
- Create a segmented mood classification algorithm that can be integrated into the lighting system.
- Integrate environmental noise using a live feed from a microphone which will affect the lighting display in real-time.
- Output the lighting system so that it can be tested and demonstrated.

2.5 Thesis Structure

The technical background of MIR techniques and musical theory is presented in Section 3. In this section, MIR algorithms relating to project will be summarised and explanations of common musical terms will be given. Mood classification will be explained, and the relationship between music, emotion and colours will be explored. The science of stage lighting will also be investigated.

The software tools used in this project will be explained in Section 4. The choice of these tools will be justified and the usage will be explained.

In Section 5 the design and implementation of the lighting system will be presented. An explanation will be given on how the MIR techniques are used to extract the required information from the audio signal, then how this information is used in mood classification and lighting cue identification. The proposed design of the lighting display will then be explained, and results from testing will be taken to evaluate the performance and accuracy of the system.

Finally, in Section 6, the work will be concluded and critically evaluated. Project aims will be assessed and future plans will be highlighted.

3 Technical Background

3.1 Digital Signal Processing

A sound is a sequence of waves that travel from a source to a listener that can be represented in the time domain by a set of sinusoidal waves. In digital signal processing the shape a sound is known as its waveform.

A sine wave is a trigonometric function that produces a smooth, repetitive oscillation. In its most basic form a sine wave can be represented by the equation:

$$y(t) = A \cdot sin(\omega t + \phi)$$

The three parameters of a periodic waveform are its phase (timing), its amplitude (volume) and its frequency (pitch). The amplitude (A), is the magnitude of change of the function from its centre position. The frequency (ω), specifies how many oscillations occur within a time interval. The phase (ϕ), specifies where the oscillation cycle begins. A collection of sinusoidal waves with different amplitudes, frequency and phase parameters can be used to approximate any periodic waveform [3].

3.2 Musical Theory

In this section, musical concepts and definitions relevant to the implementation will be explained.

An onset marks the beginning of a musical note or sound found in an audio signal. This is then followed by an attack, which is identified as a sharp rise in amplitude from zero to a maximum amplitude peak. The attack is then followed by a decay phase; the characteristics of these two phases have a large effect on an instrument's sound.

Timbre is what distinguishes a given musical note, sound or tone from another, even if they have the same pitch (note) and amplitude (volume). Timbre describes the physical characteristics of a sound and is responsible for making a sound produced from a guitar sound different from another musical instrument.

A transient is defined as the period in time when the signal changes quickly and in an unpredictable way. The start of a transient is marked by a note onset. A musical note must have an onset but does not have to contain an initial transient. These terms are shown in Figure 1.

A beat is a unit of time in music that indicates the steady pulse of a song. Beats are rhythmically organised by the time signature and are given speed by the tempo of a musical piece. The tempo

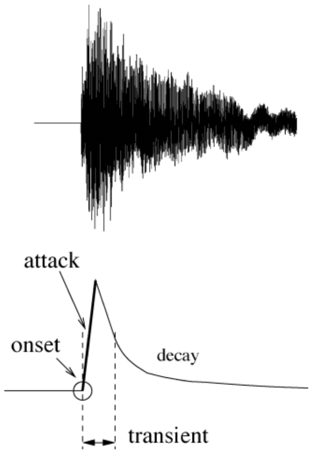


Figure 1: The onset, attack and transient of a musical note [4].

is usually indicated in beats per minute (BPM).

A time signature specifies how many beats are in each bar and what note value constitutes one beat. For example, the time signature 4/4, also known as common time, specifies four quarter-notes per bar.

When working in common time, the first beat of the bar, known the downbeat, usually signifies a strong accent in the melody or a chord change. The next downbeat beat after the first is the third beat of a bar. These two strong beats are called "on" beats, and the weaker second and fourth beats are called "off" beats.

A tuplet, also known as an irrational rhythm or grouping defines any rhythm that involves dividing a beat into a different number of equal subdivisions that is usually permitted by the time signature. The most common example of a tuplet is a triplet. Usually, a quarter note (crotchet) contains four sixteenth notes (semi-quavers), but it can also contain three triplet eighth notes (quavers). These three triplet quavers span over the same time as the four semi-quavers and the duration of each triplet quaver is 2/3 of a standard eighth note.

In tonal music, a piece is said to be "in" a certain key, for example G major or E minor. The key refers to a single, discrete tonal centre which is determined by the relationship of pitches and its associated scale. In classical music a primary key is usually given in the title, e.g. Violin Sonata in A major, however there can be numerous related and unrelated keys in a given music piece.

Keys are usually identified as being either major or minor. In music theory, a major chord contains a root note, a major third and a perfect fifth. A minor chord differs from a major chord by containing a minor third rather than a major third. This alteration lowers the third note by a semitone and greatly changes the mood of the music; a song in a minor key sounds darker and more serious. Major and minors keys can be referred to as modes.

A semitone is the smallest musical interval used in tonal music. A chromatic scale is a musical scale that contains twelve pitches, each a semitone apart. In an equal-tempered chromatic scale these semitones are all the same size.

3.3 Onset Detection

Onset detection is an active research area in signal processing and an essential component to a number of MIR techniques. The main aim in onset detection is to look for transient regions in the audio signal. A transient region can be identified in a number of different ways, because of this a number of different approaches to onset detection have been created.

After reviewing a number of current onset detection techniques proposed in [4, 5] it became clear that the method of onset detection can be reduced to three steps that are followed in most implementations. The first step is to pass in the audio input, the second is to transform the input audio into a detection function and the final part is to pick the peaks of the detection function which give the note onset locations. These steps are shown in Figure 2.

The detection function transforms the original signal in the time domain to a suitable and simpler one-dimensional form that can locate onset transients. A robust detection function should consist of sharp peaks located at transients which can then be assessed during the peak picking stage. Peak picking should only locate note onsets and its success greatly depends on the quality of the detection function. The more false peaks that are presented by the detection function, the more work must be done to ensure the right peaks are picked. Because of this, research is focused in generating strong detection functions which make the exercise of peak picking trivial.

The simplest approach to onset detection involves analysis of the audio signal in time domain. Using the time domain samples, high amplitude surges in the audio signal can be identified. This would be a reasonable solution if only a single instrument was used, but most audio signals are polyphonic and contain a number of instruments with different timbres and amplitudes. Therefore, this simple approach leads to either a large number of false positives or false negatives. Unless the signal can be broken down into its constituent parts perfect onset detection is impossible, but many algorithms have been proposed to try and get close to an accurate estimation.

The onset detection method that is used in my project is proposed by Duxbury, Bello, Davies

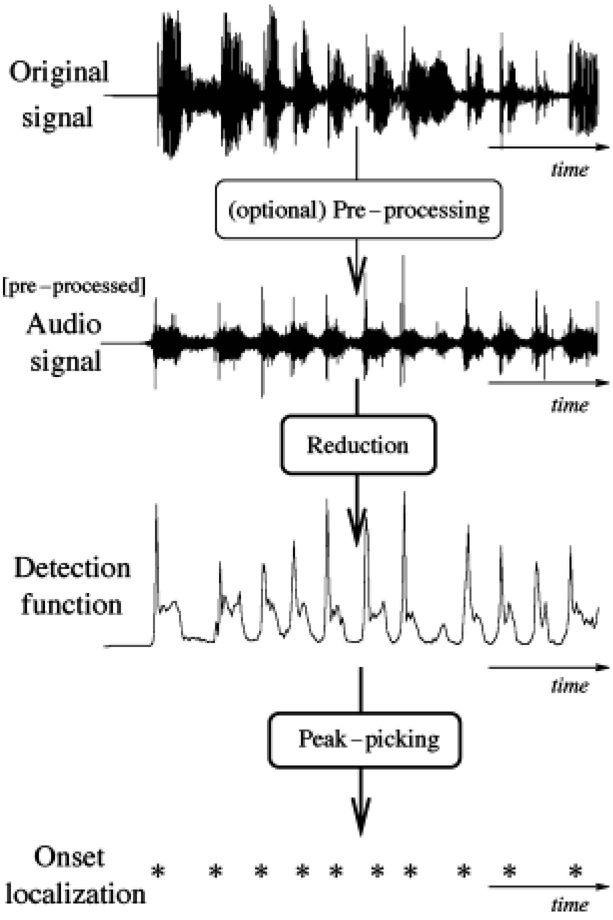


Figure 2: Flowchart of a typical onset detection algorithm [4].

and Sandler [6]. They present a novel method for onset detection which combines energy and phase information that is gathered in the complex domain.

The method described by Duxbury et al. combines two existing techniques, the first is on energy-based onset detection. In a musical signal the introduction of a note onset will lead to an increase in the energy of the signal. For strong percussive note attacks this increase is extremely prominent. Consequently energy-based methods perform well for these "hard" note onsets. The energy of a signal is defined as the area under the squared signal and can be represented by the following formula:

$$E(m) = \sum_{n=(m-1)h}^{mh} |x(n)|^2$$

Where h is the hop size of the analysis window and m is the hop number. Differentiating $E(m)$ produces a detection function which can then be used to identify note onsets using a pick peaking scheme. Energy-based methods are computationally efficient and easy to implement but are not as effective on less pronounced or "soft" note onsets.

The other common analysis method in the complex domain is phase-based onset detection [7]. Fourier analysis allows a continuous signal to be represented as a group of sinusoidal oscillators. During a steady section of the audio signal, the sinusoidal oscillators tend to have constant frequencies, resulting in consecutive phase values remaining constant. During a transient phase, values will deviate from the norm in the phase spectrum. These observations can be observed and statistically analysed to form an effective detection function for isolating “soft” note onsets.

Identifying correct note onsets from the detection function is done using a peak picking algorithm with a variable threshold value. This value can be set manually, but it is usually done automatically, either globally or locally. Peak-picking algorithms attempt to identify the median average of a signal within a sliding analysis window and use this as a threshold value. Any peaks above this threshold value are classified as note onsets.

The combination of energy and phase information results proposed by Duxbury *et al.* produces an accurate onset detection algorithm. The algorithm was tested on a range of polyphonic music samples which contained hand-labelled onsets. Different peak picking threshold parameters were tested, and assessed on the number of good detections and the number of false negatives. The optimal results gave a 95% good detection rate with 2% false positives.

Note onset locations are a valuable feature that will be needed in my implementation when trying to identify lighting cues. Therefore, an onset detection algorithm will need to be used in my system.

3.4 Beat Detection

The goal of beat tracking is to replicate the human ability of tapping in time to music. Given an audio signal a beat tracking algorithm is required to return beat onsets times. The task of beat tracking is closely related to onset detection that has been by explained in Section 3.3. It is also essential to other music information retrieval MIR techniques such as structural segmentation and chord estimation.

The beat detection algorithm used in my project is documented by Davies and Plumbley [8]. They provide a solution that can process both audio and symbolic signals without any prior knowledge of the input. The algorithm is also capable of following changes in tempo.

The first step of the beat detection algorithm is to transform the input into a more meaningful and compact mid-level representation. Using the original audio signal, a complex spectral difference onset detection function is generated. To allow the tempo and phase of beats to vary over the

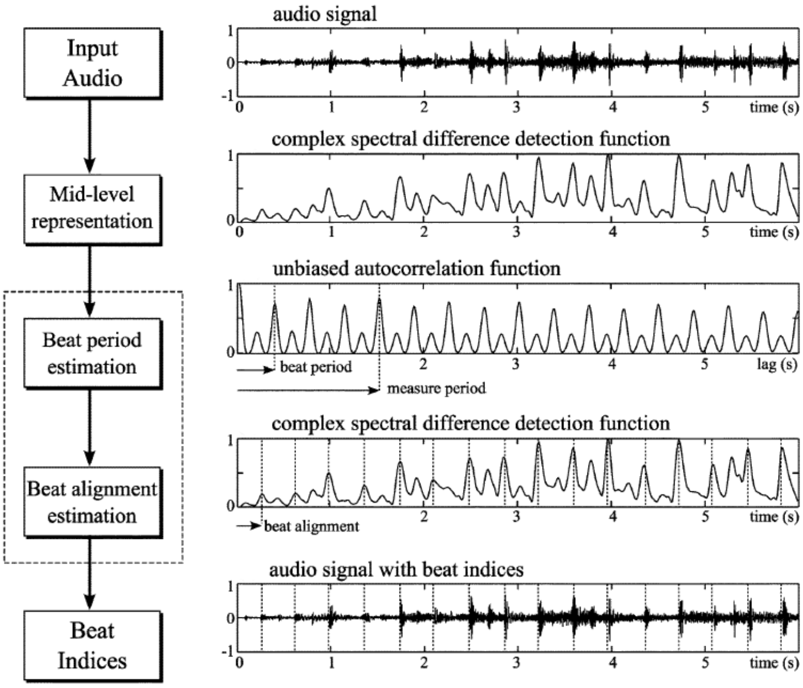

Input Audio
Mid-level representation
Beat period estimation
Beat alignment estimation
Beat Indices
audio signal
complex spectral difference detection function
unbiased autocorrelation function
beat period
measure period
lag (s)
complex spectral difference detection function
beat alignment
audio signal with beat indices
time (s)
1
0
-1
0
1
2
3
4
5

Figure 3: Flowchart of a typical beat detection algorithm [8].

audio signal, the onset detection function is partitioned into overlapping analysis frames. Each analysis frame is constructed using a window of 512 onset detection function samples, each 11.6 milliseconds in length, with a step increment of 128 samples. This gives each analysis frame a size of just under 6 seconds in length with a 1.5 second step increment. The size of this analysis frame is long enough to estimate the tempo, and the step increments are short enough to track any changes in tempo that might occur.

To identify the beats correctly, the detection algorithm makes two passes over the audio signal. These passes are known as the General State (GS) and the Context-Dependent State (CDS). The purpose of the GS is to estimate the beat period (time between successive beats) and the beat alignment, without any prior knowledge of the input audio signal. This is done through a process of repeated induction. However, without any prior knowledge no effort can be made to enforce continuity.

This lack of continuity during the first pass results in two common errors during the beat detection process. The first is the switching of metrical levels, i.e. the beat changes to either double time or half time. The second error involves the estimated beat position moving from on-beat to off-beat. The consistency of beats over the entire audio signal is a vital element of beat tracking

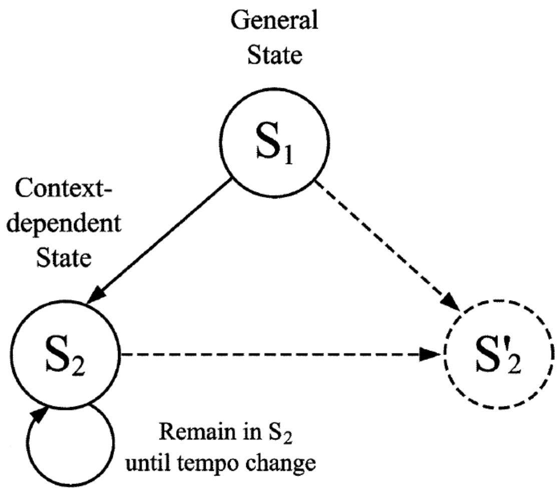


Figure 4: Diagram of the beat detection two-state model [8].

and is essential to building any musically influenced system.

These unwanted errors produced from the GS are resolved by the CDS. The CDS extracts the beat period and beat alignment in a similar way to the GS but uses context-dependent information learnt from the first pass. This newly acquired information about the tempo and beat locations allows the CDS to focus on maintaining the continuity of the beats over the whole input. The CDS prevents these errors by limiting the range of likely beat periods and alignments based on the entire set of data generated from the first pass. However, the CDS cannot adapt to tempo changes. To resolve this issue and guarantee near perfect beat tracking the GS and CDS work in harmony to form a two-state model [9].

Within the two-state model the role of the GS is to estimate the initial beat period and detect any future tempo changes. The CDS is tasked with maintaining the continuity of the beats throughout the song.

When the two-state model is initialised, we start in the general state S_1. As soon as constant beat periods over three consecutive frames have been observed the context-dependent state S_2 is activated. At this point S_1 continues to work in parallel with S_2, looking for tempo changes and rhythmic variations. If at any point S_1 identifies a change in tempo, S_2 terminates and grants control back to S_1 for a frame of audio. A new context-dependent state S_2' is then created for the new tempo value and continues to maintain consistency in the beat output. The two-state model is summarised in Figure 4.

This beat detection algorithm has been rigorously tested and the results have shown that it works best for musical tracks within the Rock and Dance genres, where beats are prominent and the tempo is usually constant. It performs at the same level as the current state of the art beat detection algorithms with a significant reduction in computation. Results of their non-casual system, for allowed metrical levels with enforced continuity gave a beat tracking accuracy of 68.1%. The main flaw of this algorithm is that it does not work in real time as the audio track has to be analysed first before any beat locations can be output.

Acquiring the locations of beats from an audio signal is essential to my lighting system. Beats give music structure and are the backbone to any music related application.

3.5 Key Detection

Automatic key detection is a fairly recent discovery in terms of MIR techniques and has become an important part of tonal analysis. The key detection algorithm used in my implementation is proposed by Noland and Sandler [10] and is based on Hidden Markov Models (HMM).

A HMM is used to model a system containing hidden states, that is assumed to be a Markov process. A Markov process is a stochastic process that satisfies the Markov property, meaning that the process is "memoryless", i.e. the process is conditional on the present state of the system, its future and past are independent [11].

Like all of the other MIR techniques mentioned so far in this project the first stage of the key detection algorithm is to transform the raw audio data into a form that is more suited for tonality estimation. As we are interested in tonality rather than onsets, we do not use a detection function, instead we transform the original audio signal to a logarithmically spaced frequency representation. This transformation is chosen because of the natural relationship between pitch and frequency. Pitch is perceived roughly as the logarithm of frequency, which means that audio samples can be mapped to the notes of the equal-tempered chromatic scale. The use of logarithmic scaling is also intended to imitate the response of the human ear [12].

Central to the key detection algorithm and many other tonality-estimation algorithms are tone profiles. A tone profile is a twelve-element vector which represents an equal-tempered chromatic scale. Each element of the vector represents a specific pitch. The importance of certain elements in a tone profile vector can be analysed to ascertain a musical key. It is assumed that tone profiles have rotational symmetry, so the importance of the C element in the C major chord is the same as the importance of the E element in the E major chord. This results in two well defined tone profiles, one for major keys and one for minor keys. These tone profiles can either be created

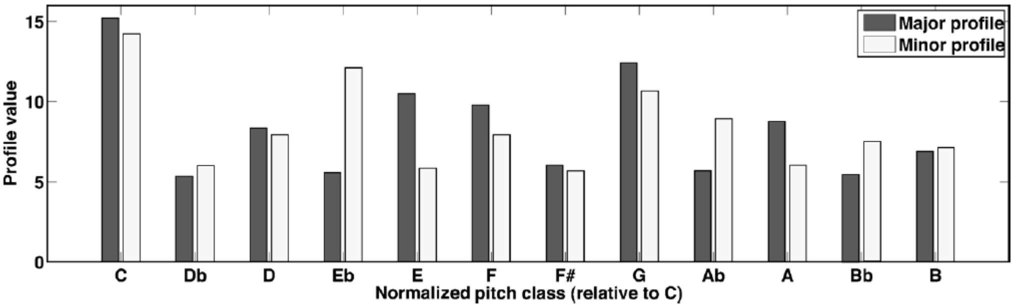


Figure 5: Krumhansl's tone profiles showing the relative importance of pitches in a key [10].

based on musical theory, cognitive studies or statistics from real music. Profiles derived from cognitive studies were chosen for this implementation because Noland and Sandler believed they provided more information about the relative importance of pitches in a key. The tone profiles used in the algorithm were generated by Krumhansl and are shown in Figure 5.

The process of key detection proposed by Noland and Sandler relies on identifying chord patterns and progressions. They believe that understanding the sequence of chords over time can help locate the tonal centre of a musical piece. This initial-chord recognition step generates a spectrogram. A spectrogram is a time-varying spectral representation that shows how spectral density of a signal changes over time. The generated spectrogram shows the energy at different pitches (regardless of octave). A twelve-bin chroma vector is then created for each frame, which gives an energy measure for each of the twelve pitch classes. This chroma vector is then multiplied by all possible rotations of the four main chord templates (major, minor, diminished and augmented). The template and rotation with the highest value is identified as the correct chord for the current frame. The performance of the key detection algorithm is greatly influenced by the accuracy of this chord recognition step.

These chord progressions are modelled by a discrete HMM, shown in Figure 6. The HMM contains 24 states for all possible major and minor keys and is fully connected. Each state of the HMM represents a key and each observation in the system consists of a pair of consecutive chords. All HMMs contain three sets of probabilities, these are the initial state probabilities, state-transition probabilities and observation probabilities. The initial state probabilities define the likelihood of the model starting that state (with no prior information given to the system all of these probabilities are set to zero). The state-transition probabilities represent the likelihood of changing from one state to any other state. It is expected that that the music will usually stay in the same key, however if it is to move to another state, it will be to a key that is

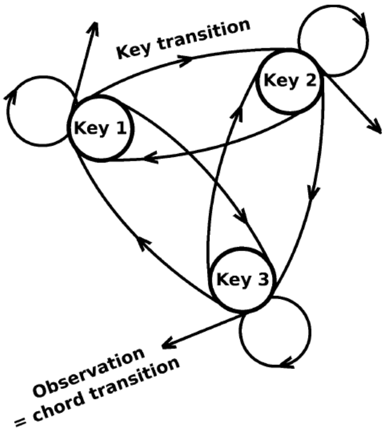


Figure 6: Diagram of the beat detection two-state model [10].

closely related. A closely related key in one that shares many common tones with the original key. The state-transition probabilities are predetermined using key-profile correlations given by Krumhansl [13]. The observation probabilities represent the likelihood that given the model is in one state, of the model emitting a particular observation. The initial observation probabilities should represent the human expectation of the key(s) implied by a given chord transition.

For a given audio track the expectation maximisation (EM) algorithm is used to learn the HMM parameters. HMM decoding is then used to obtain the posterior-state probabilities which show the likelihood of the process being in any key at each time frame. For global key estimation, these probabilities are totalled across the time domain, and the key with the largest likelihood value is chosen as the key of the song.

The key detection algorithm was tested on two sets of audio samples. The first set consisted of the songs from the first eight Beatles albums and the second test set consisted of 48 preludes and fugues by J. S. Bach performed on a piano. For the first data set the key was correctly identified 72% of the time. The second had a higher success rate of 94%. It was hypothesised that this variation was produced by the content of the recordings; a clean piano recording would be a lot less distorted than a full band performance.

Learning the key of a piece of music is the most important feature in relation to mood classification because a musical key can convey emotion. For this reason alone a key detection algorithm was integrated into my system.

3.6 Segmentation

A piece of music consists of several structurally different sections. The high level structure of these sections is referred to as “musical form”. A musical piece tends to be written in a particular form that conforms to a widely-recognised structural pattern. The most popular musical form is Verse-Chorus which consists of the following components:

intro-verse-chorus-verse-chorus-bridge-verse-chorus-outro

It is possible to implement a structural segmentation algorithm on the assumption that every musical piece must correspond to one of these widely-recognised patterns. However composers and musicians rarely follow these canonical forms without diverging away from it at times - choosing to change the length of certain sections or remove them altogether. The strong assumptions this method is based on make it unreliable on larger, less predictable data sets.

State of the art structural segmentation should be based on informative musical features like drum fills which signify a change in rhythm and the structure of lyrics. However, segmentation is limited by the music information retrieval techniques that are currently available. Current methods observe changes in timbre and harmonic features to identify section boundaries and also look for repetitions of similar sections in the audio signal [14].

Levy and Sandler have created a segmentation algorithm that that considers the intrinsically hierarchical nature of a piece of music [12]. They base their model on the fact that in music, notes are grouped into beats, beats are grouped into bars, bars are grouped into phrases, and phrases are grouped into sections. They also assume that the timbre will be the same in corresponding structural elements of music.

The proposed method works in a similar way to the key detection algorithm mentioned above in Section 3.6. A transform is applied to the original audio signal which produces a logarithmically spaced frequency representation. Timbre features are then extracted at frequency bands for each note in the equal-tempered chromatic scale, using a hop size equal to the beat-length (300-400ms) and a window size three times the hop size. The extracted features are then normalised and Principal Component Analysis (PCA) is used to extract the 20 principal components of each window. This yields a sequence of 21-dimensional feature vectors.

A 40-state HMM is then trained on the whole sequence of feature vectors. This process splits the timbre-space of an audio track into the 40 possible states. As mentioned above the assumption of the model is that the distribution of these features remains consistent over similar structural segments. The HMM is then decoded and the musical piece is assigned a sequence of timbre-features

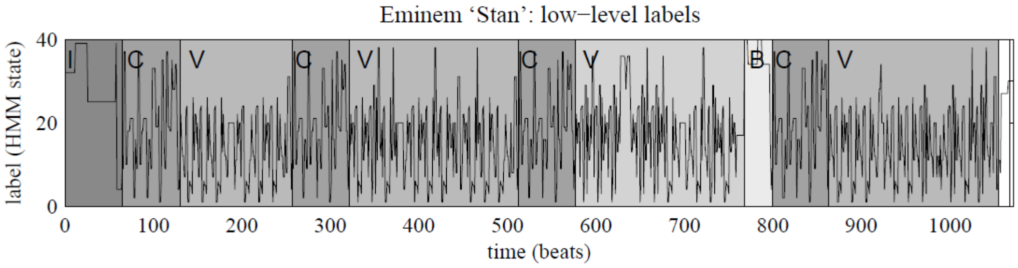


Figure 7: Sequence of low-level labels against manual segmentation [12].

according to specific timbre-type distributions for each possible section. In simpler terms, a series of states, based on timbre-features, are created that represent the song. Each structural segment of the song should contain a specific pattern of these states, which can be used to as a form of identification.

A pass is made over these states and a series of histograms are produced based on timbre-types; a histogram is created for each analysis window. These histograms are then grouped into clusters using an adapted soft k-means algorithm, with each cluster corresponding to a specific structural segment type. The histograms describe the timbre distribution for each section. The final segmentation is calculated from the cluster assignments. Figure 7 shows the segmentation results of the song "Stan" by Eminem, the colours show the segmentation results which are accompanied by letters that represent the manually labelled sections.

The segmentation algorithm was tested by Levy and Sandler on a set of 60 tracks. Their performance metric was to compare labelled beats from the segmentation algorithm against a set of manually labelled reference beats. Overall, they found that 60.3% of the labelled beats produced from the system were assigned to the correct segment. This might seem inaccurate, but the test was based on assigning beats to the correct components of the musical form of the track (verse, chorus, etc.). The segmentation algorithm can guarantee that the structural sections it produces will have strong similarities, and for my implementation which does not need section labels this is perfectly acceptable.

My proposed mood classification algorithm is designed to work on segments of a given audio track. Without a segmentation technique the mood classification could not be implemented, hence why the segmentation algorithm mentioned above has been described and analysed.

3.7 Emotion & Mood

Defining emotion has proved to be a major issue within its own field of research, causing confusion and heated debates between academics. In 1981 Paul and Anne Kleinginna attempted to resolve this issue by gathering all 92 definitions available at the time to construct the following description of emotion.

> Emotion is a complex set of interactions among subjective and objective factors, mediated by neural-hormonal systems, which can (a) give rise to affective experiences such as feelings of arousal, pleasure/displeasure; (b) generate cognitive processes such as emotionally relevant perceptual effects, appraisals, labeling processes; (c) activate widespread physiological adjustments to the arousing conditions; and (d) lead to behavior that is often, but not always, expressive, goaldirected, and adaptive [15].

A mood is more easily defined as a relatively long lasting emotional state that can be classified in terms of valence and arousal. A mood can either have a positive or a negative valence which represents a good mood or a bad mood respectively. Arousal represents a persons level of energy which can range from lazy and lethargic to excited and scared.

3.7.1 Psychology of Music

It is well established that a piece of music can be emotive, but understanding what musical aspects of a song convey emotion to create a mood is an interesting subject of research.

Being able to identify the key of a song is something only musically well-trained people can do, but almost everyone can identify the mood of a piece of music. It is well known that many composers carefully assign certain affections or emotional characteristics to different keys. Schubart published a list of characteristics for each musical key in 1806 [16]. Some examples of these include D♭ Major, which is described as, “A leering key, degenerating into grief and rapture. It cannot laugh, but it can smile; it cannot howl, but it can at least grimace its crying. Consequently only unusual characters and feelings can be brought out in this key.” He also described F major as “Complasiance and calm” and G minor as “Discontent, uneasiness, worry about a failed scheme; bad-tempered gnashing of teeth; in a word: resentment and dislike.” Two Russian composers Rimsky-Korsakov and Scriabin even tried to link musical keys to particular colours. However when they compared results, the composers found that their associations were completely different, disproving their hypothesis.

Tempo has a strong effect on the mood of a song, especially on the arousal. A quick tempo will result in a fast and upbeat musical track that can trigger such emotions as excitement, fear and surprise. On the other hand, a slow tempo will result in a more relaxing piece of music, which can elicit emotions such as sadness, boredom and calmness.

Husain *et al.* closely examined the effects of tempo and mode on mood [17]. They created a symbolic representation of a Mozart sonata in a MIDI file and varied its tempo and key. One of these variations was then presented to a number of listeners and levels of arousal and valence were measured. The results from these tests showed that a change in key directly affects the valence, but not arousal. They also showed that a change in tempo affected the arousal but not the mood.

It has been shown by Bhatata *et al.* that a change in amplitude of a musical piece provokes an emotional response [18]. They demonstrated this by varying the amplitude in piano performances, and then examined the way in which these variations effected emotions of listeners.

3.7.2 Psychology of Colour

It is common for people to associate colours with certain emotions and moods. It has been shown that colours can also affect an individual's mood [19]. In this project, the relationship between specific colours and specific moods needs to be addressed, so that lights display appropriate colours to reflect the current mood of the track.

It has been argued that assigning colours to moods is subjective. People can perceive things differently depending on their current mood and an individual's favourite colour could always been viewed positively [20]. Teachings and phrases such as "green with envy" also guide people's perception; in the United Kingdom we associate black with mourning, but in China they associate this with the colour white.

However, a general trend of colours associations has emerged. Commonly we associate red with energy, yellow with happiness, green with balance and blue with relaxation [21]. Despite this generalisation certain colours can represent a number variety of emotions. The colour red can be associated with anger, excitement and danger. Godlove observed this contradiction by saying that green is usually seen as a tranquil colour but continued to explain that "greens can be very irritating - solid bright green walls can cause red after-images which are very disconcerting" [22].

3.7.3 Mood Classification

The challenge in mood classification is to identify the levels of arousal and valence for a particular piece of music. For my project, the mood acts as another informative feature, which needs to be mapped to a set of colours that encompass that mood.

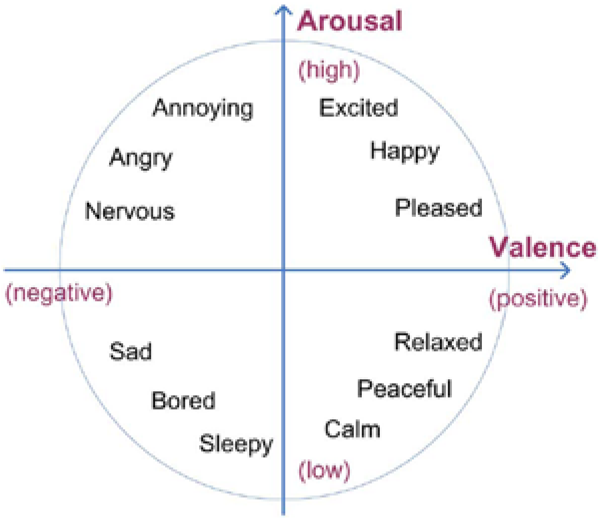


Figure 8: Thayer's valence-arousal plane for mood classification [23].

Thayer proposed a model for the description of emotions based on arousal and valence [23]. In his model he created a two dimensional mood space which consists of four quadrants. The right side of the plane represents positive emotions whilst the left side represents negative emotions. The upper side of the plane refers to energetic emotions whilst the lower side refers to silent emotions. Thayer's valence-arousal plane is shown in Figure 8. In my implementation I will map my extracted features to a similar model.

3.8 Stage Lighting

Stage lighting is the art of lighting within the context of performance art. It can be "defined as the use of light to create a sense of visibility, naturalism, composition and mood, (or atmosphere)" [24]. The four main properties of lighting are intensity, colour, direction and focus.

In this project I will concentrate on generating controls for intensity and colour properties. Intensity refers to the "strength" of a light source, and colour is self-explanatory.

4 Development Tools

4.1 LAME

Before any information could be extracted from the audio track it needed to be in a format which is widely used in a variety of software applications. The uncompressed WAV format has a simple structure and is referred to as the “lowest common denominator” when programs need to exchange sound files. All sound files are converted to the WAV format using the high quality LAME encoder.[5] This conversion means that my application is not limited to a certain input format.

4.2 Vamp Audio

The Vamp audio analysis plugin system[6] was developed at the C4DM. This audio processing system is designed for MIR algorithms to be contained in a number of plugins that can then be used to extract descriptive information from audio data. A number of these audio feature extraction plugins have also been developed at the C4DM, but a number are created offsite by independent developers using the cross-platform Vamp SDK and API.

A Vamp plugin is a binary module that can be loaded by a host application just like a normal audio effects plugin. But when fed audio data, a Vamp plugin generates valuable symbolic data instead of a transformed audio waveform. This is shown in Figure 9.

A Vamp plugin is configured once before each processing run and can accept data in both the frequency domain and the time domain. Vamp plugins also have a large amount of control over their inputs and outputs and can be non-casual - choosing to gather and store data on the whole track before any analysis and output is given.

4.3 Vamp Simple Host

Provided with the Vamp plugin SDK is a host application called *vamp-simple-host.* This host application can be run from the command line with a command of the form:

```
vamp-simple-host [-s] pluginlibrary[.dll]:plugin[:output] file.wav [-o out.txt]
```

This command loads a specific Vamp plugin and sets its required output. It also specifies the input audio file and output text file where the results are to be written. The first stage of my feature extraction implementation was to make a call to this host for each plugin that I used, and consequently to read the output into relevant data structures in my program. The [-s] flag

[5] lame.sourceforge.net

[6] vamp-plugins.org

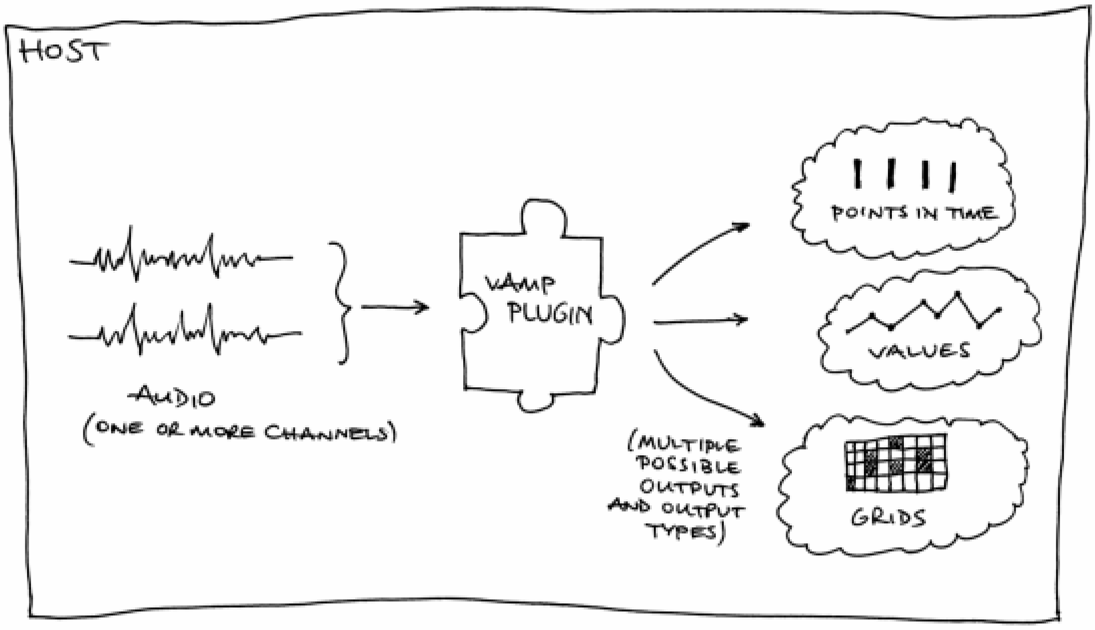


Figure 9: A simple flow chart of the Vamp Audio system.

allows me to specify whether the outputs should be labelled with the audio sample frame at which they occur or, alternatively, with the time in seconds.

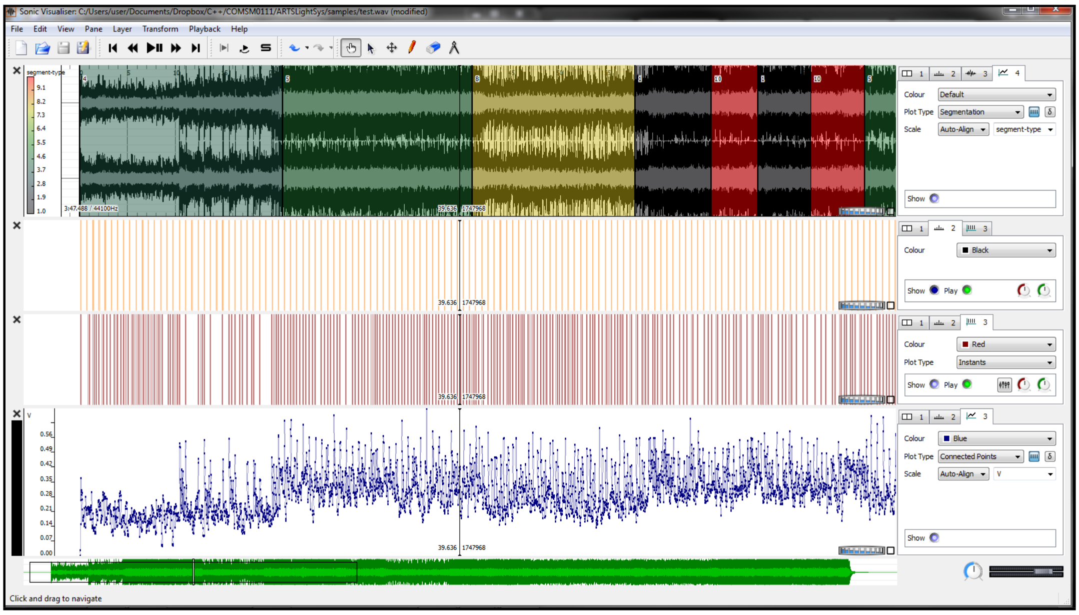
4.4 Sonic Visualiser

Gathering a relevant set of Vamp plugins was not a simple task; each plugin had to be tested to get a full understanding of their outputs. As an example, the Bar and Beat Tracker plugin developed at C4DM has four different outputs including beats, bars, beat count and beat spectral difference. Although some documentation has been made on a number of the available plugins the most beneficial way to learn about them is by visualising them. Another tool created at C4DM known as Sonic Visualiser[7] lets this happen.

Sonic Visualiser is a highly customisable playback and visualisation environment that allows the user to study the contents of a musical recording closely. The consumer friendly audio application allows the outputs of Vamp plugins to be viewed, analysed and annotated with a number of different parameters and settings. Shown below in Figure 10 is a typical view of Sonic Visualiser showing the results of a variety of plugins on a single audio track.

With this tool at my disposal it was easy to test and identify relevant Vamp plugins and set their parameters accordingly to maximise the data obtained. Without this piece of software it would have been almost impossible to verify that beat and note onset locations were correct as

[7] sonicvisualiser.org



Sonic Visualiser: C:/Users/user/Documents/Dropbox/C++/COMSM0111/ARTSLightSys/samples/test.wav (modified)

Colour: Default
Plot Type: Segmentation
Scale: Auto-Align, segment-type

Colour: Black

Colour: Red
Plot Type: Instants

Colour: Blue
Plot Type: Connected Points
Scale: Auto-Align, V

Click and drag to navigate

Figure 10: Results of four Vamp plugins shown in Sonic Visualiser. From top to bottom; segmentation, beat locations, note onset locations and amplitude

the accuracy of these timings are too precise to be measured by hand. Sonic Visualiser is well documented and a tutorial produced by the Centre for the History and Analysis of Recorded Music was a useful resource when learning to use the software [25].

4.5 Autodesk Maya

The 3D computer graphics software Maya[8] developed by Autodesk was used in my project to demonstrate and test the lighting system. The decision to use this software was made when it became clear that the lighting control system would not run need to run in real-time. A real-time alternative would have been to display the output using the 3D graphics engine Ogre.

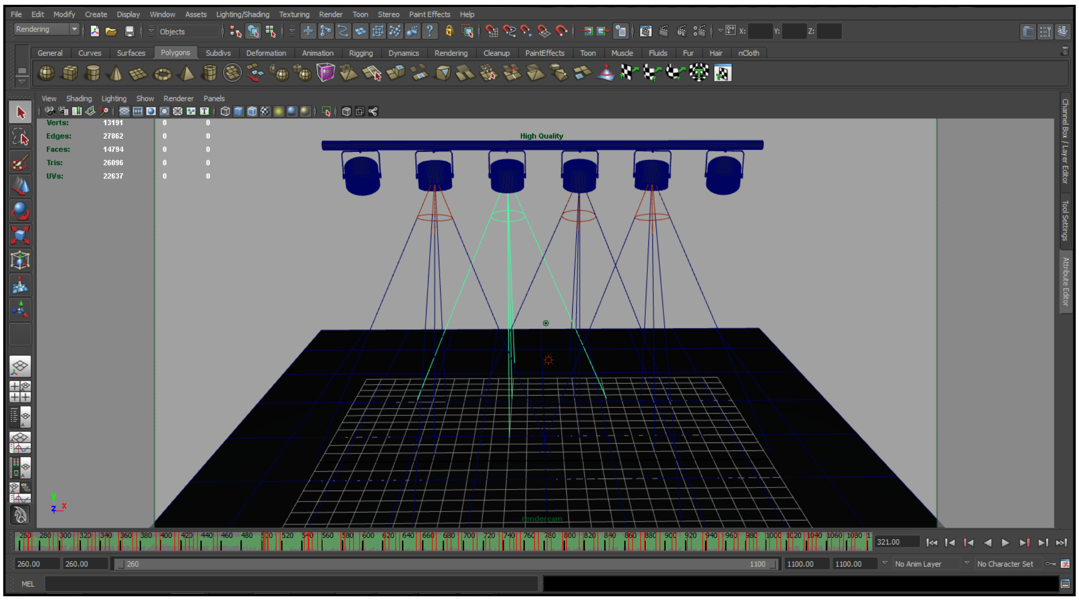
Maya was chosen as it can provide a high quality output through detailed rendering, but also because it can generate the lighting show automatically using its built in scripting language MEL (Maya Embedded Language). Figure 11 shows the testing environment of the lighting system created in Maya

4.6 Adobe Kuler

Adobe Kuler[9] is an internet application provided by Adobe Systems that lets users submit colour schemes and tag them based on their mood and other keywords. These colour schemes consist of a set of five related colours. Using colours suggested by [19, 21, 22] relevant colour schemes can

[8] usa.autodesk.com/maya

[9] kuler.adobe.com



File Edit Modify Create Display Window Assets Lighting/Shading Texturing Render Toon Stereo Paint Effects Help
Rendering
Objects
General Curves Surfaces Polygons Subdivs Deformation Animation Rigging Dynamics Rendering Cleanup PaintEffects Toon Muscle Fluids Fur Hair nCloth
View Shading Lighting Show Renderer Panels
High Quality
No Anim Layer
No Character Set
MEL

```
Verts:   13191   0   0
Edges:   27862   0   0
Faces:   14794   0   0
Tris:    26096   0   0
UVs:     22637   0   0
```

Figure 11: The 3D testing environment created in Maya.

be found on Kuler. Once a colour scheme is identified for each mood in my system, the RGB values are extracted and given to my lighting system.

4.7 Development Platform

The choice of development platform was not straightforward with this project. The choice was greatly affected by the audio libraries that I could find. After I had looked through available audio libraries it was clear that C++ would be the language needed to develop the lighting system due to its flexibility and performance. The next choice was to decide on an operating system. The two main choices available to me were Linux and Windows. Once I decided to use the Vamp Audio system, which is supported on both of the platforms in question, it became clear that a smaller selection of the plugins worked on Linux. To make sure that I had all the feature extraction techniques necessary I chose to develop in Windows.

5 Implementation

5.1 Music Information Retrieval

In order to create my lighting system, significant audio analysis techniques and libraries would have to be used to obtain all of the required information. A collection of audio libraries including libsndfile[10] and FFmpeg[11] grant access to low-level representations of the audio signal like spectral energy statistics and cepstral features. However, the majority of these audio libraries do not provide tools for extracting features from this low-level representation. Typically a number of low-level features are needed to gain a real understanding of an audio signal. For my lighting system I needed to obtain mid-level features such as note onset times and chord patterns. From these mid-level characteristics high-level features like song structure and mood can be created. These high-level features are used directly in the lighting system to produce an output that matches a given audio track as closely as possible.

After exploring a selection of available audio libraries I decided to use the Vamp audio plugin analysis system which is explained in Section 4.2. The system was written in C++ and can work on Windows, Mac and Linux. It was chosen because it contained all of the MIR techniques that I needed for my lighting system and provided a number of tools that helped me to understand the plugins and their underlying algorithms. The development is also maintained by a strong group of researchers at the C4DM. Unfortunately, the Vamp audio system does not require all of its plugins to work in real time. However, it was still chosen because no other audio system could be found with such a range of MIR techniques that Vamp provided. By making this decision the aim of integrating an external microphone that could capture environmental audio had to be dropped.

In my system the Vamp plugins are executed during the pre-processing stage for each audio track. As long as this pre-processing stage takes less time on average than the average length of an audio track then the implementation will be capable of continuous playback of a playlist of audio tracks. After testing the MIR stage on 5 audio tracks it became clear that the processing time grew in proportion to the length of the track. It was found that the pre-processing stage takes around 24% of the time of track being processed.

5.2 Onset Detection

5.2.1 Aubio Onset Detector Plugin

For onset detection, two Vamp plugins from different developers were used in combination to increase accuracy and identification. The first plugin used was the Aubio Onset Detection plugin

[10] mega-nerd.com/libsndfile

[11] ffmpeg.org

that was developed using Brossier's Aubio library.[12] Aubio is another audio library that attempts to extract features from an audio signal. The features it provides are not as extensive as those found in Vamp but they have been adapted to create a number of Vamp plugins. The onset detector estimates note onset locations of an audio track. A closer inspection of the output from this plugin using Sonic Visualiser revealed that certain 'hard' onsets were not being identified. To solve this problem I looked at combining the results from this plugin with that of another onset detection plugin.

The Aubio Onset Detector Plugin takes just over 11 seconds to execute.

5.2.2 Percussion Onset Detector Plugin

The C4DM provides two onset detection plugins. The first, is a more generalised onset detector and the second is focused on locating "hard" percussive onsets. Using the default energy rise threshold and sensitivity parameters, I found that this percussive onset detector located a number of onsets that were missing from the onset detector provided by Brossier, along with some duplicates.

Once the outputs from both plugins were input into my system, they needed to be combined and filtered to eliminate any errors or duplicates. The assumption was made that each onset should be spaced across the audio signal according to some musical theory. My first approach was to accept onsets that matched closely to fractions of the beat length. A semi-quaver is a known as a sixteenth note and each beat contains four of them. My first approach was to identify onsets that were close to the position of semi-quavers in the audio track, but I found that this resulted in a large number of false negatives. The presence of syncopated rhythms and melodies produced by tuplets discussed in Section 3.2 highlight the flaws of this first approach. Complex rhythms highlighted that it was not possible to match all onsets to any form of musical pattern without a higher level knowledge of the audio signal. Hastily removing onsets would result in a loss of accuracy of the lighting system. Instead, focus was put on extracting duplicates and identifying pairs of onsets that occurred within a very small window.

These pairs were found by locating onsets that occur less than a sixteenth note away from each other. The length of a sixteenth note varies for each track depending on the beat length. The beat length was then multiplied by a fraction of 0.85 to ensure that correctly spaced semi-quavers were not accidentally identified. The conditional statement for finding duplicates and closely spaced notes is:

$$\text{if}(currentOnset - lastOnset < (beatLength/4) \cdot 0.85)$$

[12] aubio.org

The times of the two closely lying onsets were then analysed to see which one lay closer to a sixteenth note interval. Although tuplets may be present in music, they are rarely used at this note speed and selecting an onset that is close to a beat will also not look out of place in a lighting display.

The Percussion Onset Detector Plugin takes around 3.5 seconds to execute. This reduction in time compared to the other onset detector plugin is due to the plugin only using energy-based detection methods. The Aubio Onset Detector Plugin uses both energy-based and phased-based detection methods.

5.3 Tempo & Beat Detection

5.3.1 Tempo & Beat Tracker Plugin

The tempo detection algorithm I utilised was the Tempo & Beat Tracker plugin produced at the C4DM. It analyses an input audio track and estimates the position of metrical beats within the music. These extracted points in the audio are analogous to a human listener tapping their foot to the beat. The required output of this plugin was the tempo which was returned as a feature, each time the tempo changed, as a single value in beats per minute. These tempo values were passed into my program and stored in a vector.

After obtaining the number of beats per minute, the beat length for an audio signal can be calculated as:

$$beatLength = \frac{60}{tempo}$$

The Tempo & Beat Tracker plugin takes just under 8 seconds to execute.

5.3.2 Bar & Beat Tracker Plugin

The Bar & Beat Tracker produced at the C4DM works in a similar way to the Tempo and Beat Tracker mentioned in Section 5.3.1, giving the same results for beat locations. However, this plugin also estimates the positions of bar lines, and given an input parameter of beats per bar, it can count the position of each beat in a bar. This is equivalent to a human listener counting in time to the music e.g. with four beats per bar the count would be:

1, 2, 3, 4, **1**, 2, 3, 4, **1**, 2, ...

The output that I required from this plugin was the beat locations in an audio track and a label for each beat representing its beat count. Being able to differentiate between these types of beat allows me to create strong lighting effects for certain beats.

Having acquired the tempo and beat locations from the selected Vamp plugins the rhythm of a musical piece can be observed to extract more valuable information. These rhythm-related features include monitoring the time signature stability and the tempo confidence over the whole song. But the rhythmic features of most importance are the beat variance and beat confidence both of which can be obtained from the data we possess.

Beat variance is used to monitor the regularity of rhythm in a piece of music, a low beat variance equates to a regular rhythm. This is calculated by observing the set of time intervals between beat locations, and calculating the variance of this set. If X is the set of size n which contains the beat interval times the beat variance is calculated using the following formula:

$$beatVariance = \frac{\sum\limits_{i=1}^{n} (X_i - \bar{X})^2}{n}$$

Beat confidence is measured by inspecting the amplitude levels at the beat locations, and ensuring that it is above a given threshold.

The Bar & Beat Tracker plugin takes just under 8 seconds to execute.

5.4 Key Detection

5.4.1 NNLS Chordino Plugin

The first attempt at key detection was using a chord detection plugin developed by Muach at the C4DM [26]. Knowing the prominent chords of a piece of music allows you to determine the key.

The Non-Negative Least-Squares (NNLS) Chordino Plugin outputs estimated chords times and labels from an input audio signal. The chord labels include all major, minor, augmented and diminished chords. Using the data from the plugin, I recorded the proportion of time that was spent in each chord, and from these times, selected the three most commonly occurring chords in a song. These “top” chords were then matched against of table of commonly used chords for each key to gain a final estimate for the key of the track.

5.4.2 Key Detector Plugin

In addition to my approach to key detection, I also identified a plugin provided by the C4DM called Key Detector. The plugin analyses an audio track and estimates key changes throughout an audio track. It then outputs the time of these estimated key changes along with the detected key.

The Key Detector plugin is capable of detecting all 24 different keys (12 major, 12 minor) but the only real important data that needed to be extracted for my system was whether the predominant key was major or minor. To do this, the sequence of keys from the audio input was split into two lists, one or major keys and one for minor keys, the time spent in each of these lists was totalled and then compared against the other column, to acquire the mode of the track.

5.5 Amplitude Analysis

5.5.1 Amplitude Follower Plugin

The amplitude of an audio signal was another feature that I needed to monitor to allow me to interpret the volume of the song and how this changes over time. The amplitude follower plugin was created by Dan Stowell and is a simple implementation of the amplitude-follower algorithm found in audio programming environment, SuperCollider[13]. The proposed algorithm tracks the amplitude of an audio signal sample by sample and returns peak values block by block. The output amplitude peaks are returned in volts rather than in decibels. This is because a decibel is a dimensionless unit which can be viewed as relative gain or loss. A volt is an absolute measure which can be compared between different samples over a number of different tracks.

5.6 Segmentation

5.6.1 Segmenter Plugin

The final plugin which I integrated into my system was tasked with segmenting the audio track into a number of structurally consistent segments. This was done by a Vamp plugin produced at C4DM called Segmenter. The plugin identifies structurally different sections of an audio track, but does not attempt to label them - it does not try to work out which section is a chorus and which is a verse. Instead it gives each section an identifier and guarantees that segments with the same identifier will contain related features and have a similar structure. This is a beneficial property as during the feature modelling process it is not important to know if a section is a verse or chorus, but it is important that repetitions of the chorus in a musical track have the same identifier. The output from the plugin is the estimated time of the segment boundaries and a numerical value representing the segment type.

[13] supercollider.sourceforge.net

5.7 Light Creation

Having gathered a large amount of information from a selection of advanced MIR techniques, the next step is to manipulate these features into a lighting control system. This process can be divided into three important steps:

- Mood classification of structurally consistent segments.
- Identification of lighting cues.
- Design of the lighting display.

5.7.1 Mood Classification

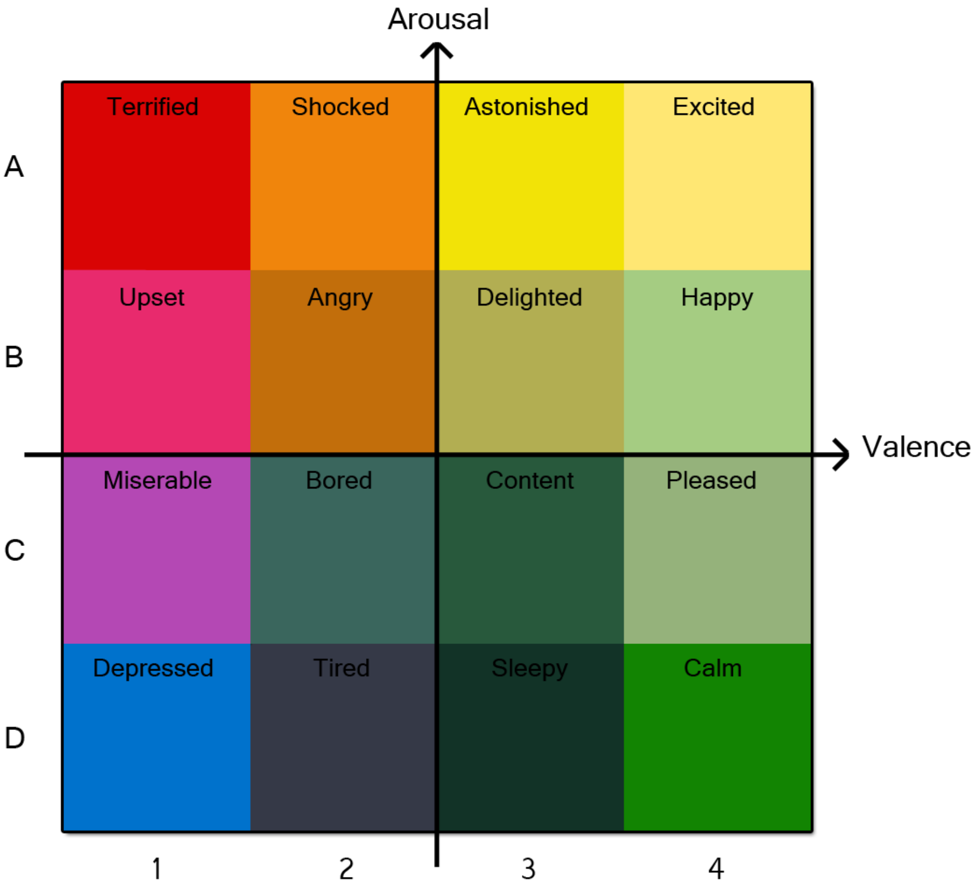
With a selection of features in my possession it was now time to conclude which of them would be relevant for mood classification. The collection of features needed to be analysed to identify which of them conveyed emotion. Mood can be affected by two factors, arousal and valence, which are defined in Section 3.7.

The most informative extracted feature for mood classification is the mode of the audio track, which refers to the key being either major or minor. During the mood classification process, the key of each segment is detected as well as the key of the entire track. This means that when calculating the valence of each section, the mode of that section is the main influence, but the underlying mode of the entire track can also be considered.

The next feature to be used in mood classification is the tempo. Tempo has a strong effect on the arousal levels of a song. The tempo of a track is usually constant and is unlikely to change between identified segments. Nevertheless the system is capable of identifying tempo changes.

The final feature that will be incorporated into the classification system is the amplitude of the track. Before any segmentation of the audio track, the average amplitude value is calculated for the whole track. Then for each section, the average amplitude is calculated and compared to the amplitude value of the whole track. Using this method, louder and softer segments of the audio signal can be found. A change in amplitude will affect the arousal of the mood in a similar way to the tempo. Loud amplitude will be associated with a stimulating piece of music, whereas softer amplitude will relate to a less arousing piece of music.

The design for mood classification was to use these emotive features and map them onto a two dimensional space similar to that proposed by Thayer [23]. This process will be repeated on each section produced by the Segmenter plugin. As stated by Thayer, a valence-arousal plane



	1	2	3	4
A	Terrified	Shocked	Astonished	Excited
B	Upset	Angry	Delighted	Happy
C	Miserable	Bored	Content	Pleased
D	Depressed	Tired	Sleepy	Calm

Figure 12: Classification grid of mood and colour based on valence and arousal.

can be used to classify a wide variety of moods. I decided to fix a 4x4 grid on top of this proposed feature space. For each grid coordinate I assigned a mood and a set of colours collected from Adobe Kuler. The proposed classification grid is shown in Figure 12. Grid coordinate A2 for example, lies in the high arousal, low valence quadrant of the classification space. Its specific mood at this point is classified as "shocked" and contains shades of reds and yellows.

To map our set of features to this plane, the level of valence and arousal for an audio segment had to be calculated. In my system, valence is affected by the key of the current section and the key of the entire track. From my research it was clear that a minor key evokes a low valence mood, so a minor section of minor song was chosen to relate to the lowest valence level. In contrast a major section belonging to a song in major key was chosen to produce the highest level of valence. The final decision was to decide whether the key of a specific section, or the key of the whole song should be given a higher precedence. As my mood classification focuses on segmentation, I decided that priority should be given to the mood of the current section. A simple decision tree shown in Figure 13 was created to show the mapping of key values to valence

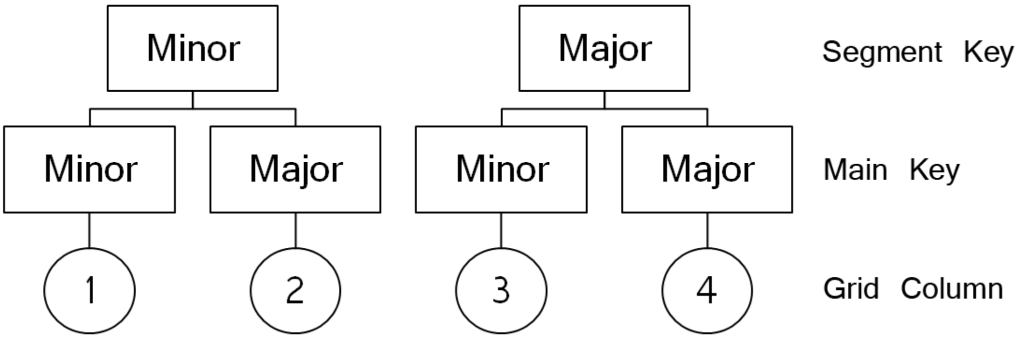


Figure 13: Decision tree for valence grid column.

grid columns of my mood classification grid.

Once the valence of the mood had been calculated I moved on to identifying arousal levels for each segment. The two factors that I identified to have an effect on arousal were tempo and amplitude. After acquiring a list of classical tempo markings[14] it was clear that roughly any tempo greater than 100 BPM could be considered fast and any tempo below 100 BPM could be considered slow.

In my system it was decided that the average tempo was regarded as *Andante*, which means at walking pace, with tempo values ranging from 76-108bpm. Any tempo in this range would be considered to sit in the middle of arousal plane for mood classification. My system then created a range of tempo values around this central point, based on the classical tempo markings. It is expected that almost all pieces of music have a tempo within the range of 60-200 BPM. Any tempo less than 60 BPM, known as *Largo*, is considered extremely slow and marks the bottom end of the range. A tempo of above 200 BPM is described as *Prestissimo*, or extremely fast, and marks the top end of the tempo range modelled in my mood classification system.

The importance of tempo was calculated using the beat consistency value for each section. If beats were distinct and consistent in each section then the arousal value would be left as it is. Conversely, if the presence of beats was more discrete the arousal value would be halved, meaning tempo would have less of an impact on the mood.

Amplitude was also involved in calculating the final arousal level of a musical piece. By comparing the amplitude level of a section against the overall amplitude, contrasting sections of the music could be highlighted. To compute an arousal level from the amplitude information, the amplitude of the entire piece was taken and multiplied by a number of coefficients. This resulted in a number of amplitude intervals in which amplitude levels of segments could be placed. One interval was

[14] dolmetsch.com/musictheory5.htm

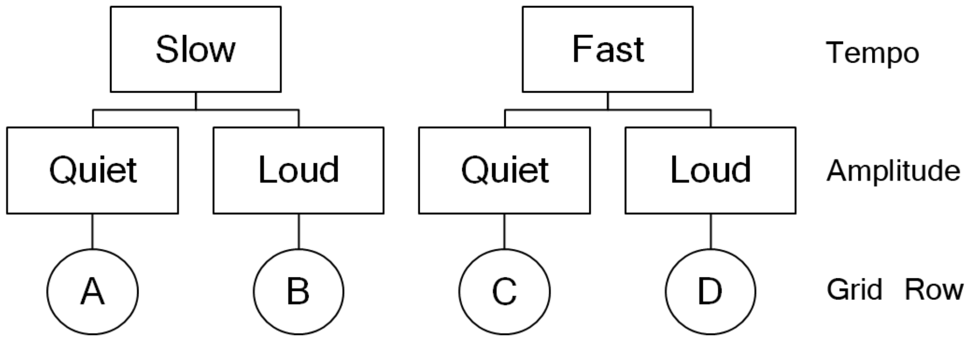


Figure 14: Decision tree for arousal grid row.

within a 10% increase of the main amplitude and another was within a 10% decrease of the main amplitude. The last two intervals were for any large increase or decrease above or below the 10% intervals. For each segment the amplitude was placed into one of the intervals which correspond to different arousal outputs. A simple decision tree shown in Figure 17 summarises how the arousal levels produced by tempo and amplitude features were mapped to rows of my mood classification grid.

Once the arousal and valence levels are calculated, a specific grid point can be identified and a mood can be given to each segment of the audio input.

5.7.2 Identification of Cues

With the mood classification step complete, the next step was to identify appropriate times in the music where lights should be triggered, known as cues. In my system I looked to identify cues produced by note onsets and beat locations. Both of these features were provided by Vamp plugins and processed by my system as described in Sections 5.2 and 5.3 respectively.

The note onsets generated were accurate and ready to be used in the output lighting display. However, the beat locations needed to be manipulated to calculate the importance of each beat. To ensure continuity of beat locations over an entire musical track beat locations were calculated even over silent sections of the audio signal. To check if beats were present at certain points in time, specific amplitude values of the audio track had to be queried. To simplify this process the amplitude values produced by the Vamp plugin were stored in bins with 0.1 second intervals. An average of each bin was taken and then these values were stored in a hash table, with the time as the key and the mean amplitude as the value. This meant that beats could be rounded to the nearest 0.1 second interval and the amplitude could then be quickly obtained from the amplitude hash table.

Beats were considered to be present if their amplitude value was at least a quarter of the average amplitude of the input audio track. This made sure that both strong on-beats and weaker off-beats were still accounted for, but made sure beat cues were not present during silent sections of audio.

5.8 Lighting Design

The final stage of my implementation was to output suitable information from my system so that it could be translated into a lighting display by Maya. I decided to design a simple file format that would contain all the information necessary for a lighting display. Automatic Lighting Control (.alc) files were produced by my system which consist of a chronological list of instructions for each light in the display. Each line of the .alc file represents a light event which consists of the following information:

- The light cue time, in seconds.
- The duration of the light event, in seconds.
- The intensity of the light event, a double between 0 and 1.
- The colour values of the light, represent as RGB values between 0 and 255.

These values were separated with a ":". An example line of an .alc file would be:

19.5789:0.638549:0.75:255:112:32:

Without being told the structure, it is hard to understand what each number represents, but using this simple format means the data can be quickly tokenised in Maya using simple MEL script commands.

Once a basic scene of a typical music venue had been modelled and textured, a script was made to create and extract the light information from a collection of .alc files. Each file was read by the MEL script, and the information was tokenised for each light event. The timings were converted to frame numbers based on the default Maya frame rate of 24. The duration of the light event, its intensity, and colour values could then be key framed.

Having a strong knowledge of Maya, I decided to take advantage of some of animation features of the graph editor. The graph editor in Maya is an animation tool that allows animated elements to be represented by curves. These curves represent the change in key framed values over time, a view of the graph editor is shown in Figure 15.

The important components of these curves that can be modified are the tangents. By default,

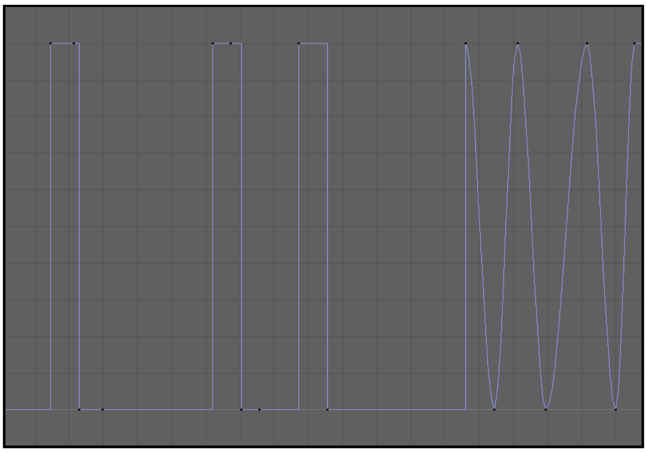


Figure 15: A range of tangents shown in Maya's graph editor. Stepped (left), Spline (right)

any key frames set in Maya will use a spline tangent, resulting in a differentiable function with smooth transitions between animated key frames. This is a suitable curve to use if the mood of the lighting is calm and relaxing, however is unfitting if the arousal level of the mood is high. Depending on the results from the mood classification algorithm, the tangent of individual keys can be changed. For high arousal moods I decided to use stepped tangents. This meant that instead of a value gradually changing from A to B over a time period, the value stays at A and then moves straight to B at its set time. This instant impact is a great addition to the lighting display and could be replicated in other output forms.

Not a lot of time was allocated to create a complex output lighting rig, but to show the capabilities of the lighting control system the output of my implementation was split over four lights. Two were used to display beat patterns, two were used to display note onsets, and all four were affected by mood changes. The note onsets were split between the two lights as well as the beat patterns. Down-beats were highlighted by being present in both beat lights. Although rather simple, I feel that this setup sufficiently displays the features of the system that I had created. Time could have been spent developing clever cues and lighting assignments but not a lot of scientific research could be found in the area, so focus was directed to other parts of the project.

5.9 Testing & Results

5.9.1 Mood Classification

Testing mood classification results is not an easy task because mood can be subjective; papers have even been written attempting to create a ground truth for automatic music mood classification [27]. For my ground truth, I used a method similar to [28]; using the social tags from the music recommendation site last.fm.[15] As of 2009, the social network has over 30 million users in over

[15] last.fm

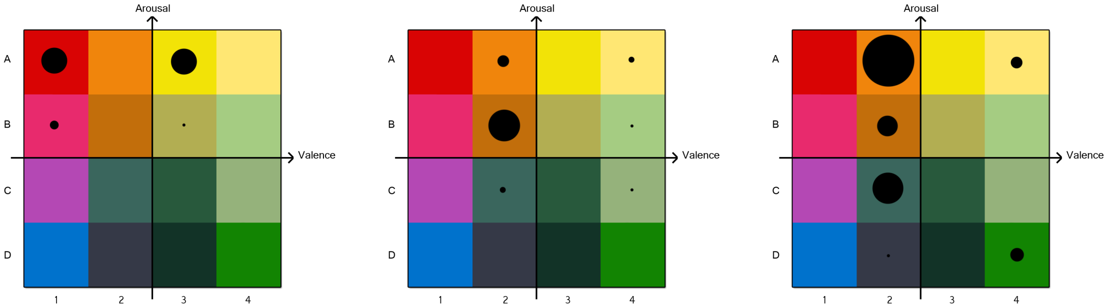


Figure 16: Mood classification results of “Woo Boost”, “Hysteria” and “Walking After You”, from left to right.

200 countries. These users are able to tag any relevant words to a musical track, some of which are mood or emotion descriptors.

During the testing phase of my mood classification, I used three tracks from my music library. The first track, “Woo Boost” by Rusko, is a fast paced, electronic dubstep track. “Hysteria” by Muse, is an atmospheric progressive rock piece and the final track, “Walking After You” by the Foo Fighters, is a slow, rising acoustic piece. The results from my mood classification algorithm are shown in Figure 16.

The results show the number of sections that belong in each coordinate of the mood classification grid. The larger the circle, the more time was spent in that particular mood.

After looking through the database on last.fm, some tags proved more useful than others. As these tags are user generated, there can be some inconsistencies between the usefulness and number of tags depending on the popularity of the song. The following mood relevant tags were found on last.fm concerning the three tested songs:

> “Woo Boost” - Massive, heavy, extreme, dark.
> “Hysteria” - Dramatic, intense, upbeat.
> “Walking After You” - Moody, mellow, sad, yellow.

Overall, these results are extremely positive. By classifying structurally different sections of the audio I did not expect them all to classify to the same mood. This would have disproved my assumption that there is musical variation within an audio track. However, I did hope to find that the majority of the results from the different segments were located in a certain region (e.g. high arousal, low valence).

My mood classification algorithm identified sections of “Woo Boost” to reflect a terrified and astonished mood. I feel that these correlate very well to the social tags “extreme” and “dark” found on last.fm. “Hysteria” was identified mainly as an angry song, with some happy sections - again translating well to the “intense” and “dramatic” tags. Most importantly, almost all of the mood classification results for this track had a positive arousal value. This matched perfectly with the “upbeat” tag found on last.fm. The final track, “Walking After You”, was classified as mainly shocked, bored and calm. The last two tags match those produced from last.fm but the classification of the shocked emotion was not what I expected. This was caused by a large increase in relative volume at the end of the track, compared to the first section which is extremely quiet and mellow. I feel this problem could be solved by looking at amplitude values relative to other tracks. However, audio tracks never tend to be the same volume, so some normalisation would have to be done.

5.9.2 Lighting Test

Once the system had been completed a test was constructed to see if people believed that the lighting show was synchronised to the audio input. From this test I hoped to learn if the lighting system was a success and also to understand the importance of audio and visual synchronicity based on people’s perceptions.

The test was constructed using an 840 frame (35 second) extract from the Muse track “Hysteria”. I created a video of my lighting display to match the audio extract. At the start of the video the lights were random, not synchronised. I then slowly increased the synchronicity of the lights to a selected point in time during the video where my system believed the lights to be completely synchronised. This selected point was at the frame 580 (∼24 seconds) and from this point to the end of the test the lights were 100% synchronised. To create this slowly synchronised process I added a level of random noise to the timing of my lighting cues which was reduced over time. I chose to make the synchronisation process gradual rather than instant to try and get an idea on how accurate lighting has to be to the music. If the video was totally random at one point, then instantly was in sync, almost everyone would notice such an obvious transition.

30 test subjects were presented with the test video along with the description:

> At some point in the video the lights will become synchronised with the music. When this happens, pause the video and take a look at the frame number in the top left corner. If you could then post that frame number in the comments section below it would be much appreciated. If you don’t think that the lights become synchronised at all answer with 0.

The results from these tests are represented in a histogram shown in Figure 17. The ground truth point of synchronisation is represented by the red line and the mean value from the tests is

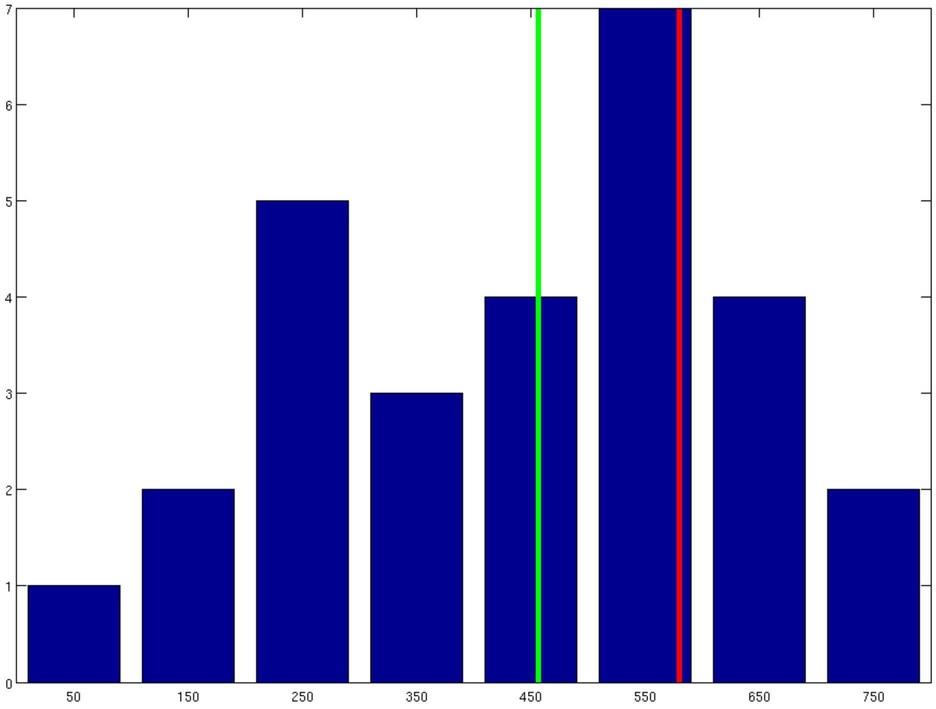


Figure 17: Histogram showing the results of the lighting test.

represented by the green line. As expected, the results form a skewed normal distribution based around a mean value lower than the ground truth. This was caused by people believing the song had synchronised at an earlier point in time than the ground truth which was due to the nature of the test gradually getting more and more in sync. Furthermore, it was unlikely to see any significant number of results after the point of synchronisation.

Some of the interesting results to talk about are the ones located in the 200-300 frame bin. It is clear from the histogram that the number of samples that fall into this bin is larger than expected. During this time in the lighting display, there is a mood shift as the song switches from the intro section to the chorus. The mood shift changed the colours of the lights which might have lead some people to believe that synchronisation had occurred because the lights were showing some relationship to the pattern of the music. Another mood shift occurred within the 700-800 frame bin which might explain the larger than expected number of results after the point of synchronisation.

The only real negative feedback from these tests was a single test result of 0, meaning that the individual didn't believe the lights were ever synchronised with the audio. I repeated the test and asked them if at some point the lights stopped being random and were related to the music. From this test I got a result of 508, which was a more positive result.

6 Conclusion

Considering the aims that were set at the start of this project, it is clear that the majority were completed successfully. The first objective was to collect a strong and relevant set of advanced MIR techniques. After significant analysis and research into available audio libraries, the Vamp audio plugin system was located and identified as the strongest collection of MIR algorithms. Many other audio libraries were arduously tested, but I found that the Vamp system produced reliable, high quality symbolic data about each input audio track and also provided supportive development tools. However, a few high-level feature extraction techniques, such as instrument and vocal extraction were not available. The features were not integral to the design of the lighting system but could have had some influence on the final output.

One available feature that was not integrated into my system that would have proved useful was modelling the progression of melodies. This feature could have been extracted using the Aubio Note Detector plugin, which outputs notes and their positions in the audio signal. Currently in my system, the only feature that influenced the valence of the musical track was the key. A melody is a succession of musical notes which consists of pitch and rhythm information. It has been shown that melodies can effect both mood and arousal levels [29].

The next aim was to use these MIR techniques to create a lighting control system that was synchronous to the provided user input. By showing the accuracy of the MIR algorithms that were used, I can guarantee that none of the precision was lost when I created my lighting system. On all the inputs that I tested, the lights were synchronous to the extracted beats and the notes. If a problem with accuracy occurred, then it would be out of my control to fix it, and I would have had to find new MIR algorithms. However, by ensuring that the Vamp plugin system worked well across a number of inputs, by using Sonic Visualiser, I am positive that these MIR techniques have been successfully applied to create a synchronous lighting control system.

To increase the realism and immersive nature of the lighting display, I decided to create a mood classification algorithm directed at classifying segments of the audio signal. The difficulties in this algorithm were understanding how to map audio features to a selection of moods. By using Thayer's model of moods based on arousal and valence, the process of mapping the audio features became clearer. I believe the motivation behind my segmented approach to mood classification was valid and that the results have proved its success.

I believe that the mood classification algorithm could be further improved by looking into machine learning; running my classifier on a training set could have improved the results. However, obtaining a ground truth for mood classification of the training set would not have been easy or necessarily correct.

One goal that could not be completed in this project was the integration of an external microphone to the system. This is was due to the Vamp plugin system not enforcing a real-time constraint on its plugins. I feel that this additional feature would have definitely increased the performance of the display and it would have been an interesting challenge to understand crowd noise and reflect it accurately in the lighting display. I believe the capture and analysis of environmental noise would prove to be an exciting piece of future research in this area.

Looking back at this project, I believe that other lighting properties could have been controlled to improve the quality of the lighting display, using the features that are already made available by Vamp. The light outputs from this project focus on intensity and colour properties, but in the real world, automated lights are capable of so much more. The other property that could have been considered was movement, which involves tilt and pan; it would have been fascinating to see how mood affects these movements.

The focus of this project was on gathering audio features and proving that they could be used to create an automated lighting display. I wanted to ensure that the focus was on this core of the system, ensuring that the feature extraction and mapping worked as it should. Because of this, not much attention was given to the creation of complex lighting effects. I felt that much time could be spent creating entertaining light combinations using a large number of light fixtures. However, I believe that this would be the job of a lighting expert.

This project has produced a fully automated lighting system that is in time with the music and responsive to changes in the audio signal. To develop this technology into a high quality software product, I believe that the knowledge of a lighting professional should be incorporated, in order to create and store advanced lighting combinations. The information I gathered from the MIR techniques could not only map to a classified mood, but also to a subset of these combinations, improving the performance of the lighting system with the knowledge of an industry expert.

References

[1] E. Sohlberg. *Light-Projecting Apparatus*, US Patent: us 819611.

[2] R. Cadena. *Automated Lighting*, Burlington: Elsevier, 2006.

[3] J. Fourier. *Thorie Analytique de la Chaleur (The Analytical Theory of Heat)* 1822.

[4] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler. *A Tutorial on Onset Detection in Music Signals*, IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035-1047, 2005.

[5] N. Collins. *A Comparison of Sound Onset Detection Algorithms with Emphasis on Psycho Acoustically Motivated Detection Functions*, in 118th Convention of the Audio Engineering Society, 2005.

[6] C. Duxbury, J. P. Bello, M. Davies and M. Sandler. *Complex Domain Onset Detection function for Musical Signals*, in Proceedings of the 6th Conference on Digital Audio Effects, 2003.

[7] J. P. Bello and M. Sandler. *Phase-Based Note Onset Detection for Music Signals*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2003.

[8] M. E. P. Davies and M. D. Plumbley. *Context-Dependent Beat Tracking of Musical Audio*, in IEEE Transactions on Audio, Speech and Language Processing. vol. 15, no. 3, pp. 1009-1020, 2007.

[9] M. E. P. Davies and M. D. Plumbley. *Beat Tracking With A Two State Model*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. 241-244, 2000.

[10] K. Noland and M. Sandler. *Influences of Signal Processing, Tone Profiles, and Chord Progressions on a Model for Estimating the Musical Key from Audio*, Computer Music Journal, vol. 33, no. 1, Spring 2009.

[11] Encyclopaedia Britannica. *Markov process*, Encyclopaedia Britannica Online, Retrieved May 2012, from http://www.britannica.com/EBchecked/topic/365797/Markov-process

[12] M. Levy and M. Sandler. *Structural Segmentation of Musical Audio by Constrained Clustering*, IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 2, pp.318-326, February 2008.

[13] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*, Oxford: Oxford University Press, 1990.

[14] M. Levy, K. Noland and M. Sandler. *A Comparison of Timbral and Harmonic Music Segmentation Algorithms*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1433-1436, 2007.

[15] P. R. Kleinginna and A. M. Kleinginna. *A Categorized List of Emotion Definitions, with Suggestions for a Consensual Definition*, Motivation and Emotion, vol. 5, no. 4, pp. 345379, 1981.

[16] C. Schubart. *Ideen zu einer sthetik der Tonkunst (Ideas on an Aesthetic of Sound Art)*, 1806.

[17] G. Husain, W. F. Thompson and E. G Schellenberg. *Effects of Musical Tempo and Mode on Arousal, Mood, and Spatial Abilities*, Music Perception, vol. 20, no. 2, pp.151171, 2002.

[18] A. Bhatara, A. K. Tirovolas, L. M. Duan, B. Levy and D. J. Levitin. *Perception of Emotional Expression in Musical Performance*, J. Exp. Psychol. Hum. Percept Peform. vol. 37, no. 3, pp. 921-934, 2011.

[19] A. Albert. *Color Hue and Mood: The Effect of Variation of Red Hues on Positive and Negative Mood States*, Journal of the Behavioral Sciences, 1, 2007.

[20] H. R. Carruthers, J. Morris, N. Tarrier and P. J. Whorwell. *The Manchester Color Wheel: Development of a Novel Way of Identifying Color Choice and its Validation in Healthy, Anxious and Depressed Individuals*, BMC: Med Res, Methodol, 2010.

[21] S. Chiazzari. *The Complete Book of Color*, Boston: Element, 1998.

[22] I. H. Godlove and E. R. Laughlin. *The Psychology of Color*, 1940.

[23] R. E. Thayer. *The Biopsychology of Mood and Arousal*, Oxford University Press, 1989.

[24] S. R. Russell. *A Syllabus of Stage Lighting*, New Haven: Drama Book Specialists, 1964.

[25] N. Cook and D. Leech-Wilkinson. *A Musicologist's Guide to Sonic Visualiser*, 2009.

[26] M. Mauch and S. Dixon. *Approximate Note Transcription for the Improved Identification of Difficult Chords*, in Proceedings of the 11th International Society for Music Information Retrieval Conference, pp.135-140, 2010.

[27] J. Skowronek, M. F. McKinney, S. van de Par. *Ground Truth for Automatic Music Mood Classification* in Proceedings of the 7th International Society for Music Information Retrieval Conference, 2006.

[28] C. Laurier, M. Sordo, J. Serrà and P. Herrera. *Music Mood Representations from Social Tags*, in Proceedings of the 10th International Society for Music Information Retrieval Conference, 2009.

[29] E. G. Schellenberg, A. M. Krysciak and R. J. Campbell. *Perceiving Emotion in Melody: Interactive Effects of Pitch and Rhythm*, Music Perception, vol. 18, no. 2, pp. 155-171, 2000.

[30] S. Pauws. *Musical Key Extraction from Audio*, in Proceedings of the 5th International Society for Music Information Retrieval Conference, pp. 96-99, 2004.

[31] Y. Yang, C. Liu and H. Chen. *Music Emotion Classification: A Fuzzy Approach*, in Proceedings of the 14th Annual ACM International Conference on Multimedia, pp. 81-84, 2007.

[32] A. Gabrielsson and E. Lindström. Music and Emotion: Theory and Research, chapter *The Influence of Musical Structure on Emotional Expression*, pages 223-248. Oxford University Press, 2001.