A Psychoacoustically Motivated Technique for the Automatic Transcription of Chords from Musical Audio

Daniel Throssell

School of Electrical, Electronic & Computer Engineering
The University of Western Australia

18th October 2013

ABSTRACT

All music, and especially pop music, is based upon ‘chords’: structured combinations of musical notes that harmonise when sounded together. However, because of the presence of inharmonic spectral peaks known as ‘upper partials’ within most musical sounds, current techniques for automatically transcribing chords from musical audio suffer from varying degrees of inaccuracy as these upper partials obscure the actual perceived sound during computerised analysis. Since different musical instruments exhibit different upper partial signatures, blending multiple elements in a mix worsens this effect due to their interference with one another.

In this thesis a psychoacoustically motivated technique for processing audio is simulated to evaluate its effectiveness upon chord transcription. The human auditory system is imitated by taking four ‘streams’ from songs – individual tracks corresponding to the bass, vocals, drums and other instrumentation – and mixing these in various proportions to determine whether, by reducing the interference of these streams with one another, better chord recognition performance can be achieved with a subset of them.

A total of 434 audio files corresponding to 62 individual chords are analysed using an algorithmic technique for automatic chord estimation. It is demonstrated that the best chord recognition performance over the sample set is achieved by partial removal of the drums, vocals and bass, whilst leaving other instrumentation at full signal level. This achieves a significant 40.33% increase in chord transcription accuracy compared to the original unseparated chord samples, showing that it is theoretically possible to improve chord recognition performance by separating audio streams within a song. This result assists the development of more accurate chord recognition techniques in fields such as music information retrieval and also provides some insight into the principles behind human music perception.

ACKNOWLEDGEMENTS

'Tis with great joy and thankfulness
I now consign this year of stress
And study, though it seemed an age,
To but mere ink on history's page.
But as these final words I write
A sobering thought springs forth to light:
This goodly work came not to be
By strength of mine exclusively.
Yea, though I laboured eve and morn
By others I was hither borne
And so I aim my thanks to share
In iambic tetrameter.

To AJ, Chris and Jeremy
Companions fine through my degree;
And friends from Engineering's ranks
To whom I also must give thanks.
To Hayley, both distraction and
A well-intentioned helping hand;
To brothers who know not a whit
How they have helped me every bit.

To mother and to father both
To whom I owe these years of growth
My gratefulness is overdue:
I'm where I am because of you.
*To those who I have not here named**
By brevity I am constrained
To from these lines your name exclude:
Please know you have my gratitude.

But ere my pen is rendered still
I know that it befits me ill
To thank such others, but to not
Give thanks to Him to whom I ought.
For one great truth reflection brings:
Through Christ alone I do all things
And so to Him may glory be
For He it is that succours me.

And now with rhyme I may dispense,
For done are my acknowledgements;
And so, relieved, I finally say
It's done –
Now let the music play!

D.J.T. 2013

*One exclusion I cannot make is that of Western Power, to whom I am truly grateful for providing me with the Western Power Scholarship upon which I have completed this work.

Contents

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

This thesis is ultimately concerned with the analysis of recorded music by computers to automatically estimate its chord progression. In this chapter we introduce to the reader the context for this project. Section 1.1 begins with a general overview of the thesis; we then discuss our motivation and aims in Sections 1.2 and 1.3 respectively. Finally, we conclude in Section 1.4 with an explanation of the structure of this dissertation.

1.1 Overview

Chord transcription – the process of representing a musical piece in symbolic or written form – is a centuries-old art, being a fundamental component of music: it is the primary means for the transmission of classical music, and is an invaluable tool for musicians today to learn to play popular music. However, the performance of the task of automatic chord transcription by computers instead of people is a relatively new problem, as only recently have computers advanced to a stage at which they are able to handle the computational complexity required to analyse music in real time. Despite recent advances in the field, even the current state of the art is inferior to the results achieved by trained human transcribers. That this is the case is self-evident: the 'ground truth' chord data used to evaluate most automatic transcription methods is obtained from manual transcription itself. Clearly, human transcription is still the 'gold standard' in the field of chord recognition from music.

This suggests that to achieve optimal accuracy, algorithms for estimating chords from songs should attempt to imitate the approach taken by the human brain to the recognition and analysis of chord information in music. Consequently, although this project is an engineering one, in it we consider a model derived from psychoacoustics, the study of human music perception. After all, the human brain is the reference point by which we must discuss music perception: it makes no sense to consider it from a mechanical point of view alone, as computers cannot 'perceive' music, and an orchestral composition is no more pleasant to a computer than a dissonant mix of clashing tones.

We intend, then, to use a technique inspired by a simple model of human audition in order to improve the performance of automatic chord estimation algorithms. Specifically, we focus on the capability of human listeners to individually recognise and process separate audio 'streams' in a mix of signals (a concept we explore further in Section 2.3.1). Whilst this technique has been proposed for use in other fields, it has not yet been applied to automatic chord transcription, and so this thesis will investigate its effectiveness at improving chord recognition performance.

1.2 Motivation

Before stating our research goals, it is useful to discuss our motivations in undertaking this project, and to briefly outline the significance of this study. Our motivation for considering automatic chord extraction is twofold. Primarily, automatic transcription could assist musicians in more easily obtaining transcriptions of songs in order to play along on their instrument. Whilst the trained human brain is excellent at the task of music transcription, it is not a universally natural skill, and requires sometimes years of practice. An automatic method would provide the capability for musicians not competent at transcription to access symbolic representations of music they wish to learn. Furthermore, transcription is a somewhat laborious process even for those trained in the art, and an automatic method would greatly reduce the effort required to obtain chord transcriptions. This is the primary motivation for this project, and as such, its scope will be confined to popular (or 'pop') music, for which the requirement for automatic transcription is most prominent due to the strong basis of this kind of music upon chord progressions [1].

This is clearly a practical motivation. We have also a theoretical motivation for the project by framing it as a research problem for the field of artificial intelligence: can machine-based algorithms match the auditory processing capabilities of the human brain? Any such algorithms that achieve accuracy comparable to that of humans might be better models of how humans perceive music, and techniques that successfully improve the performance of chord estimation algorithms could provide insight into the human audition process. Since the technique proposed in this project involves assuming a model of human perception, proving the validity or falsity of our primary hypothesis may also assist in verifying our assumptions about human perception.

Because our main motivation relates to the analysis of popular songs, we shall use 'audio', 'song' and 'music' interchangeably throughout this thesis except where indicated otherwise, as our work is most relevant to those forms of musical audio that are identifiable as songs (as opposed to, for example, symphonies). Likewise, although the process of estimating chords from musical audio can be best termed 'automatic chord estimation', we employ the terms 'extraction' and 'transcription' as well: it can be argued that all music *transcription* (that is, the process of notating the chord information of musical audio in a written form) is a process of *extracting* such information from the music, and comprises the transcriber's best *estimate* of the chords contained therein. Thus these terms will be used to denote the same process henceforth.

1.3 Aims

Considering our primary motivation, then, we may formulate our research aims. In a general sense, of course, the goal of this thesis is to contribute to the field of automatic chord transcription by improving current techniques for estimating chords from audio. More specifically, we aim to achieve this by assessing whether using a model of human audio perception (i.e. separating instruments from the audio signal before analysis) to analyse musical audio improves the performance of chord estimation systems.

Given the supremacy of the human brain over any computational method in the area of chord transcription, we hypothesise that an algorithm able to segment and classify a song's spectro-temporal content by audio 'streams', analogously to the brain – that is, an algorithm capable of identifying and separately analysing the various components (such as different musical instruments) of the song – should outperform one which seeks to analyse that same song as an undivided whole. To this end, this project considers the conceptual development of the theory behind such an algorithm, based on currently known techniques for extracting audio 'stream' information from songs, and attempts to evaluate the effectiveness of this proposed technique at improving chord estimation accuracy.

1.4 Structure of the Dissertation

In order to achieve our ultimate goal of evaluating the effectiveness of a proposed method of improving automatic chord estimation, we begin by establishing the state of the art in a literature survey. This is covered in Section 2, along with an overview of some basic music theory concepts, to provide context for the topic.

In Section 3 we design an experiment to assess the effectiveness of the technique of audio stream separation. We present a model of human music perception, followed by a high-level conceptual design for an experiment based upon this model. We then address the low-level and practical ramifications of this design and describe our actual technique.

In Section 4 we present, organise and briefly comment upon the results of our experiment. This is followed in Section 5 by a more in-depth discussion of these results: the trends evident therein, the factors driving these trends, the significance of the work and the ramifications of our results.

Section 6 summarises the results and contributions of this thesis. We conclude by identifying areas either deemed as lacking or promising as future work in this field.

2. Background & Related Work

Chord estimation is closely related to key estimation, melody extraction, beat detection and other musically motivated processes. As such, it is common practice for authors dealing with automatic chord transcription to cover these topics in literature reviews. However, although we draw from some work done in these fields, we exclude their comprehensive treatment from our scope, as our focus is not on developing a new chord estimation method 'from the ground up', but rather on modifying existing methods to improve their performance.

In this section, then, we seek to investigate such existing methods, along with a cursory study of human music perception, in order that we may propose a technique for application and investigation. We achieve this by structuring the chapter into four main parts. In Section 2.1 we provide context for the problem of automatic chord extraction by giving an overview of musical theory and the concepts that are required to understand the methods discussed herein. In Section 2.2 we discuss various contemporary techniques for achieving automatic chord extraction from musical audio and their underlying principles, their effectiveness and their limitations. Following this investigation, we present in Section 2.3 a discussion of possible avenues for improvement, and propose the use of a method based on a model of human auditory perception that would address some of the aforementioned limitations. Finally, we defend the feasibility of this method in Section 2.4 with a review of current techniques in the field of audio processing that demonstrate the potential to realise and implement it.

2.1 Musical Theory

It is reasonable to begin a thesis concerned with the analysis of music by first studying its fundamental principles: after all, the notion of 'automatic chord estimation' is meaningless without first defining what is meant by a 'chord'. The concept of 'pitch' arguably underpins the entire field of music theory; this seems a logical point from which to start, and so we begin there.

2.1.1 Pitch

Put simply, pitch is the inherent characteristic of musical tones that allows a listener to classify one sound as 'higher' or 'lower' than another; however, similarly to the musical notions of loudness and timbre, it is a subjective concept and not easily empirically determinate. Klapuri gives the following definition for pitch [2]:

Pitch is a perceptual attribute which allows the ordering of sounds on a frequency-related scale extending from high to low.

Pitch arises as a consequence of the mechanics of sound. Sound is perceived when vibrations in the air or some other medium are picked up by the ear; these 'sound waves' have a certain repetition rate, or frequency. It is this repetition rate that roughly translates to the concept of pitch [3]. Figure 2.1 shows a waveform sample from a vocal part within a song, with the period marked; the frequency corresponding to this period is the perceived fundamental pitch. A higher frequency generally means a higher perceived pitch [4], although the relationship between frequency of the signal and the perceived pitch is not wholly straightforward – a concept we explore shortly.

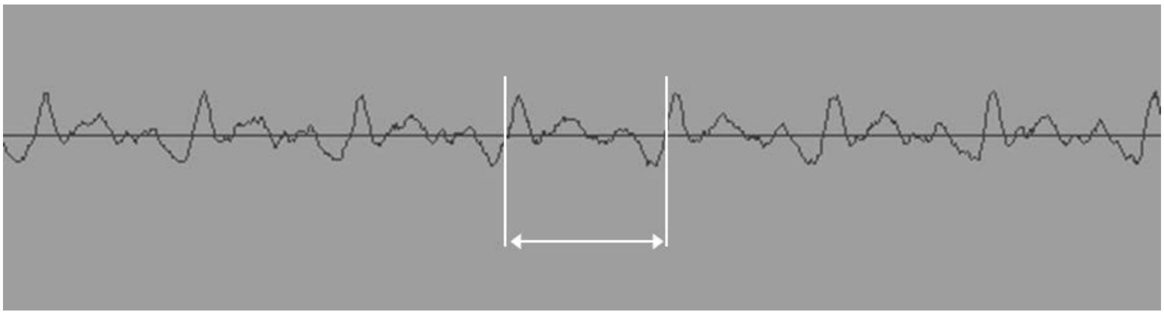

Figure 2.1: Complex waveform with fundamental period indicated

Pitch has two main aspects: *pitch class* and *pitch height*. Pitch height refers to the aforementioned 'highness' or 'lowness' perceived in a tone, and is that characteristic of the sound that generally varies proportionally to the frequency. Pitch has a special property in that pitches that are an integer number of octaves apart – that is, with frequencies satisfying $\frac{f_2}{f_1} = 2^n$ for some integer *n* – are perceived as being higher or lower equivalents of the same tone [5]. This phenomenon is known as pitch class (or *pitch octave equivalence*), and any two frequencies satisfying this relationship are defined as belonging to the same pitch class [6]. Western music defines twelve standard pitch classes[1]: A, A#, B, C, C#, D, D#, E, F, F#, G and G#. (See Figure 2.2 for an illustration of these pitch classes on one octave of a piano keyboard.) These twelve standard classes are also known as *notes*, and hereafter shall be

[1] In actuality, the *enharmonic equivalents* – that is, notes denoted here with a *sharp* ('#') after the letter – also have equivalent *flats* ('♭'), so that, say, A# and B♭ are the same note (the note one semitone above A and one below B). Generally the sharps are used when listing an ascending scale and the flats for a descending one. However, for simplicity, we only use the sharp notation in this thesis.

referred to as such; they comprise what is known as the *chromatic scale*, where a *scale* is an arrangement of notes by pitch.

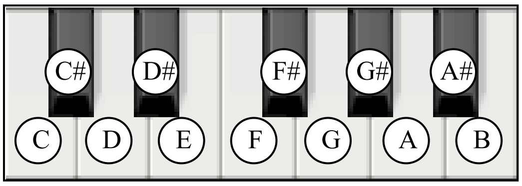

Figure 2.2: The twelve pitch classes illustrated on one octave of a piano keyboard

It is standard in Western music to define a reference pitch, 'concert A' or 'A_4' (the subscript '4' denoting the pitch height of the octave), at 440 Hz. By the rule above, this means that the frequencies of the 'A' notes above and below concert A – that is, A_5 and A_3 – are 880 Hz and 220 Hz respectively (a doubling and halving of 440). The remaining pitches in the note alphabet are equally logarithmically spaced[2], satisfying

$$\frac{f_2}{f_1} = 2^{\frac{1}{12}}$$

for the frequencies of any two consecutive notes (i.e. notes a *semitone* apart, as the interval is named), such that ascending any twelve successive pitches will yield a doubling in note frequency and hence a return to the same pitch class (since we define pitch class as being cyclic every twelve semitones).

As discussed, pitch does not always correspond exactly to frequency. Consider a piano playing an A_3 note with a frequency of 220 Hz. To create the sound, a hammer in the piano strikes a string under tension, causing it to vibrate. Because the string is fixed at both ends, it can only vibrate at certain wavelengths, as illustrated in Figure 2.3 which indicates the first four modes of vibration and their associated wavelengths. These wavelengths correspond to particular frequencies called *harmonics*. (A similar mechanism occurs in wind instruments to cause vibrations only at certain wavelengths.) Depending upon various physical properties of the mechanism – the material of the hammer, the velocity of the strike, the density of the string, and so on – each harmonic will have a different intensity level; there will also be some frequencies present which do not correspond to a harmonic exactly. The terms *overtone* or

[2] This definition actually only holds for what is known as the *equal-tempered* scale: in this scale, the octaves are tuned to consonance and then the remaining notes are spaced evenly in a logarithmic sense. This has been the standard scale definition for around two centuries. Historically, however, there have been many other kinds of scale that involve interval definitions based on rational numbers rather than roots of two.

upper partial are used to refer to all frequencies other than the fundamental that are present. This is illustrated in Figure 2.4, which shows the spectrum of an A_3 note (220 Hz) as played on a piano; the fundamental is evident at 220 Hz as the largest peak on the left, with the various upper partials appearing as all the other peaks above 220 Hz.

Interestingly, the various overtones when sounded in conjunction are not perceived by the ear as a group of distinct tones, but rather as a single complex tone with one fundamental frequency – in this case, 220 Hz. The unique combination of the particular levels of certain harmonics, instead of being perceived polyphonically, generate what is known as *timbre* in sound, and it is what allows listeners to distinguish between instruments playing the same note. Hence whilst a listener may hear a single tone and identify it as A_3, a computer would instead 'see' several frequencies and may interpret it as a polyphonic sound.

The relationship between pitch and frequency is even more convoluted than merely this phenomenon, however. For example, in 1938, Schouten demonstrated that the predominant perceived pitch in a stimulus containing harmonically related pure tones correlates to the greatest common divisor of the frequencies present (i.e. the *fundamental*) – even if this fundamental has *no actual spectral energy* in the stimulus [7]. In other words, a listener presented with two simultaneous pure tones containing a common divisor of their respective frequencies – say, 800 Hz and 1 kHz – will perceive this as a complex tone of frequency 200 Hz, the largest common factor of these two tones, even though this frequency is not represented in the spectrum at all. Furthermore, this phenomenon manifests itself only for such fundamental frequencies up to 1000 Hz [8]. Such qualities of pitch are what render it solely a perceptual attribute and not an objective one; the relationship between perceived pitch and actual frequency is still not yet fully understood. This uncertainty makes the process of automatic chord extraction considerably more difficult, as it means that the human brain can perceive pitches that simply do not exist to a computer, and is a large obstacle to the development of chord estimation methods that work on a very low level (that is, considering mainly the audio information rather than taking into account high-level musical context). Nonetheless, there is still a fundamental link between pitch and frequency, and we make use of this fact later in developing an experiment to test chord extraction.

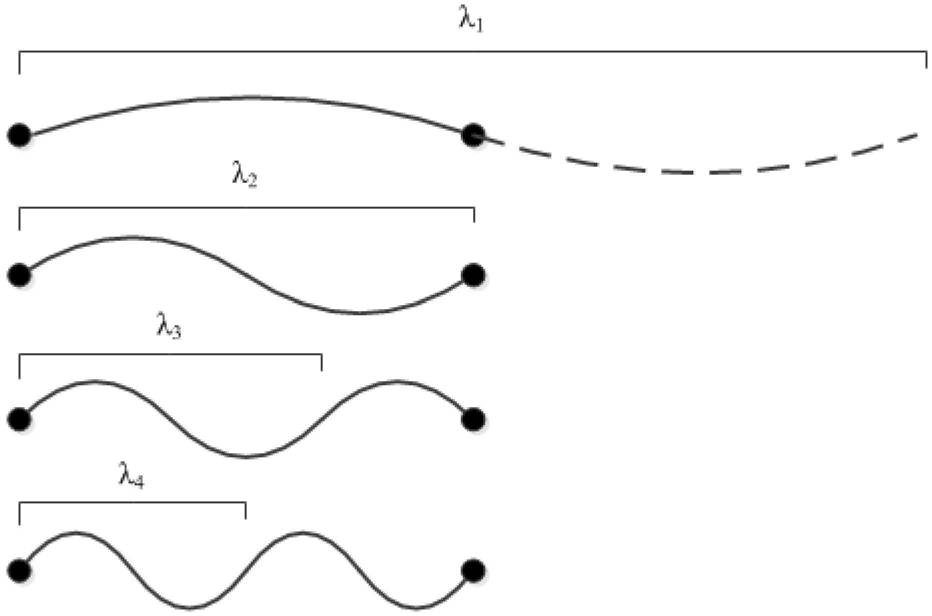

Figure 2.3: The first four modes of vibration of an idealised tensioned string fixed at both ends and their wavelengths

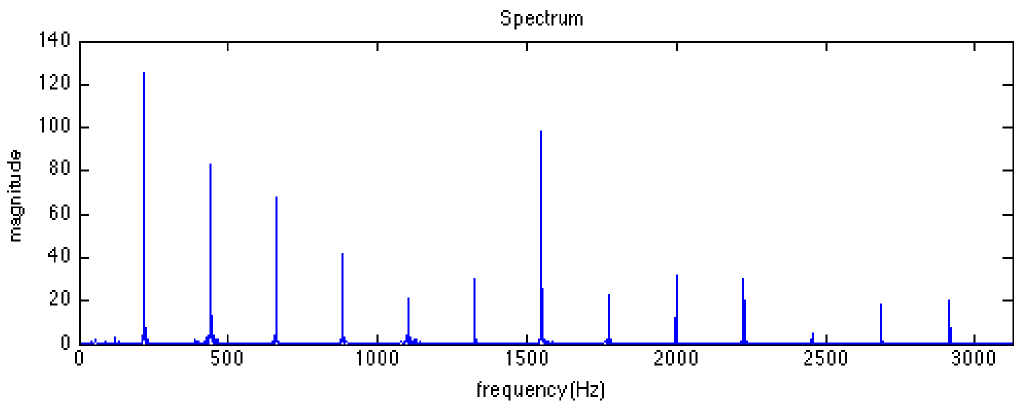

Figure 2.4: Spectrum of 'A_3' note on piano showing 220 Hz fundamental and upper partials

2.1.2 Chords

The concept of chords, then, builds upon that of pitch. A chord is defined as "a group of (typically three or more) notes sounded together, as a basis of harmony" [9]. Theoretically, there are hundreds of chords that can be formed from this definition; however, Western pop music is based upon twenty-four main ones, which can be modified in many ways to create many variations. These consist of twelve *major triad* chords and twelve *minor triad* chords, one each for each of the twelve notes of the chromatic scale: for example, C major and C minor, C# major and C# minor, and so on for each note. A major triad is formed by selecting a *root* note (from which the chord's name is derived), the note four semitones above, and the note seven semitones above; for minor triads the middle tone is only three semitones above the root instead of four. An A major chord would therefore be formed from the notes A, C# and E, but an A minor chord would consist of A, C and E. (We denote in shorthand a minor chord with a lowercase 'm', so that A minor is denoted as 'Am', whereas A major is simply 'A'.) The formation of the chords A and Am is shown in Figure 2.5. Note that only the third in each triad differs, whilst the root (A) and fifth (E) are shared.

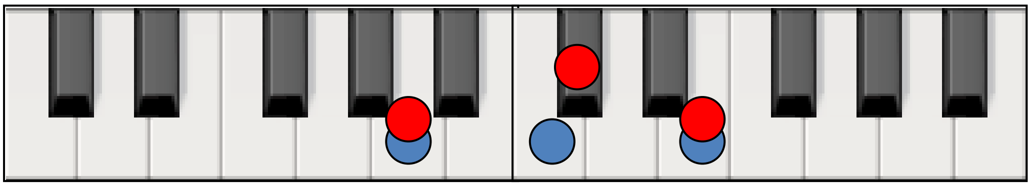

Figure 2.5: Forming the A major (red) and A minor (blue) triads

Thanks to the octave equivalence property of pitches, chords played any number of octaves above or below, or in different *inversions* (with the same pitch classes represented at different heights, in different orders, or even combinations of the two) are perceived and classified as the same chord. For example, the notes D_3, $F\#_3$ and A_3 sounded together are classified as a D major chord, just as the notes $F\#_3$, A_3 and D_4 would be, even though they are in a different order. Figure 2.6 illustrates this, showing the transposition of the D_3 note up an octave; both chords are still D major.

Various chords can also have relationships between one another. Two relevant relationships we consider in this project are *relative* major/minor chords and *parallel* major/minor chords. The relative minor chord of a given major chord is found by transposing the chord's root down three semitones and changing its *tonality* (major or minor quality) to minor, or vice versa (up three semitones and to major) for a minor chord. For example, the relative minor of C major is A minor (as three semitones down is C-B-A#-A) and thus the relative major of A minor is C major. The concept of a parallel major or minor is much simpler: we merely switch the tonality of the chord, so that the parallel major of G is Gm, and the parallel major of Gm is G.

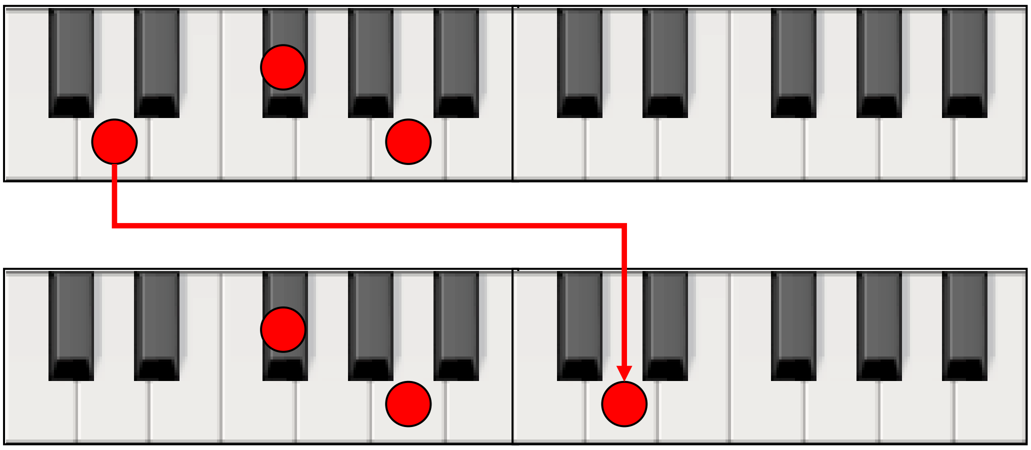

Figure 2.6: Two inversions of the D major chord

2.1.3 Key

A key is "a group of notes based on a particular note and comprising a scale, regarded as forming the tonal basis of a piece of music" [9]. A key generally uses seven of the twelve pitch classes, and the selection of these (and which one is the root note – the note on which the scale is 'resolved') determines what kind of key it is. For example, the notes F, G, A, A#, C, D and E, where F functions as the root, form the F major scale, and hence the key of F major. In practical terms, a key represents a group of tones that sound pleasant together; conversely, a note that is not in the current key will (usually) sound unpleasant or dissonant to the listener.

2.1.4 Genre and 'Pop' Music

'Genre' is defined as "a style or category of art, music, or literature"; the word *genre* itself derives from its French homograph, meaning "type" or "kind" [9]. In the musical sense, it is a commonly employed term for categorising different types of music by similar features, some of which include instrumentation, rhythmic patterns, and pitch distributions [10].

The relevance of the concept of genre to the automatic transcription of chords from audio may reasonably be called into question. We include it in consideration because of its impact upon the chordal content of songs; the literature records a notable correlation between song genre and the use of different chord progressions within that genre. This idea was first noted by Piston [11] and more recently demonstrated by Anglade et al. [12]. Consequently, songs from different genres will exhibit different harmonic characteristics. As discussed in the introduction, our work will focus on pop music, and so an understanding of the structure and musical nuances of this style of music is crucial to its proper analysis.

The definition of what makes a song a pop song is highly subjective. There is no official quantitative definition in existence for classifying a song's genre as pop; the concept, as with most genre definitions, is a mostly qualitative one only. There are some musical characteristics, however, that are highly prevalent throughout the body of music regularly classified as pop. Generally, songs are roughly two-and-a-half to three-and-a-half minutes long [13]; have a highly prominent element of rhythm or 'beat' to them [13]; strongly emphasise melody (usually carrying it in the lead vocal of the song) [14]; emphasise a song structure revolving around verses, choruses and bridges [13]; and utilise common chord progressions, often not venturing far from a single key [15]. These qualities can be used as a loose guide for classifying a song as pop. However, given the subjective nature of the definition, it is helpful to also consider both the artist's and the general public's opinion (where available) on the genre of a particular piece or artist. For example, the practically ubiquitous iTunes Store – a software-based online store retailing digital media and content, owned by Apple Inc. – is the largest online music store in the world [16]; it sorts songs by genre [17], and by virtue of its global prevalence provides a useful reference for general artist and public opinion on the classification for a particular song or artist. We make use of both the qualitative definition and this technique (where possible) in determining the genre of songs to use for our experiment in Section 3.3.1.

This concludes our discussion of the musical theory necessary to understand this dissertation, having presented a brief summary of pitch, chords, key and genre. We now proceed to discuss the field of automatic chord transcription.

2.2 Automatic Chord Transcription

The problem of chord extraction from audio is a complex one, and as such, the solutions that have been devised involve many techniques: harmonic information extraction, signal processing, noise reduction, tuning correction, beat tracking, musical context modelling, and so on. Many of these techniques are not exclusive to chord extraction (for example, noise reduction and signal processing considerations), and so whilst a comprehensive review of automatic chord transcription necessitates the study of each of them, we are more concerned with the overall process of chord estimation itself. (For a detailed summary of the development of the listed techniques, see the PhD dissertation "Automatic Chord Transcription from Audio Using Computational Models of Musical Context", by M. Mauch [18].) We instead focus on studies that aim for automatic chord extraction – that is, the complete process from audio input to chord information output – as an overall goal, and not merely studies of sub-processes that have more general applications.

The process of automatically transcribing chords from audio arguably traces its origins back to Takuya Fujishima, who in 1999 first introduced the concept of Pitch Class Profiles (PCPs) for analysing the harmonic content of a song [19]. PCPs provide a vectorial representation of the spectral presence of the twelve pitch classes in a segment of audio by classifying all harmonic content as belonging to one of these twelve classes. For example, a spectral peak at 1400 Hz would be assigned to the F pitch class, as the nearest standard pitch is the note F_6 at 1396.9 Hz. Fujishima utilised these to represent segments of audio as vectors and then classified them as chords by taking the inner product of these vectors with predefined chord template matrices to yield chord-likelihood values for each possible chord [19]. Wakefield later developed the mathematical foundation for the similar concept of the *chromagram*: a representation of the signal that discards the quality of pitch height and shows only the *chroma* information[3], wrapping all pitches to a single octave and classifying them to the twelve pitch classes [20]. One such vector is shown in Figure 2.7. These techniques provided a way for chroma information to be extracted from audio, and allowed for pitch-based analysis of musical audio in the frequency domain.

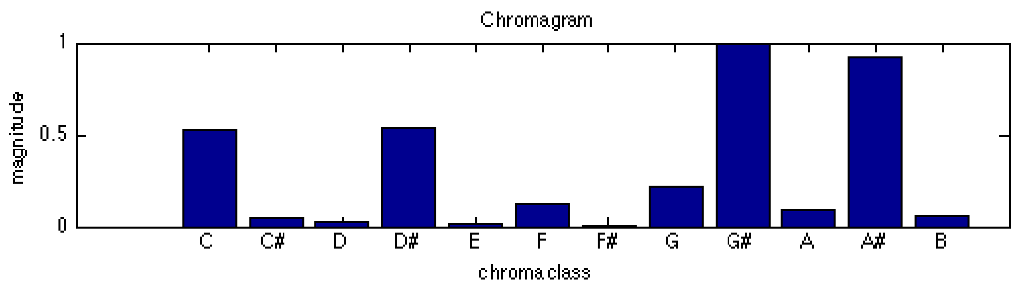

Figure 2.7: A pitch class profile or 'chromagram' vector in graphical form

Since the work of Wakefield and Fujishima, PCP or 'chroma' vectors have become near-universal techniques for performing automatic chord estimation, underpinning almost all methods thereafter. Sheh and Ellis use chroma vectors in their work on recognising chord sequences in audio. They employ a *hidden Markov model* (HMM) – a statistical model of a system's states that assumes some states are not visible to the observer – to model the chords present. This model is trained with an *expectation maximisation* algorithm, analogously to speech recognition, to select the most likely chord sequence and thereby recognise and align chord sequences in audio [21]. Harte and Sandler use a constant-Q transform (similar to a discrete Fourier transform, but with a constant ratio between centre frequency and

[3] Chroma is a quality of pitches similarly to how hue or brightness are qualities of colour; a pitch class is the group of all pitches with the same chroma.

resolution[4]) to obtain a chromagram representation of audio files; these files are then multiplied with simple chord vector templates, like Fujishima, to obtain a frame-wise estimate of chords [22]. Bello and Pickens also use chroma-based representation, along with a HMM as used by Sheh and Ellis, demonstrating the popularity of these techniques for chord estimation; however, they explicitly incorporate musical knowledge into their model to aid performance [23]. Mauch does this also, and furthermore develops his own slightly modified process for extracting chroma features from audio, which attempts to map the observed spectrum to a most likely set of pitches so as to decrease false note detection from overtones; in other words, attempting to represent pitch *salience* (perceived strength) rather than just spectral power [18]. Mauch's work is notable in that it also uses high-level musical context models to support the low-level feature analysis present in many of his precursors' works. For example, he models chord, key, bass and metric position as separate state variables in a *dynamic Bayesian network* (DBN; another statistical model for a system's states, similar to the HMM), rather than just the chord as in other methods; he also develops an algorithm which analyses song structure and makes use of repetition cues to assist chord extraction. The result is a chord recognition score of 81%, setting a new benchmark for chord transcription.

There appear to have developed two main ways of tackling the problem of automatic chord transcription: one involving simple template-based models, grounded in musical theory and chord templates (Fujishima, Harte and Sandler), and the other involving machine learning techniques such as HMMs or DBNs (Bello and Pickens, Mauch). However, one thing quite evident in most methods to date is that nearly all methods rely on chroma extraction, either via PCP or chromagram, as a fundamental step of the process. We keep this in mind as we now turn to the discussion of how current techniques of automatic chord transcription might be improved.

2.3 Improving Chord Estimation Accuracy

Most chord estimation methods can be divided into two main steps: the extraction of chroma information from the song (sometimes referred to as *low-level feature extraction* or simply the *front end*), and then the processing of this information to estimate the chord (*high-level processing*). These are represented in Figure 2.8. Taking this perspective leads to the conclusion that there are consequently two main ways in which the accuracy of such methods

[4] The discrete Fourier transform and constant-Q transform differ in that the term inside the summation of a constant-Q transform is multiplied by a window function whose width decreases proportionally with frequency (that is, logarithmically); the DFT has no such window and as such maintains a constant, linear 'bin' size, meaning that the ratio of centre frequency to transform resolution differs with frequency. Constant-Q transforms are efficient for analysing music as they correspond well to the logarithmic scale of pitches.

can be improved: by improving the front end, or by improving the high-level processing. We will consider both of these.

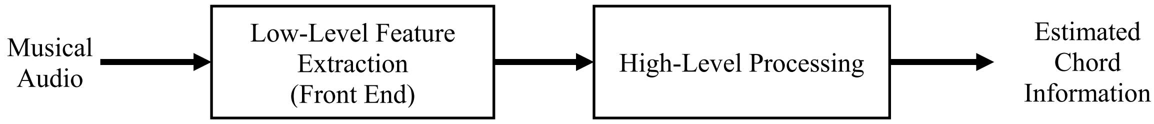

Figure 2.8: The two stages of chord processing methods

The first possibility for improving accuracy, then, is to refine the high-level processing: that is, to develop new and better methods for recognising chords from the information we can extract from songs. Save for harmonic extraction, the techniques listed at the beginning of Section 2.2 all belong to this category, as they are concerned with the processing of extracted chroma information to estimate the chord, and not with extracting the chroma information itself. Refining such high-level processes is the predominant way in which most methods over the last few decades have attempted to make improvements, and the enhancement of any of them would likely yield better results in chord transcription methods. However, each of these comprises but a small part of the overall process of chord estimation, and furthermore not all the techniques are universal: for example, tuning correction and noise reduction are not employed in all chord extraction methods. As such, whilst such improvements are necessary and will probably achieve notable improvements in chord estimation performance, they are not necessarily the most efficient way of doing so.

The other alternative for improving chord estimation accuracy is – rather than developing better ways of *interpreting* extracted features – to increase the fidelity of the front-end feature extraction processes themselves: that is, to find more accurate ways of extracting harmonic and rhythmic information from audio, so that the algorithms which use this data are presented with a more accurate input. To do this, we can consider areas where current techniques are known to have issues.

One such problem evident in chord estimation is the difficulty of correctly segmenting audio into regions containing only a single chord. Since different chords contain different notes, attempting to analyse two chords as one will inevitably lower estimation accuracy (because, at best, only one of the chords can be recognised correctly). Several authors note this problem in their discussions of chord estimation [18] [24] [25] [26]. A commonly employed solution is the integration of a 'beat tracking' algorithm into chord estimation methods, using the assumption that chords usually change on the beat of a song to allow chord change boundary detection [18] [27]. Refining such an algorithm should improve overall chord transcription performance. (We agree that beat detection is an important part of chord estimation and in fact incorporate it into our model in Section 3.2.2.)

The most obvious source of inaccuracy, however, is the property of perceived pitch not always corresponding exactly to frequency, discussed in Section 2.1.1. The problem can be succinctly summarised as this: humans identify chords based upon the *pitches* they perceive, but computers can currently only identify chords based upon the *frequencies* present in a signal. The non-unique and unpredictable mapping between frequency and pitch presents a serious problem for chord estimation methods that do not take into account musical context, and is by far the most prominent reason that automatic chord estimation is such a difficult undertaking.

To illustrate, consider the hypothetical piano of Section 2.1.1, this time playing a polyphonic musical piece. A single note generates multiple harmonics that do not correspond to the perceived fundamental; many played simultaneously will generate a spectral 'haze', making it increasingly difficult with every additional note to determine which tones are harmonics and which fundamentals. The piano may also play what are called 'non-chord tones', meaning these tones do not exist in the chord (though usually they exist in the current key). For example, an A# major chord could be intended in the context of the piece, containing the notes A#, D and F, but the melody might contain a G, which is not part of the A# major chord. Such a scenario is shown in Figure 2.9, which depicts in musical notation a melody featuring a G playing over an A# chord. An automatic method may become confused in this case as the notes A#, D and G are present, which are the notes of the G minor chord.

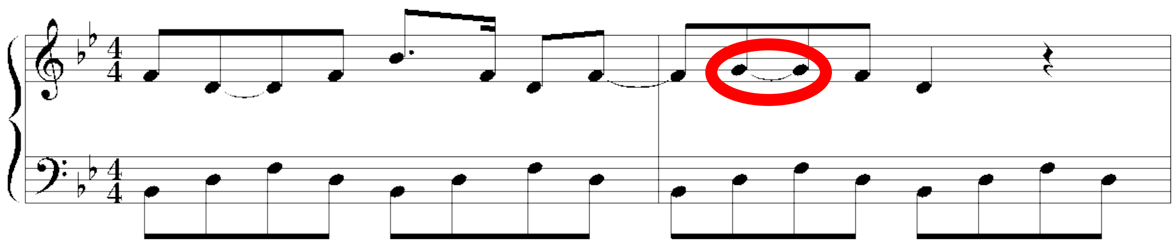

Figure 2.9: A melody in musical notation playing an A# chord with non-chord tone G highlighted.

Now consider added to this piano a lead vocal line, a bass guitar and an acoustic guitar, as in a simple piece by a pop musician. Even if playing or singing notes that match these chords (which is not always the case – bass lines, for example, commonly 'walk' using non-chord tones, and vocal melodies are seldom constrained to chord tones), these additional instruments add their own harmonics according to their timbral characteristics, further complicating the task of extraction. Vocals also add consonants such as sibilance, fricatives and plosives ('s', 'f' and 'p' sounds), which have highly atonal characteristics. Furthermore, imagine that a drum kit is now added, which provides periodic bursts of short, sharp, and largely inharmonic energy: the kick drum interferes with the bass guitar's spectral region, the snare and toms share a frequency range with the mid-range instrumentation, and the cymbal crashes produce powerful spectral bands of noise which cloud all of the upper harmonic range. The frequencies corresponding to the correct harmonics of the originally intended A# chord are now surrounded by a range of frequencies that have no relation to this chord whatsoever. A similar scenario is shown in Figure 2.10, which illustrates the spectra of (from top to bottom) an A# note on piano in isolation, an A# chord on piano, and the A# chord with drums and an acoustic guitar also playing an A# chord. Note the crowding of the spectrum as more instruments are added.

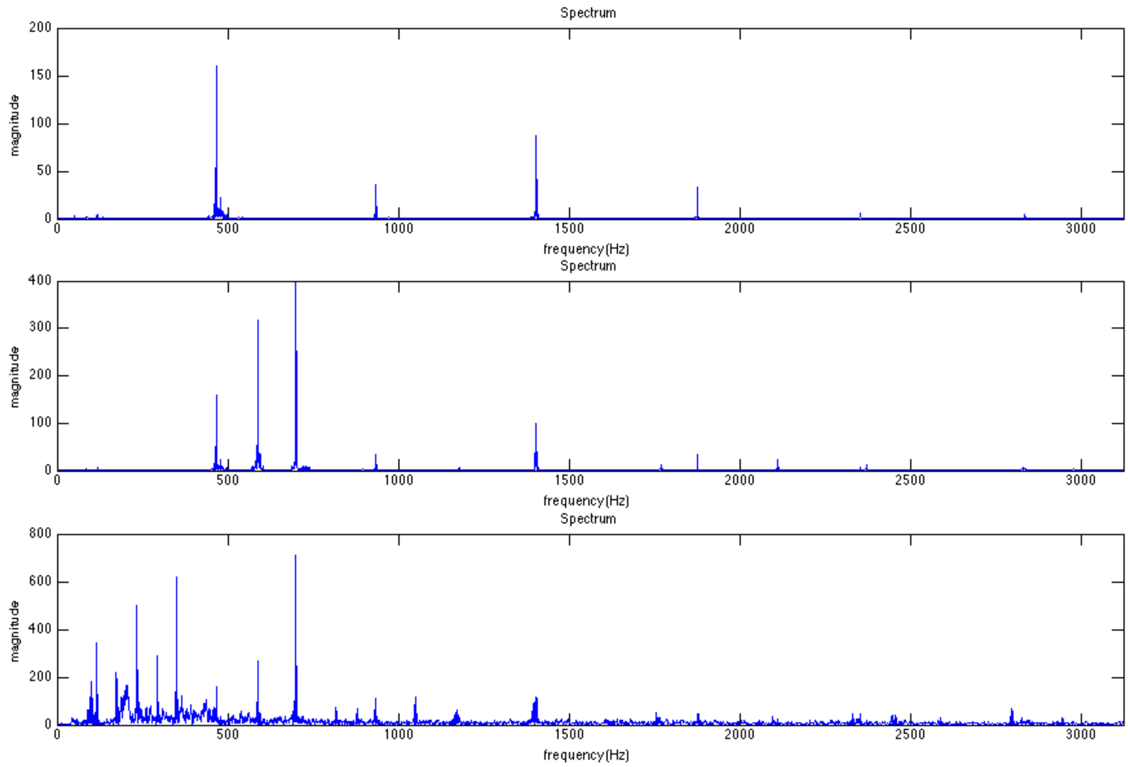

Figure 2.10: Spectra for a single A# note on piano, an A# chord on piano, and an A# chord on piano with acoustic guitar and drums

The reason that this has ramifications for chord estimation is that current methods for calculating the chromagram for an audio file – which, as we mentioned in the previous section, is used in almost all chord transcription techniques – merely assign all spectral content to one of twelve pitch classes, regardless of whether it is perceived as such by the ear, and hence are easily distorted by noise or overtones which are falsely classified. Regardless of how advanced a chord estimation algorithm may be, it can only produce results as accurate as the input it receives, and so a chroma vector corrupted by inharmonic upper partials will render even the best high-level algorithms ineffective.

We conclude, therefore, that in order to best improve chord transcription, we must address the inaccuracy inherent in creating chroma-based representations of audio, as this forms an upper bound to the accuracy of any overall method of chord estimation. To do this, we consider human music perception.

2.3.1 Audio Stream Separation

We have argued that the since the brain outperforms computers at the task of chord recognition, computerised methods should seek to imitate its operation. This, of course, necessitates the consideration of how people actually perceive music. However, since this is an engineering thesis, we cannot deal with this topic in any great depth. Instead, we focus on a single phenomenon largely ignored by the literature in the area of automatic chord estimation.

A notable feature of the process of human audition is that the brain is able to group perceived audio 'streams' by their source. For example, a listener is able to discern individually the vowel sounds /aː/ ('ah') and /iː/ ('ee') at different pitches even when they are sounded simultaneously [28]. In a musically specific sense, this concept applies to the ability to discriminate between various elements in the arrangement of a song: bass guitar, percussion, vocals, piano and other instruments can likewise be identified as separate 'streams'. The reader might wish to confirm that this is the case: were they to listen to a pop song, they would note that they would be able at leisure to identify various instruments in the mix individually, even as these instruments play simultaneously (provided they were not obscured enough by others). For example, listening to Billy Joel's 1973 hit "Piano Man", a listener could discern between the vocals, drums and piano by 'focusing' on each at will.

This idea, then, if applied to the field of automatic chord extraction, could allow algorithms to achieve performance closer to that of the brain. Harmonic recognition from audio achieves better performance with less polyphony and fewer instruments playing simultaneously [23]; therefore, if it were possible to separate instruments in a polyphonic segment of audio, it may prove to be easier to analyse the harmonic information present in this audio. Considering once more our example of the multi-instrumental piece in Section 2.3, it can be seen that almost all of the problems present in the example would be remedied by the capability of a chord detection algorithm to separate and isolate separate streams to prevent them from harmonically interfering with one another.

It is for this reason that – as the reader may recall from the introduction – in this dissertation we investigate the simulation of the process of auditory stream separation by computers, with the explicit motivation of automatic chord transcription. We hypothesise that a chord detection method that contains audio stream separation as a pre-step will outperform one that does not when considering chord recognition performance. Section 3 will deal with the development of an experiment to test this hypothesis in more detail; however, before we do so, it is expedient to justify the feasibility of implementing such a technique.

2.4 Feasibility of an Audio Stream Separation Technique

Whilst the concept of considering audio streams separately appears promising, it is a fruitless endeavour if there exists no technological capability to separate audio streams. We discuss here some recent developments in the field that show that automatic stream separation is firmly in the realm of reality and not merely a theoretical consideration.

2.4.1 Separation of Harmonic and Percussive Elements

At the heart of any method of stream separation is the segregation of the two most distinct elements of music: percussive elements and harmonic elements [29]. A cursory definition would be that percussive elements are those parts of the song that primarily provide a sense of rhythm to the song and do not contribute to the harmony of the song; in pop music, this is usually the drums. Harmonic components, on the other hand, are those that contribute to the harmony or melody of the song: that is, anything having a noticeable pitch. Most instrumentation in pop music satisfies this criterion, as it includes elements such as voice, piano, guitars and so on. The inharmonicity of percussive elements means that they detract from the performance of a purely spectrum-based method of chord extraction by adding noise that does not actually belong in any pitch class: conversely, the presence of non-percussive instrumentation can hinder the process of automatic beat detection from the percussion, which in turn hinders chord change estimation [30]. Hence it is imperative that a method for automatically separating streams is able to identify percussive and harmonic elements of a song and separate the former from the latter: effectively, separating the drums from the music.

There have been methods that have proposed to use MIDI[5] information in order to separate elements in music by using the note information and knowledge of instrumental characteristics [31]. However, in considering the automatic transcription of chords, we cannot assume MIDI data to be available, as the process should work on a song with no *a priori* information. Hence we restrict our discussion to those methods that use audio data alone.

---

[5] Musical Instrument Digital Interface; a technical standard that defines, amongst other things, a digital protocol for transmitting musical note information between devices. MIDI data contains note onset times, velocity, pitch, and other musical information.

Uhle et al. propose a method for the separation of drum tracks from polyphonic music based on *independent component analysis* (ICA). ICA is an approach to the problem of blind source separation – the estimation of component signals from an observation of their linear mixtures with limited information about the signals themselves [32]. The final stream classification is performed using audio features (characteristics of the audio in both the temporal and spectral domains, such as spectral flatness, spectral dissonance and periodicity), and manually chosen decision thresholds; however, despite allowing such manual intervention in the process, the authors describe achieving only 'moderate' audio quality in the extracted signals [29]. One of the fundamental assumptions of ICA is that the individual source signals are statistically independent of one another; we suspect that this assumption is detrimental to the effectiveness of the method, as the rhythm and harmony of a song are rarely wholly independent of one another, but rather complement one another and are often synchronised.

Helén and Virtanen propose an alternative technique to that of Uhle and his colleagues, adopting a machine learning method instead. They first use a technique known as *non-negative matrix factorisation* (NMF) to decompose the input spectrum into component sub-spectra with fixed ranges but time-varying gain, and then a *support vector machine* – a pattern recognition method based on statistical learning theory – which they train on 500 manually separated signals in order to classify these sub-spectra to either pitched instrumentation or drums [30]. They argue that this not only allows the process to become more automated, but also allows more features to be used in the automatic classification. Helén and Virtanen report achieving classification results of up to 93% with certain feature sets used for the classification process, a remarkable result; however, it should be noted that these are spectral bands and not actual audio streams. These results are therefore perhaps somewhat deceptive, as the assumption that different audio streams correspond exactly to different spectral bands is flawed: few instruments other than the bass occupy an exclusive frequency band, and so the end result would not be as absolute to the ear as the numbers might suggest. This is not to state that this result is devoid of usefulness, but rather that caution must be taken before treating it as a measure of pure stream separation performance.

Perhaps the most promising work on harmonic/percussive separation, however, is that developed by Ono et al., who propound the novel idea that harmonic and percussive elements can be separated based upon their appearance on a *spectrogram* – a visual representation of all the frequency components in a signal, with time on the horizontal axis and frequency on the vertical axis [33]. Pitched instrumentation produces sounds with mainly harmonic power content, whereas non-pitched percussion produces highly inharmonic sounds, containing a large spread of frequencies, at high power but only for a short, transient time. Ono et al. suggest that in terms of the spectrogram, this means that the contribution of pitched instrumentation is to create smooth, parallel lines which run horizontally as they maintain a constant pitch (or harmonically related series thereof) over time; percussion, however, appears as vertical lines, representing the large spread of frequencies over a short time. Figure 2.10 shows an example of such a spectrogram, exhibiting vertical lines representing the percussive onsets and horizontal lines representing the harmonic pitches present.

Ono et al. report good separation of drums from piano and bass guitar using their method. FitzGerald further develops this method and reports a faster and more effective result, actually obtaining separate tracks with which he is able to create remixes and adjust the relative volume of drums and pitched instrumentation [34]. Clearly, then, the capability for automatic separation of percussion tracks from harmonic instrumentation currently exists.

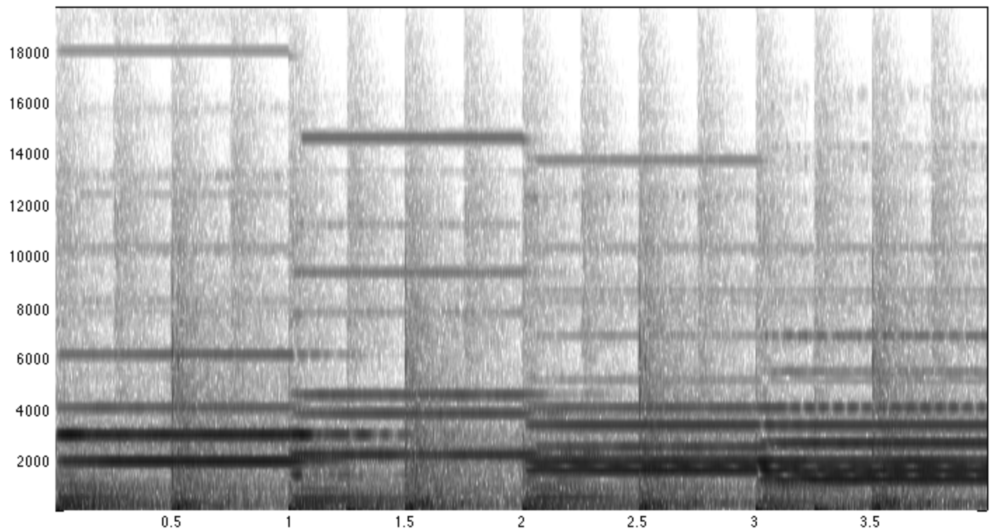

Figure 2.11: Spectrogram showing a mixture of percussion (vertical lines) and harmonic instrumentation (horizontal lines)

2.4.2 Extraction of Bass and Vocals

A more difficult problem than extracting drums from harmonic components is that of extracting certain harmonic components from a blend of many. The spectral characteristics of rhythmic instrumentation are different enough to that of harmonic instrumentation that they may be separated fairly easily, but when faced with pitched instrumentation playing the same notes, extraction is more difficult. In order to adequately perform stream separation, however, it is necessary to do more than simply extract drums: we must be able to split various pitched instruments, or instrumental 'streams' as we may refer to them, from one another. Two of the main streams we consider here are the bass and the vocals. The vocals are chosen as they are obviously the most prominent element of any pop song, usually carrying the lyrics; the bass, since it serves a special function in suggesting the chord that is being played, and it occupies a unique frequency band (the *low end*) that no other harmonic instrumentation fills, making it easier to extract [18]. An effective stream separation method (at least for our purposes) must then not only be able to extract drums, but vocals and bass guitar also.

Whilst not as completely developed as the methods for separation of drums from harmonic instrumentation, the capability still exists for bass and vocal extraction to be performed. Rao and Rao claim 'significant' success in their work on extraction of a vocal melody from polyphonic accompaniment, improving on existing techniques by creating a method capable of tracking two separate 'lead' melodies; for example, a vocal line and a guitar solo [35]. Furthermore, Goto reports an 80% success rate in detecting bass lines in polyphonic audio with an automatic detection method, using prior knowledge of the harmonic structure of bass lines in order to map to a most likely frequency [36]. Ryynänen and Klapuri use a combination of the two methods and are able to analyse 180 seconds of stereo audio in 19 seconds, demonstrating the feasibility of real-time implementation [37].

Such methods show the capability for extracting melody line information for bass and vocal components exists. However, it is not sufficient to merely 'detect' these: they must be extracted in the audio domain to be of any use. Iyotama et al. achieve an encouraging result toward this end, developing a model that allows note information (in their case, in the form of MIDI data) to be used to achieve the manipulation of individual audio components' levels within a mix [31]. Using this method, they achieve a SNR (signal-to-noise ratio) of up to 5.73 dB when separating streams; they claim that an SNR of 8 dB should be sufficient for many practical applications. This would allow notes found from a bass or vocal melody in a music track to then be used to extract these components in audio form for separate processing.

It is perhaps worth noting that these methods – melody extraction from audio (Goto, Rao & Rao) and stream extraction from melody (Iyotama et al.) – have been developed independently, and to the best of our knowledge were not intended to be used in conjunction: however, as can be seen, there is great potential in combining the two into an integrated method.

2.4.3 An Integrated Method of Stream Separation

Finally, there have also been methods that have attempted to tackle the overall problem of stream separation from musical audio as a comprehensive problem (in contrast to performing separate individual stream separations). Wang and Plumbley attempt this using NMF, like Helén and Virtanen, to decompose the spectrograms of audio signals into separate matrices:

one representing the temporal location of spectral feature occurrence and the other representing the spectral features themselves (e.g. the pitches playing at a given instant) [38]. Unfortunately, despite declaring their intention to develop a "stream separation methodology", they only demonstrate the technique on a single real audio example of a guitar and flute playing together. They describe their results as "acceptable" (being unable to quantify this as the original guitar and flute files are not available to them). However, in their experiment these two instruments play in a harmonically and rhythmically unrelated fashion, which is clearly an inaccurate sample of real music (considering that music fundamentally requires instruments to play in time and in tune); consequently, the results they achieve are likely overestimated relative to what they would be for a real song.

The somewhat related field of musical *instrument* recognition from audio (not *chord* recognition) has given rise to some other attempts at automatic stream separation. Ozerov et al. develop what they term a *flexible audio source separation framework* that uses Gaussian models and NMF to achieve various kinds of audio separation. One application they propose for this is the separation of drums, bass and melody in polyphonic music [39]. Unfortunately their results are not publicly available, but Bosch et al. employ their method to some success, reporting 32% improvement in instrument recognition compared to an algorithm that does not use the method [40]. (It is unclear how exactly this result would translate to chord recognition scores, but it seems a fair assumption that improved instrument recognition would allow for better chord recognition also, by implication of a clearer signal.) Heittola et al. also propose a method motivated by instrument recognition, decomposing a musical signal into spectral basis functions, with each modelled as the product of an excitation with a filter, somewhat conceptually similar to Wang and Plumbley's work [41]. Again, they achieve a respectable result in terms of instrument recognition (59% for six-note polyphonic signals), although it is not entirely obvious how this translates to chord recognition.

It is manifest, then, that the process of automatic stream separation from musical audio has not been achieved yet in any completely effective form. Various parts of the process exist, and some efforts have been made to create unified methods, but it still remains for a comprehensive, automatic implementation to be created. However, the abundance and diversity of the abovementioned processes demonstrate ample potential for such a technique to be realised in future, and it can be seen that whilst the accuracy of some methods are somewhat lacking, audio stream separation is not merely a theoretical conjecture, but something that is fundamentally achievable today.

2.5 Conclusion

In this chapter, we have laid the groundwork for testing a new technique of improving automatic chord estimation: namely, using stream separation. We discussed the musical theory necessary for a rudimentary understanding of the processes involved, touching upon the concepts of pitch, chord, key and genre. We reviewed the current state of the art in automatic chord transcription and examined the best methods currently available for estimating chords from audio. From this, we discussed the possibility of using an audio stream separation technique to improve chord estimation performance, and set forth reasoning for why such a method might work. Finally, we demonstrated the feasibility of implementing such a technique by examining current methods capable of being used to implement it. Having therefore established a foundation for the topic, we proceed to actually implement and test the effectiveness of audio stream separation in an experiment in the next chapter.

3. METHODOLOGY

In this chapter we develop and present our design for an experiment to test the effect of automatic stream separation on chord recognition. We first outline a model for human music perception in Section 3.1, concentrating specifically upon the process of stream separation, upon which our experiment will be based. This is followed by the creation of a conceptual design for our experiment in Section 3.2, which sets out a general structure for the method and identifies components of the original process that must be adapted in order to be suitable for algorithmic implementation. We build upon this conceptual framework in Section 3.3 by expounding the detailed design of the experiment and the actual methods by which it is implemented. Finally, we explicitly define our hypothesis in Section 3.4.

3.1 Assumed Simplified Model of Human Music Perception

As stated, our goal is to develop a method that can be implemented by algorithms and takes advantage of the brain's efficiency at recognising chords by mimicking its methods of operation. In order to do this, we first need a simplified model for how the brain deals with music. Our intention is not to develop a fully psychologically accurate analysis of the brain's function when listening to music; this is well outside the scope of this project. Rather, it is to make explicit some of the assumptions involved in creating an appropriate experiment to test our chord estimation technique. Therefore we only choose to model steps relevant to stream separation and chord analysis, rather than all steps involved.

The block diagram in Figure 3.1, then, details the assumed process by which the brain interprets music to detect chords. We will discuss each stage of this process briefly. We reiterate that this is not to gain a robust psychological understanding of the process, but to better understand how a computer-based technique could replicate it.

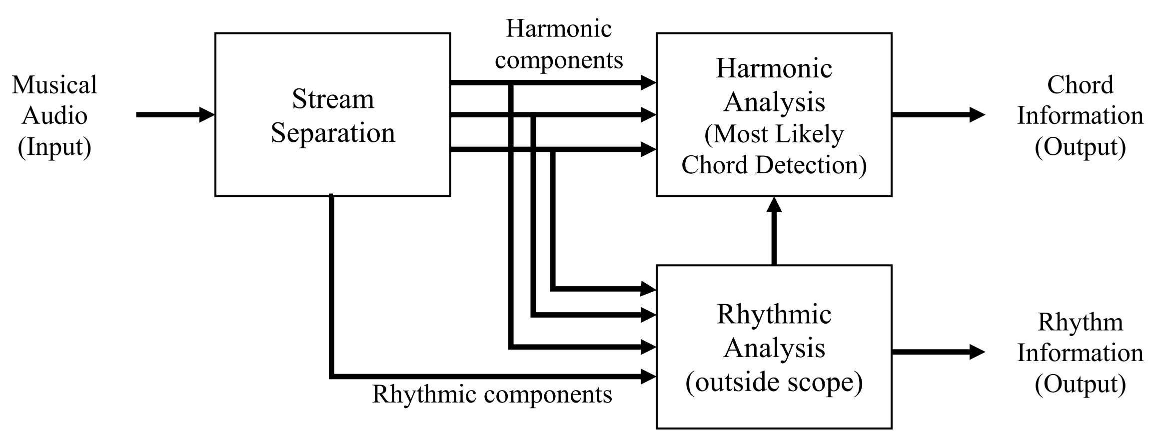

Figure 3.1: Assumed simplified model of human chord perception

Initially, the system receives musical audio as its input, and the process of stream separation as discussed earlier is performed upon this information. The result of this is that all the components of the song are separated and categorised into two groups: rhythmic, or percussive (e.g. cymbals, drums and other non-pitched percussion), and harmonic (e.g. piano, guitar, vocals and other pitched instrumentation). We have shown here only one rhythmic stream, assuming that there will just be one drum track, for example, but there could be more. These rhythmic components, along with the harmonic components (which themselves will contain rhythmic information through details like onset times and playing rhythm, such as the strumming of a guitar), are then used to conduct a rhythmic analysis, which returns information about beat, tempo and other rhythmic information. This kind of analysis is not considered in this project; we recommend it as future work in Section 6.2.2.

The harmonic components are also used to conduct what we have termed a 'harmonic analysis'. It could be argued that a 'harmonic analysis', in the sense of extracting chords from audio, is the goal of this entire procedure; however, we treat it distinctly to the overall process in the sense that it involves only the estimation of chords from purely harmonic content, without the requirement to perform any extraction of interfering streams such as percussion[6]. The actual methods by which this kind of analysis is performed in the brain are unknown: whilst we have assumed that the brain makes use of its ability to separate audio streams in order to better analyse music, we know little more than this, and cannot determine whether hearing, say, the rhythm guitar in a song assists the brain in estimating the notes of a bass guitar playing simultaneously. We are therefore compelled to treat the 'harmonic analysis' stage as a 'black box' – that is, a representation of a system or process where only the input and output are considered – and model only the relationship of the input (harmonic component information) and output (estimated chord information), as the actual workings of this process are unknown. As this thesis deals with engineering and not psychology, any more detailed understanding than this is once more outside the scope of this project anyway.

Finally, the output of this harmonic analysis is the estimated chord information. Methods that take into account musical context (i.e. knowledge of a song's existing key and preceding chords), such as that developed by Mauch, use this information to feed into a larger network combining multiple sources of information [18]; however, we conclude the model here as any further complication is irrelevant to the concept of stream separation. It must be stressed again that this model is highly simplified; nevertheless, the benefit of simplifying the

[6] Note also that the rhythmic analysis is used to assist in the harmonic analysis. This is because harmony recognition is greatly assisted by knowledge of rhythm; for example, chords are far more likely to change with the beat of a song or at least in a way easily interpretable relative to the song's rhythm [27].

process of human chord perception in this manner is that it divides the process into stages that computational methods can simulate.

3.2 Concept Design

Building upon the model presented above, we propose an experiment that implements the various stages of the process as steps capable of being performed either manually or by algorithm. Ideally, this would allow the development of an algorithm that mimics exactly the natural process for perceiving audio. However, there are certain parts of the process that must be adapted in order to conduct a fair experiment; a discussion of these follows.

3.2.1 Input and Stream Separation

The primary issue with implementing the assumed model of the brain's audition process is that of stream separation. It is evident that any audio that humans hear is not in an already separated form; when a song plays on the radio, it plays not as distinct tracks for the drums, bass, piano, vocals and whatever other instrumentation there may be, but rather as a single composition of all these elements mixed together. We have suggested that the brain is responsible for separating these streams during the process of perception. However, there are two reasons we choose not to imitate this in our experiment.

The first is a matter of practicality: stream separation by algorithms is far from a perfected art. We have discussed the work of others in Section 2.4 in performing certain subsets of the process that we have termed stream separation, but the fact of the matter is that none of these techniques is yet flawless. Furthermore, it remains for an algorithm to be created that can simultaneously implement each of these techniques and correctly separate all the streams present in a song automatically. The development of such an algorithm would likely fill a dissertation in its own right. Consequently, we do not develop the topic herein, but leave this to the efforts of future researchers in this field (see Section 6.2.1).

Another reason that this experiment does not handle the implementation of stream separation is that of accuracy. Were an automatic method of stream separation employed here, the results would doubtless not result in a perfect separation of streams; there is no evidence to suggest that even the brain is capable of such a feat. Given that the aim of this experiment is to test whether stream separation improves chord estimation accuracy, it is unwise to leave the effectiveness of the stream separation process as an uncontrolled variable and thereby introduce unknown error into our results. Rather, it is far better to separate the streams manually (or begin with separated streams, as we do) and therefore wield exact control over the degree of stream separation. (This topic is further discussed in Section 3.2.3.)

Consequently, then, this experiment uses as its input what audio mixing engineers refer to as song ‘stems’: segments of a song corresponding roughly to the theoretical ‘streams’ previously discussed [42]. In this project, four stems are used: drums and percussion, vocals, bass and other instrumentation. (We define these classifications in more detail in Section 3.3.1.) These particular streams are chosen primarily because they are distinctive enough to yield meaningful results whilst not being overly complex in their analysis. We also noted in Section 2.4 that these are streams for which previous methods have been developed that are at least partially capable of separating them, and therefore it is theoretically possible to actually perform this extraction.

3.2.2 Rhythmic Analysis

The second way in which this experiment differs from the theoretical process is the rhythmic analysis. The justification for this is similar to the case of stream separation: although rhythmic analysis has basically been implemented before in many methods (usually called ‘beat detection’ or ‘beat tracking’), to actually use such an automatic method to perform our extraction would introduce unnecessary error. Even if we used a method that was 90% accurate, this would still introduce error relative to a perfect method and thereby add another independent variable into our experiment. As such, rather than using a rhythmic analysis, we opt to manually separate the audio files such that each segment will contain only one chord. This removes the need for a rhythmic analysis to be performed and also means that it is possible to test the method’s chord estimation accuracy independent of its beat separation accuracy.

Because this stage now involves separating the input files by hand, it must be performed before the input stage, so that the input is actually a series of files containing one chord each rather than a multichordal sequence. This is unlike the theoretical model in which the rhythmic analysis (and hence chord separation) would be performed after the input has been fed in.

3.2.3 Signal Mixing

Section 3.2.1 noted that it is not known whether the brain is actually capable of perfect stream separation, and stated that regardless, because the objective of this experiment is to assess its effectiveness as a chord estimation technique, the degree of stream separation is controlled manually and exactly. It was also stated that the initial inputs are audio files containing completely separated streams: pure bass, pure percussion, pure vocals and pure instrumentation. However, in order to conduct a more realistic and useful experiment, we use three different levels of stream separation. We define these three levels as follows:

- **Full separation or 'pure' stream** (complete isolation of streams from one another and no cross-mixing of signals, such that an instrument at full separation will have no contamination from other instruments);
- **Partial separation or 'impure' stream** (which we define as having one stream at full signal level whilst other streams are mixed in at 25% of full signal level; for example, a partially separated vocal stream contains vocals at full volume, and then drums, bass and instrumentation at 25% of full volume); and
- **No separation or 'control' case** (all streams at full volume mixed together, just like the original song).

See Figure 3.2 for an illustration of these three levels. Here each of the four colours represents one stream. It is evident that the 'no separation' case will be identical for each stream as it merely involves an equal mix of every stream.

Notice that rather than merely comparing the base case (no separation) and the theoretical best case (full separation), we also analyse streams at partial separation. This allows a far more practical and useful experiment in that it also allows *imperfect* stream separation to be tested as a chord estimation technique; given that any algorithmic stream separation method is bound to achieve an imperfect result (a fact easily demonstrated by the application of any conventional noise reduction technique on noisy audio), this renders the results far more relevant and practical.

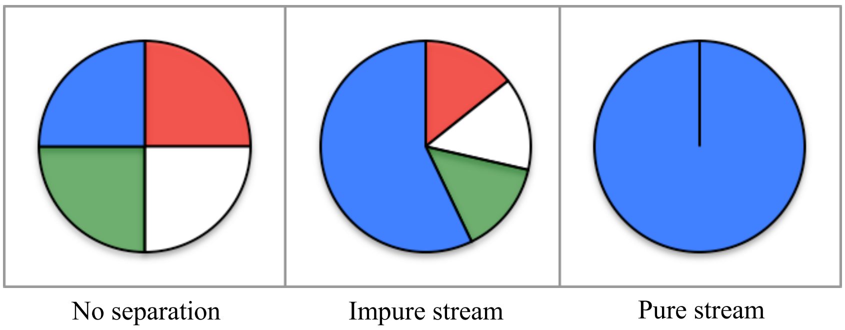

Figure 3.2: Relative signal levels of streams for different separation amounts

This step of the process, whereby we form streams at various separation levels, is represented in Figure 3.4 as a 'signal mixer' stage, which takes the completely separated inputs and mixes them to some extent before they are passed to the harmonic analysis stage.

3.2.4 Harmonic Analysis

As noted in Section 3.1, the 'harmonic analysis' referred to here describes only one specific stage in the overall process of chord estimation. If we abstract this process as a 'black box' as it is represented in Figure 3.1, we can see that it accepts harmonic song components at its input, and outputs estimated chords for these components. Hence the process used in our experiment must behave in the same manner: it will be presented with harmonic song components and will be required to output the estimated chords of these components. The method by which we determine these estimated chords (i.e. 'most likely chord detection') and the format of the inputs and outputs are discussed in more detail in Section 3.3.4 under the detailed design of the experiment.

There is one significant difference, however, between the model of Section 3.1 and the method developed here. The former involved being presented *simultaneously* with each harmonic component of the song (for example, the vocals, piano and bass guitar) and analysing these all at once. In creating an experiment, however, we may (and must) dispense with this requirement as we are explicitly testing whether analysing audio streams *separately* improves chord recognition performance. Instead, we perform a separate harmonic analysis on each separated stream. The main ramification of this is that, rather than one single output containing overall estimated chord information, there are several: there would be four, one per stream (as discussed in Section 3.2.1), except that we discard the drums as they do not contain any relevant harmonic information. Hence we have three separate harmonic analyses. This is obviously different to our original conceptual model, as the number of outputs has tripled; however, this is quite deliberate, as this now allows us to compare individually each output and test which of them, if any, yield better chord estimation performance than the original model.

Figure 3.3 shows a comparison of the black-box harmonic analysis processes from the original model and the one developed here. Note that whilst our original model assumed the brain dealt with the streams at once, the experimental model handles each mix separately. Note also that we assume the brain uses a rhythmic analysis to assist its chord detection, whereas this is handled in the experiment by manually separating the streams; hence no input from the drums is shown in the model for the latter.

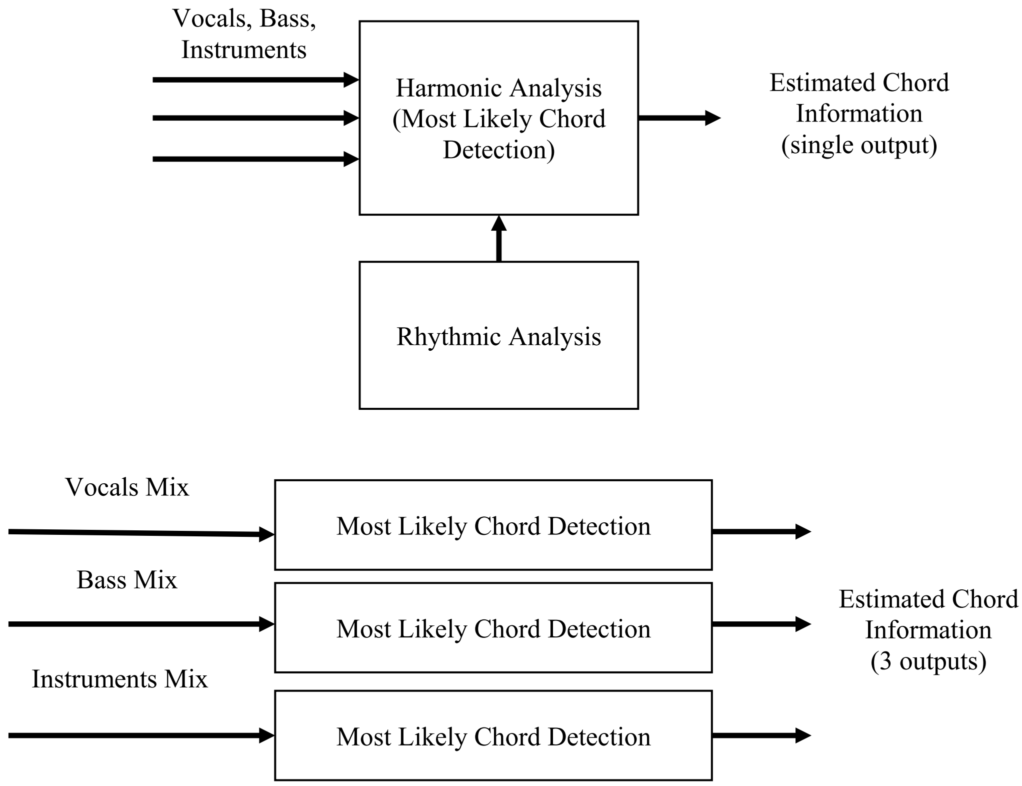

Figure 3.3: Black-box comparison of original (top) and new (bottom) harmonic analyses

3.2.5 Summary of Concept Design

Taking into account these necessary modifications to the process, we arrive at our final conceptual design for the experiment. Figure 3.4 shows a pictorial representation of this system. Note that this diagram shows the three mix cases being analysed for one stream; this seems contradictory to Figure 3.3 which shows three streams being analysed for one mix case. However, remember that we are testing each mix case (no separation, partial separation, full separation) for each stream (instruments, bass, vocals), and so both diagrams are correct but only show a subset of the full process for the sake of conserving space.

It should be noted at this juncture that, even if our assumption that the brain separates audio streams were proven to be incorrect, it would have no bearing on the validity of our hypothesis (namely, that undertaking the same process with an algorithm would increase its chord estimation ability): whilst this conjecture may be grounded in psychoacoustics, it stands regardless as an independent engineering problem. As such, we need not be overly concerned with the exact process whereby the brain analyses music to estimate chords. Our focus on this process is merely because we believe it is a superior model to those currently employed, and not necessarily because it is a theoretically perfect system.

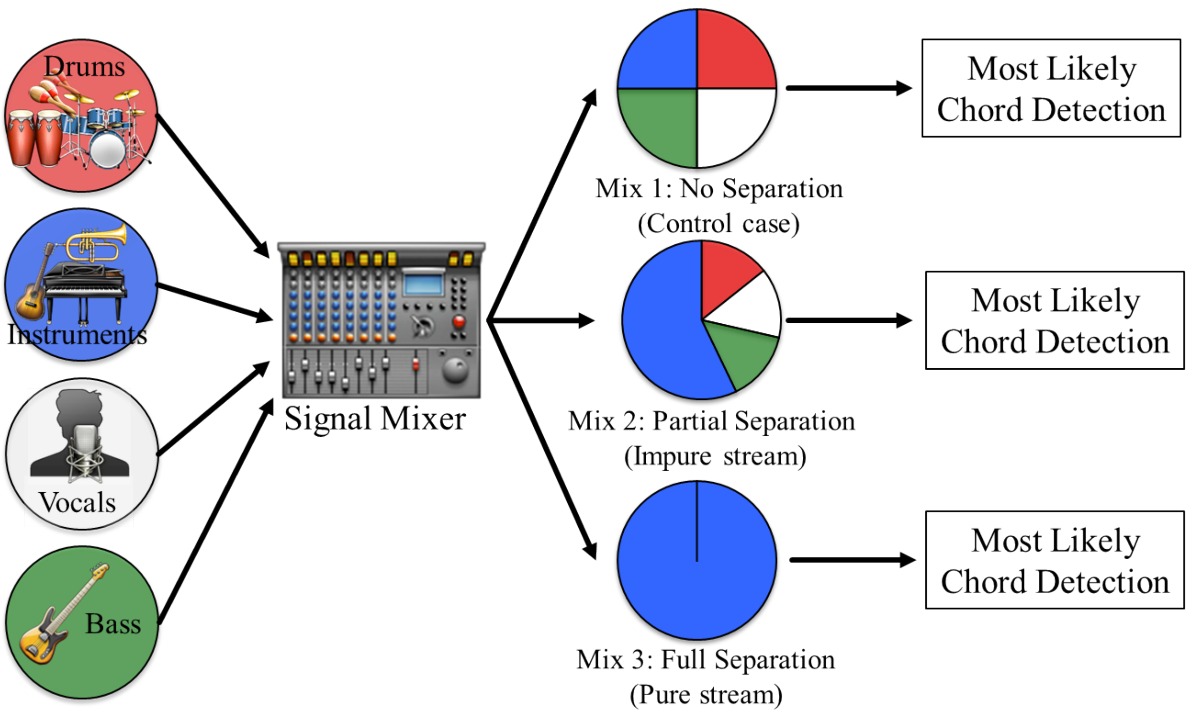

Figure 3.4: Conceptual design of experiment for one single stream (to be repeated for each)

Having therefore constructed a conceptual design for our experiment, we proceed to develop the specific mechanisms through which we realise this design.

3.3 Detailed Design

In this section we present the detailed design of our experiment. Our aim here is to be able to accurately describe the process by which our results are obtained such that future researchers might be able to replicate it. We describe below each stage of the process given in Section 3.2 separately.

3.3.1 Input

The first stage of our process is the input to the system. This section deals with all considerations of the format and content of the input in order to achieve the most useful and accurate result from this experiment.

3.3.1.1 File Format

The first such consideration we make is that of the file format used for the experiment. It is obvious that the natural audition process works on real audio in the form of sound waves to the ear; however, for a machine-based process, this is not possible. We instead use digital representations of the song in order that computational methods may be employed, and consequently it is imperative that an adequate format is chosen before we consider how the experiment will be conducted. There are a host of audio file formats capable of encoding audio, but for our purposes, we require a format that faithfully represents the spectral content of the song. Many popular audio encoding formats (e.g. MPEG-3, AAC, WMA etc.) are ‘lossy formats’, meaning that lossy data compression is employed to reduce file size, causing information to be discarded. For audio codecs such as mp3, this is usually a loss in high-frequency information: often barely audible to the human ear, but a noticeable distortion to a computer. (This is a consequence of the Nyquist-Shannon sampling theorem, which holds that to reproduce a signal containing spectral components up to frequencies of x Hz, the signal must be sampled at $2x$ Hz [43]; lower sampling frequencies, therefore, such as those found in compressed audio, lose information about high-frequency content and must ‘guess’ the lost information when reconstructing the file.)

To demonstrate the effect of lossy compression on a file’s spectral content, two audio files containing the song “Chelsea” by the artist Summertime’s End were generated. One was a lossless WAVE (.wav) file; the other was a compressed mp3 (.mp3) file. Each file was stereo (having a left and right audio channel), had a sampling rate of 44.1 kHz (that is, contained 44,100 samples per second of the real audio waveform) and was taken at 16-bit resolution (meaning that there were 2^{16} or 65,536 possible amplitude levels that each sample could take). The WAVE codec is an uncompressed format, meaning that no data compression is employed. Consequently, the .wav file was a completely accurate computerised representation of the song (to the stated accuracy of 16 bits of audio resolution and a sampling rate of 44.1 kHz). However, the mp3 file had a stereo bit rate of 128 kbps, corresponding to a compression ratio of roughly 11:1, which involved some loss of information[7]. The mp3 file was exported as a WAVE file[8] in Logic Pro, a professional digital audio workstation, and then both files were loaded into MATLAB to construct spectrograms representing their frequency content. Figure 3.5 and Figure 3.6 show the resultant plots. Note the clear degradation present in frequencies above 16 kHz in the mp3 file: whereas the WAVE file exhibits a smooth, uninterrupted representation of frequencies up to 22 kHz, the spectrogram of the mp3 file shows a noticeable difference in frequency representation above a certain frequency. This artefact is the aforementioned consequence of compression resulting in a loss of high-frequency content. For this reason, all files used in this experiment are non-transcoded and are in the WAVE format to ensure the highest possible fidelity to the true song information.

---

[7] 16-bit stereo audio at 44.1 kHz has a bitrate of 16 bits × 2 channels × 44,100 Hz = 1,411,200 bps. 1,411,200 / 128,000 = 11.025:1 compression rate.

[8] This step does not invalidate the experiment as the resolution of WAVE exceeds that of mp3. A WAVE file can therefore reproduce an mp3 file without loss.

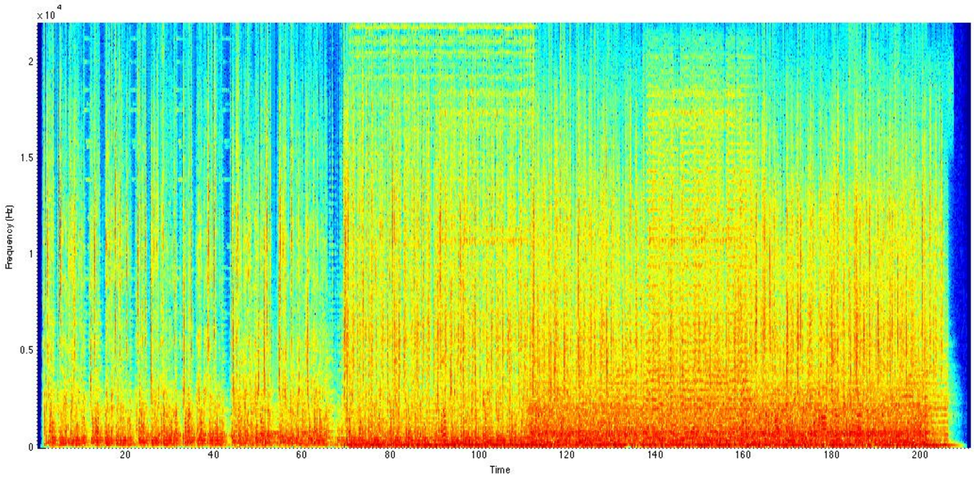

Figure 3.5: Spectrogram of uncompressed WAVE file

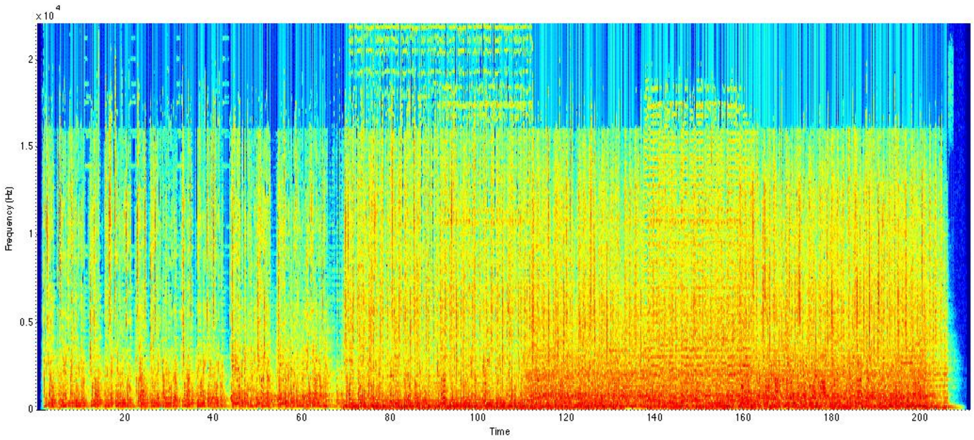

Figure 3.6: Spectrogram of lossy compressed mp3 file

3.3.1.2 Song Selection

The next consideration that must be made for the input is deciding which songs to use for the experiment. It was discussed in the introduction that this thesis focuses mainly on pop music; therefore it is desirable to focus on songs that represent the pop genre well. There is a large constraint, however, in the requirement to have audio stems available for the songs we select (see Section 3.2.1): as Uhle et al. note in their work, such files are not readily available [29]. The songs we select must satisfy both of these criteria. We therefore choose four pop songs: "Chelsea" and "Natalie Grey" by Summertime's End, "Good Time" by Owl City and "Tread The Water" by Malachi Joshua. Of primary importance is that there are stem files available to us for each of these songs, allowing us to conduct our experiment upon them. Furthermore, between them, these songs feature a range of common pop instrumentation (synthesisers, pianos, acoustic and electric guitars, drums and strings); exhibit typical pop song structure, with each making strong use of the verse-chorus-bridge based format; run between 3:15 and

3:45 minutes in length; and use common chord progressions sourced from a homogeneous key (except for a single case in "Natalie Grey"). It might be recalled that these factors were mentioned in Section 2.1.4 in the discussion of which characteristics define pop music; likewise, both Owl City and Summertime's End are listed on the iTunes Store as "Pop" artists (Malachi Joshua does not appear in the store). Table 3.1 details some of the genre-relevant details of each song. Other factors that can assist in determining whether a song could be considered pop, such as the melodic form and lyrics, are fairly irrelevant for a chord detection method that does not take into account melodic form and cannot understand lyrics. Therefore, taking into account the characteristics we have discussed about these three songs, it can be seen that they are a good representation of the pop music genre.

Song	Chelsea	Natalie Grey	Good Time	Tread The Water
Length	3:17	3:18	3:28	3:36
Instruments	Lead & harmony vocals, electric guitar, acoustic guitar, piano, strings, drum kit, bass guitar	Lead & harmony vocals, electric guitar, piano, synthesisers, drum kit, synthesised bass	Lead & harmony vocals, synthesisers, synthesised bass, electronic drums	Lead & harmony vocals, acoustic guitar, strings, piano, bass guitar, drum kit
Key	E (no change)	D# (one change)	G# (no change)	C (no change)
Structure	Verse-Chorus-Verse-Chorus-Bridge-Chorus	Verse-Chorus-Verse-Chorus-Bridge-Chorus	Verse-Chorus-Verse-Chorus-Bridge-Chorus	Verse-Chorus-Verse-Chorus-Bridge-Verse

Table 3.1: Genre-relevant song details for selected songs

3.3.1.3 Stream Definition

Having selected songs to use as well as having determined an appropriate file format for the experiment, we now discuss the actual input that we use. We stated in Section 3.2.1 that in order to conduct a fair and useful experiment, rather than using whole songs as the input to the input of the process, 'stems' of these songs must instead be used. It was further noted that four such stems are used for the purposes of this experiment: drums & percussion, bass, vocals, and other instruments. There is no 'standard' definition for what each of these categories refers to, as our distinction between four groups is somewhat arbitrary, and so we provide our suggested working definitions:

- **Drums and percussion** refer to all non-pitched or percussive instrumentation whose main purpose in the song is to provide a sense of rhythm.
- **Bass** refers to the monophonic instrument sitting lowest in the frequency register of the song, whose function is to (usually) indicate the fundamental of the root chord and to 'fill out' the lower register, which is rarely occupied by other harmonic instrumentation.
- **Vocals** are the sounds produced by the human voice. In the sense we use it, however, we refer to the lead vocals, as in pop music these are usually the main focal point of the song; they are the vocal line (or lines) that usually carry the lyrics and the main melody of the track.
- **Other instrumentation** means all instrumentation remaining after the previous three groups have been extracted; in pop music, this can range from typical instrumentation such as acoustic or electric guitar and piano to instruments such as orchestral string arrangements and synthesisers.

We therefore begin with stem files corresponding to these four distinctions; one for each of the streams listed above. (By the definitions we have used, if these four stem files are played simultaneously, they will sound as the original song would[9].) These stem files are in .wav format to preserve the most information about the song and yield the most accurate results. However, these files contain many individual chords, and so they require further processing so that each chord instance can be analysed individually. This requires the next step of our process: the rhythmic analysis.

3.3.2 Rhythmic Analysis

Rather than to actually perform a rhythmic analysis – which, as we noted, would involve a more detailed consideration of the beat and rhythm of the song – the function of this stage in the experiment is to achieve the same results that a real rhythmic analysis would. The justification for doing so was outlined in Section 3.2.2. This step, then, is relatively simple. Samples from the four songs mentioned in Section 3.3.1 are taken and divided manually at each point at which the chords change. This yields a collection of audio files containing one single chord each. These files are the actual input used for the process; rather than feeding in

[9] The stem files will not sound *exactly* like the final production; in professional recordings, effects are applied in the mixing and mastering processes after the stems have been assembled. For example, some commonly applied effects are equalisation (the application of filters to adjust the spectral characteristics of the sound) and dynamic range compression (the automatic adjustment of volume levels to make loud parts relatively quieter), both of which would slightly change the sound of the final master.

only four unseparated files (one for each stream) per song, we use four files for *each* chord in the song. In all, we take 62 chord samples from between the four songs: 18 from Chelsea, 16 each from Good Time and Natalie Grey, and 12 from Tread The Water. These are chosen such that each chord progression in these songs is fully represented through verse, bridge and chorus, but without picking any chord progression more than once so as to have a fair representation of chords.

3.3.3 Signal Mixing

After the stream files are divided according to chordal content, they are passed through a simulation of the signal mixing stage of Section 3.2.3, which involved mixing in varied amounts of other streams into the stream in question. We achieve this by again utilising Logic Pro to perform the signal mixing. The four files corresponding to individual streams for a given chord are taken and duplicated so that we have three copies of each. One copy is left untreated so that it contains only one stream; this is the 'complete separation' mix (full stream isolation). The second copy is left at full volume whilst the other three streams are mixed in at 25% of full power (that is, a 12 dB reduction in signal voltage level is applied[10], significantly reducing the signal but still retaining a very audible 'footprint'), simulating the results of real-life audio separation. This is the 'partial separation' mix. Finally, a 'control' or 'no separation' mix is created, in which each of the other three streams is mixed in with equal power to the first (such that the file is merely a section of the original song).

Consequently, after this stage, we have three different sets of files or 'mixes', divided by stream and by chord. At this point, we discard the files corresponding to the rhythmic (drum) stream: they are no longer useful to us as only a harmonic analysis is to follow and these files contain no useful harmonic information. We are therefore left with three kinds of mixes – complete, partial and no separation – divided into three streams (vocals, bass and other instrumentation, with the drums having been discarded) and with one chord per file. Since the 'no separation' case is equivalent for each stream (as bass with drums, vocals & instruments is equivalent to drums with bass, vocals & instruments, and so on) we have only one such mix for each chord and not three like the other cases; this means that there are seven, not nine, stream combinations for each chord. These are: no separation; 'impure' and 'pure' vocals; 'impure' and 'pure' instruments; and 'impure' and 'pure' bass. (Figure 3.7 illustrates the various proportions of streams in these seven 'mix cases' used in this

[10] According to the formula $L = 10\log_{10}\left(\frac{P}{P_0}\right)$, where L denotes the decibel ratio between power levels P and P_0, and given that $P \propto V^2$, we can equivalently state that $L = 10\log_{10}\left(\frac{V}{V_0}\right)^2 = 20\log_{10}(\frac{V}{V_0})$. Consequently one quarter of original power corresponds to a voltage decibel reduction of approximately $20\log_{10}(0.25) = -12.04\text{ dB} \approx -12\text{ dB}$.

experiment, using the same stream colour scheme as that of Figure 3.4.) As such, since we have 62 chords for analysis, there are 434 .wav files analysed in total.

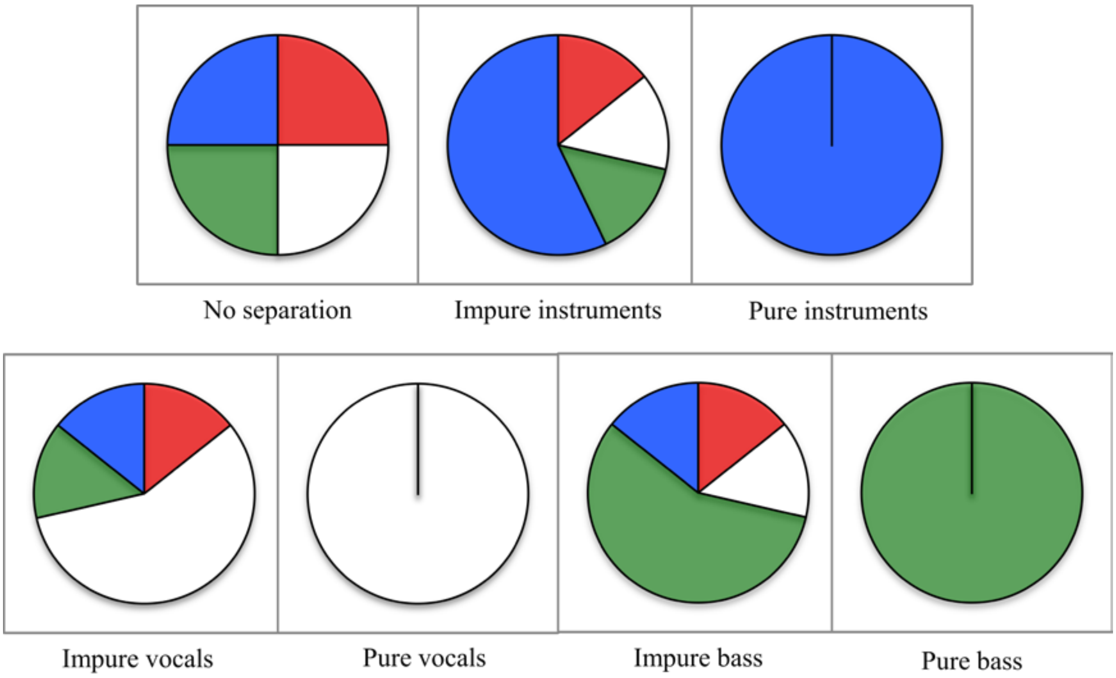

Figure 3.7: A graphical representation of stream proportions in the seven 'mix cases'

3.3.4 Harmonic Analysis

The harmonic analysis is the final step in both the theoretical method and the one designed here, and is perhaps the most difficult to emulate accurately since the method by which the brain recognises chords from audio is, as discussed, not yet understood. In terms of the scientific method, however, using a consistent technique for the process of harmonic analysis throughout our experiment is more important to obtaining a meaningful result than is absolute fidelity to the actual process. (This is not to say that a grossly inaccurate chord estimation method can be employed so long as it is not changed between data sets; it is merely that a process need not be totally perfect in order to return useful results. After all, as any untrained listener attempting to identify a chord progression can attest to, the human brain is known to be imperfect too.)

Using this guiding principle, then, we eschew more complex, contextual methods of harmonic analysis, and opt for one which is simpler and which lends itself more readily to computerised implementation. The method we employ is the same used by Fujishima in his pioneering work on PCP vectors, as referenced in Section 2.2. The steps of this process are outlined from Sections 3.3.4.1 to 3.3.4.3 below for a given audio signal (assumed to be a single chord).

3.3.4.1 Forming the Chroma Vector

First, the audio signal's energy is decomposed into a spectral representation using the discrete Fourier transform (DFT). This means that the spectrum X_k of the signal x_n is calculated according to

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\left(\frac{j2\pi}{N}kn\right)}, k = 0, \dots, N-1$$

where N is the number of samples present in one audio 'frame'. After this representation is obtained, we perform spectral wrapping to condense this chroma data into a single octave by grouping all power spectrum coefficients by their closest pitch class (note). This wrapping is done by defining the metric *P(i)* for the ith sample of the signal as

$$P(i) = \mathrm{round}\left(12\log_2\left(\frac{f_s}{f_{ref}}.\frac{i}{N}\right)\right) \mathrm{mod}\ 12$$

where f_s is the sample frequency and f_{ref} is the reference frequency of the pitch class we wish to define as the first pitch class of the chromagram (in this case, that of the note C). The log operation in this step ensures that the pitch classes we classify to are equally logarithmically spaced, like the 12 pitch classes themselves (recall the definition of notes from Section 2.1.1). The 'mod 12' operation ensures that this metric is cyclic every 12 pitch classes. Therefore *P(i)* returns an integer from 0 to 11 representing the closest of the 12 pitch classes (in our case, from C through to B respectively) for the given sample. The chroma vector ***c*** is then populated element-wise by the operation

$$\boldsymbol{c}(p) = \sum_{P(i)=p} |X_k|^2, p = 0, 1, \dots, 11.$$

Since pitch classes are equivalent across octaves (i.e. a doubling of frequency), these steps discard the pitch height (see Section 2.1.1) of each spectral peak by effectively equating $2^n f \equiv f$ for integer n. For example, spectral peaks at 441 Hz and 879 Hz would both be classified to the 'A' pitch class as they are closest to the A notes at 440 Hz and 880 Hz respectively. The chroma vector is hence a 12-dimensional column vector containing elements that correspond to the signal power of each pitch class in the audio signal. The formation of this vector is finalised by dividing through by the value of the maximum element to normalise it.

3.3.4.2 Chord Template Matrix

Next we define a chord template matrix $\boldsymbol{T}$. First we represent (consistent with our previous notation) the notes from C through B as integers from 0 to 11. We then create a binary 'bit mask' for each chord by representing tones present in the chord with a '1' and tones not present as a '0'. Hence, for example, an F minor chord, comprised of the notes F, G# and C, would have ones in elements 1, 6 and 9 and zeroes elsewhere. Our intention here is to form a matrix composed of these chord 'bit masks' as columns. This begs the question of which chords to include in such a matrix; as in Section 2.1.2, we only consider the twenty-four main major and minor triads common in pop music. (Bello and Pickens, who take a similar approach in forming a chord template matrix, select the same chords [23].) As such, the resultant matrix $\boldsymbol{T}$ is a 12×24 binary array, with each row representing the chords C major through B major and then C minor through B minor, and each column representing the presence of the notes C through B in each of these chords. Table 3.2 shows an excerpt of this matrix, with the chords from C major to F minor shown (omitting F# minor to B minor).

Chord	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm
C	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
C#	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
D	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0
D#	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0
E	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0
F	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1
F#	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
G	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0
G#	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1
A	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
A#	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0
B	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0

Table 3.2: Chord template matrix excerpt

3.3.4.3 Selection of Most Likely Chord

Having formed both the chroma vector and the chord template matrix, the selection of the most likely chord is then performed. We take the inner product[11] of the chroma vector $\boldsymbol{c}$ and the chord template matrix $\boldsymbol{T}$ to achieve $\boldsymbol{l}$, the array of chord likelihoods:

$$\boldsymbol{l} = \boldsymbol{c} \cdot \boldsymbol{T} = \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{T}$$

$\boldsymbol{l}$ is a 1×24 vector containing a likelihood score for each potential chord in the given audio signal. The reasoning behind this is simple: elements of the chroma vector that do not correspond to potential chord tones are multiplied by zeroes in the template matrix $\boldsymbol{T}$, whereas elements that correspond to chord tones are multiplied by ones. Hence triads whose notes most strongly match those present in the chroma vector will achieve higher element-wise products and thus a larger likelihood score.

The process of determining the most likely chord from $\boldsymbol{l}$ is then the trivial act of selecting the maximum element from $\boldsymbol{l}$ and determining the chord from this element's index: for example, if the 12th element was the largest, the most likely chord would be B major, the 12th column in $\boldsymbol{T}$. This is the final step in the harmonic extraction process used here.

3.3.4.4 Assessing Accuracy

To complete the analysis of the audio, then, the results from the previous step are taken and compared to manual transcriptions of the songs involved in this experiment. We use three metrics to measure the accuracy of chord recognition, based on the discussion of chord relationships in Section 2.1.2:

- **Exact Identification**. This metric is simply determined by the percentage of chords that are correctly identified.
- **Relative Identification**. This metric measures not only chords that are identified exactly, but also deems correct the relative major/minor chords of the correct answer; for example, if the correct chord was G major, this metric would treat not only G major but also E minor, its relative minor, as a correct identification.
- **Parallel Identification**. Similarly to the relative identification metric, this metric deems parallel major/minor chords to be equivalent, so that, say, either F

[11] The inner product as relating to matrices is the equivalent of the vector dot product. It is formed by taking the product of a row vector a on the left with a column vector b (or in this case, an array of column vectors B) on the right such that the i^{th} element of the resultant array is $\sum a_k B_{ki}$.

major or F minor would be considered a correct identification if the actual chord were F minor.

It is naturally expected that the 'exact identification' metric will score the lowest as it is the most restrictive. Note that since relative and parallel identification count erroneous classifications as correct, they are not technically correct measures of accuracy as is exact identification; however, they are useful in that they assist in illustrating where classification errors occur.

3.3.4.5 Practical Implementation

In this experiment, the steps detailed in Sections 3.3.4.1 to 3.3.4.3 are implemented in MATLAB. Extraction of chroma data from audio files is performed by the function 'mirchromagram.m' from the MATLAB function set MIRtoolbox, an "integrated set of functions written in MATLAB, dedicated to the extraction from audio files of musical features" [44]. To facilitate this operation on hundreds of files, and to collect and analyse the data (including the matrix multiplications and maximum-likelihood detection above), original MATLAB functions have been created to interface with this program and store the results in a useable format. These functions appear in the Appendix in Section 8.1. The resultant likelihood data is exported to Microsoft Excel for analysis.

3.4 Hypothesis

Having therefore described the practical details of our experiment, we may once more restate our hypothesis. We hypothesise that the results of the chord estimation technique described above will reflect a higher accuracy in chord recognition when using stream separation than when not using it. To be specific, we expect the instruments (and most of all, the pure stream) to exhibit better 'exact' chord recognition performance than the overall mix due to the removal of interference from other streams. We also expect that the bass (again, particularly the pure stream) will achieve better 'parallel' chord recognition performance than the control case. The reason for this is that as the bass is a single tone, it does not carry the information necessary to determine the tonality of the chord and so it is likely that many parallel chord confusion errors will be made; however, given that the bass mostly plays chord tones, it should reflect the true chord more than other streams.

The results in the next chapter will allow us to test the validity of this hypothesis.

4. RESULTS

This chapter presents the results obtained from the experiment designed in Section 3, along with brief commentary upon notable features of the data. In-depth analysis is reserved for the next chapter. For the sake of brevity, the full tables of chord likelihood data are not reproduced in this chapter, but appear in the Appendix in Section 8.2. Results are given in a specific format, illustrated below:

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	2.1371	'A'	1

Table 4.1: Format of results

The 'Predicted Chord' column's entries are selected according to the maximum-likelihood chord detection rule described in Section 3.3.4.3, and represent the chord with the highest row-wise likelihood score from ***l***. The actual chord is compared to this predicted chord value, and a binary correctness score assigned: '1' for a match and '0' for a mismatch. The percentage of correctly identified chords (according to the 'exact identification' metric given in Section 3.3.4.4) is given at the bottom of each table.

4.1 Control Case: 'No Separation' Mixes

Below are the results achieved for the 'control' case for each song: that is, the mixes with no stream separation (i.e. simply unaltered excerpts of the original song). These are the results against which we measure the performance of chord estimation on the other stream-separated mixes, as they represent a 'base case' of sorts without stream separation.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood	Actual Chord	Correct?
1	'A'	2.1371	'A'	1	**10**	'C#m'	1.6226	'C#m'	1
2	'A'	1.2617	'E'	0	**11**	'Bm'	1.5729	'B'	0
3	'B'	1.9799	'B'	1	**12**	'C#m'	1.751	'C#m'	1
4	'C#m'	1.5046	'C#m'	1	**13**	'D'	1.7487	'A'	0
5	'C#m'	1.5767	'C#m'	1	**14**	'E'	1.5879	'E'	1
6	'C'	1.1338	'E'	0	**15**	'Bm'	1.941	'B'	0
7	'A'	2.1363	'A'	1	**16**	'C#m'	1.7045	'C#m'	1
8	'Bm'	2.0414	'B'	0	**17**	'Bm'	1.5473	'B'	0
9	'D'	2.2214	'F#m'	0	**18**	'Dm'	1.872	'A'	0
								% Correct:	50.00%

Table 4.2: Chord recognition performance for 'Chelsea', no separation mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	1.6691	'C#'	1	**9**	'F#m'	1.4347	'C#'	0
2	'G#'	1.8541	'G#'	1	**10**	'C#'	1.6932	'G#'	0
3	'D#'	2.4869	'D#'	1	**11**	'D#'	2.3899	'D#'	1
4	'D#'	1.9611	'Fm'	0	**12**	'Fm'	1.9929	'Fm'	1
5	'F#'	2.0696	'C#'	0	**13**	'C#'	2.0299	'C#'	1
6	'C#'	1.4692	'G#'	0	**14**	'D#'	2.1925	'D#'	1
7	'D#'	2.0765	'D#'	1	**15**	'C#'	1.7481	'C#'	1
8	'A#'	1.9903	'Fm'	0	**16**	'C#m'	1.4768	'C#m'	1
								% Correct:	62.50%

Table 4.3: Chord recognition performance for ‘Natalie Grey’, no separation mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'G#m'	2.0716	'G#'	0	**9**	'G#m'	2.1819	'G#'	0
2	'D#'	2.8786	'D#'	1	**10**	'D#'	2.7814	'D#'	1
3	'A#'	2.57	'A#'	1	**11**	'A#'	2.4245	'A#'	1
4	'Cm'	2.7943	'Cm'	1	**12**	'Cm'	2.9071	'Cm'	1
5	'G#m'	2.5485	'G#'	0	**13**	'Fm'	1.3883	'G#'	0
6	'D#'	2.3899	'D#'	1	**14**	'G#'	1.5758	'D#'	0
7	'A#'	2.7997	'A#'	1	**15**	'A#'	1.263	'A#'	1
8	'G#'	2.8543	'Cm'	0	**16**	'Cm'	2.1714	'Cm'	1
								% Correct:	62.50%

Table 4.4: Chord recognition performance for ‘Good Time’, no separation mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'F'	1.3384	'F'	1	**7**	'F'	1.1447	'F'	1
2	'G'	1.3219	'G'	1	**8**	'Am'	1.5993	'C'	0
3	'Am'	1.8732	'Am'	1	**9**	'F'	1.6966	'F'	1
4	'F'	2.4433	'C'	0	**10**	'G'	1.3616	'G'	1
5	'Am'	1.932	'Am'	1	**11**	'F'	1.5207	'Am'	0
6	'Bm'	1.4608	'G'	0	**12**	'Am'	2.2918	'C'	0
								% Correct:	58.33%

Table 4.5: Chord recognition performance for ‘Tread The Water’, no separation mix

The control case achieved overall chord recognition accuracy of 36 out of 62 chords exactly recognised, or 58.06%. The results for each of our three success metrics are tabulated in Table 4.6. Here the relative and parallel identification metrics do not significantly outperform the exact identification metric, with accuracy rates of 62.90% and 67.74% respectively: less than 10% more than the exact identification metric.

Metric	# Correctly Identified	Accuracy (%)
Exact Identification	36	58.06
Relative Identification	39	62.90
Parallel Identification	42	67.74

Table 4.6: Overall chord recognition performance for ‘no separation’ mixes

4.2 ‘Pure Instrument’ Mixes

Below are the results achieved for the ‘pure instrument’ case for each song: that is, the mixes with only instruments left in, and no vocals, bass or drums.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	1.9286	'A'	1	**10**	'C#'	1.8355	'C#m'	0
2	'E'	2.1868	'E'	1	**11**	'B'	2.3251	'B'	1
3	'B'	1.6978	'B'	1	**12**	'G#m'	1.6847	'C#m'	0
4	'C#m'	2.4029	'C#m'	1	**13**	'A'	2.012	'A'	1
5	'C#m'	1.9428	'C#m'	1	**14**	'E'	1.5788	'E'	1
6	'E'	1.5959	'E'	1	**15**	'B'	2.5993	'B'	1
7	'A'	2.4323	'A'	1	**16**	'C#'	2.1267	'C#m'	0
8	'B'	2.3928	'B'	1	**17**	'B'	2.4826	'B'	1
9	'F#m'	1.2536	'F#m'	1	**18**	'A'	2.3072	'A'	1
								% Correct:	83.33%

Table 4.7: Chord recognition performance for ‘Chelsea’, pure instrument mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	2.2018	'C#'	1	**9**	'C#'	1.4732	'C#'	1
2	'G#'	1.4783	'G#'	1	**10**	'G#'	1.7659	'G#'	1
3	'D#'	1.8962	'D#'	1	**11**	'D#'	1.5384	'D#'	1
4	'F'	1.8879	'Fm'	0	**12**	'Fm'	1.9553	'Fm'	1
5	'C#'	1.5238	'C#'	1	**13**	'C#'	1.5651	'C#'	1
6	'G#'	2.2677	'G#'	1	**14**	'D#'	2.2014	'D#'	1
7	'G#'	1.7531	'D#'	0	**15**	'C#'	1.995	'C#'	1
8	'Fm'	1.6864	'Fm'	1	**16**	'C#m'	2.0948	'C#m'	1
								% Correct:	87.50%

Table 4.8: Chord recognition performance for ‘Natalie Grey’, pure instrument mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'G#'	1.9315	'G#'	1	9	'G#'	1.908	'G#'	1
2	'D#'	2.2141	'D#'	1	10	'D#'	2.7921	'D#'	1
3	'A#'	1.342	'A#'	1	11	'A#'	1.7585	'A#'	1
4	'Cm'	1.3733	'Cm'	1	12	'Cm'	1.8997	'Cm'	1
5	'G#'	1.745	'G#'	1	13	'G#'	1.7384	'G#'	1
6	'D#'	1.7881	'D#'	1	14	'D#'	1.9492	'D#'	1
7	'A#'	1.6676	'A#'	1	15	'A#'	1.4032	'A#'	1
8	'Cm'	1.8309	'Cm'	1	16	'Cm'	1.4407	'Cm'	1
								% Correct:	100.00%

Table 4.9: Chord recognition performance for 'Good Time', pure instrument mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'F'	2.0723	'F'	1	7	'F'	1.4466	'F'	1
2	'G'	1.7446	'G'	1	8	'C'	2.3482	'C'	1
3	'Am'	1.6199	'Am'	1	9	'F'	1.9719	'F'	1
4	'C'	2.0709	'C'	1	10	'G'	1.82	'G'	1
5	'Am'	2.0302	'Am'	1	11	'Am'	2.3001	'Am'	1
6	'G'	2.0561	'G'	1	12	'C'	2.229	'C'	1
								% Correct:	100.00%

Table 4.10: Chord recognition performance for 'Tread The Water', pure instrument mix

This case achieved overall chord recognition accuracy of 57 out of 62 chords exactly recognised, or 91.94%. The results for each success metric are tabulated in Table 4.11. Note that here, no relative identification errors were made, since relative identification accuracy was also 91.94%. The parallel identification metric was not significantly different at 96.77%; however, these results are all very accurate and strongly outperform the base case.

Metric	# Correctly Identified	Accuracy (%)
Exact Identification	57	91.94
Relative Identification	57	91.94
Parallel Identification	60	96.77

Table 4.11: Overall chord recognition performance for 'pure instrument' mixes

4.3 'Impure Instrument' Mixes

Below are the results achieved for the 'impure instrument' case for each song: that is, the mixes with primarily instruments left in, with vocals, bass and drums reduced to 25% of their original signal power.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	2.0461	'A'	1	10	'C#m'	1.8367	'C#m'	1
2	'E'	1.6186	'E'	1	11	'B'	1.6746	'B'	1
3	'B'	1.684	'B'	1	12	'C#m'	1.3641	'C#m'	1
4	'C#m'	2.2182	'C#m'	1	13	'A'	1.8576	'A'	1
5	'C#m'	2.0925	'C#m'	1	14	'E'	1.4822	'E'	1
6	'E'	1.3252	'E'	1	15	'B'	2.1074	'B'	1
7	'A'	2.362	'A'	1	16	'C#m'	1.7459	'C#m'	1
8	'B'	2.0778	'B'	1	17	'B'	1.801	'B'	1
9	'F#m'	1.3942	'F#m'	1	18	'A'	2.1753	'A'	1
								% Correct:	100.00%

Table 4.12: Chord recognition performance for ‘Chelsea’, impure instrument mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	1.9519	'C#'	1	9	'C#'	1.4756	'C#'	1
2	'G#'	1.7406	'G#'	1	10	'G#'	1.6706	'G#'	1
3	'D#'	2.5016	'D#'	1	11	'D#'	1.9466	'D#'	1
4	'Fm'	1.8203	'Fm'	1	12	'C#'	1.5144	'Fm'	0
5	'C#'	1.6837	'C#'	1	13	'C#'	1.9901	'C#'	1
6	'G#'	1.7624	'G#'	1	14	'D#'	1.9293	'D#'	1
7	'D#'	1.8667	'D#'	1	15	'C#'	1.9337	'C#'	1
8	'Fm'	1.6177	'Fm'	1	16	'C#m'	1.7683	'C#m'	1
								% Correct:	93.75%

Table 4.13: Chord recognition performance for ‘Natalie Grey’, impure instrument mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'G#'	2.0918	'G#'	1	9	'G#'	1.8582	'G#'	1
2	'D#'	2.781	'D#'	1	10	'D#'	2.9061	'D#'	1
3	'A#'	2.3494	'A#'	1	11	'A#'	2.3344	'A#'	1
4	'Cm'	2.6779	'Cm'	1	12	'Cm'	2.6545	'Cm'	1
5	'G#'	2.2464	'G#'	1	13	'G#'	1.3221	'G#'	1
6	'D#'	2.57	'D#'	1	14	'D#'	2.0232	'D#'	1
7	'A#'	2.1164	'A#'	1	15	'A#'	1.2864	'A#'	1
8	'Cm'	2.5351	'Cm'	1	16	'Cm'	2.3424	'Cm'	1
								% Correct:	100.00%

Table 4.14: Chord recognition performance for ‘Good Time’, impure instrument mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'F'	1.7701	'F'	1	**7**	'F'	1.3238	'F'	1
2	'G'	2.4039	'G'	1	**8**	'C'	2.0869	'C'	1
3	'Am'	2.1597	'Am'	1	**9**	'F'	1.6825	'F'	1
4	'C'	2.4861	'C'	1	**10**	'G'	1.6495	'G'	1
5	'Am'	2.0222	'Am'	1	**11**	'Am'	1.9962	'Am'	1
6	'G'	1.7118	'G'	1	**12**	'C'	2.2862	'C'	1
								% Correct:	100.00%

Table 4.15: Chord recognition performance for 'Tread The Water', impure instrument mix

This case achieved overall chord recognition accuracy of 61 out of 62 chords exactly recognised, or 98.39%. The results for each success metric are tabulated in Table 4.16. It is evident here that chord recognition success was excellent with this set of mixes, since only one error was made in classification for all metrics. This error occurred by selecting C# when the actual chord was Fm. It should be noted that in this case the likelihood score for Fm was 1.474, only a 2.7% difference from the value of 1.514 for the selected C# chord. Furthermore, the chords Fm and C# differ by only one note (C/C#), and hence the mistake – whilst not accounted for in any of our success metrics – is comparatively minor. Consequently we conclude that this case significantly outperforms the control case.

Metric	# Correctly Identified	Accuracy (%)
Exact Identification	61	98.39
Relative Identification	61	98.39
Parallel Identification	61	98.39

Table 4.16: Overall chord recognition performance for 'impure instrument' mixes

4.4 'Pure Vocals' Mixes

Below are the results achieved for the 'pure vocals' case for each song: that is, the mixes with only vocals left in, and no instruments, bass or drums.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'E'	1.2635	'A'	0	10	'C#m'	1.4694	'C#m'	1
2	'D'	1.4635	'E'	0	11	'G#m'	1.7421	'B'	0
3	'F#'	1.4402	'B'	0	12	'B'	1.3314	'C#m'	0
4	'E'	2.3066	'C#m'	0	13	'B'	1.7454	'A'	0
5	'C#m'	1.6819	'C#m'	1	14	'G#m'	1.941	'E'	0
6	'B'	1.5654	'E'	0	15	'Bm'	1.3171	'B'	0
7	'E'	1.388	'A'	0	16	'B'	1.7864	'C#m'	0
8	'F#'	1.0782	'B'	0	17	'G#m'	1.5864	'B'	0
9	'F#m'	1.6698	'F#m'	1	18	'B'	1.5274	'A'	0
								% Correct:	16.67%

Table 4.17: Chord recognition performance for ‘Chelsea’, pure vocals mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	1.3379	'C#'	1	9	'C#'	1.1166	'C#'	1
2	'Am'	1.6838	'G#'	0	10	'A#m'	1.2934	'G#'	0
3	'Fm'	1.6141	'D#'	0	11	'Gm'	1.5561	'D#'	0
4	'G#m'	1.3536	'Fm'	0	12	'Fm'	1.5717	'Fm'	1
5	'A#m'	1.0941	'C#'	0	13	'C#'	2.2016	'C#'	1
6	'A#m'	1.439	'G#'	0	14	'D#'	2.1576	'D#'	1
7	'Fm'	1.6043	'D#'	0	15	'C#'	1.3227	'C#'	1
8	'F'	1.2149	'Fm'	0	16	'C#m'	1.5749	'C#m'	1
								% Correct:	43.75%

Table 4.18: Chord recognition performance for ‘Natalie Grey’, pure vocals mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'Fm'	1.9206	'G#'	0	9	'Gm'	1.8177	'G#'	0
2	'D#'	2.2249	'D#'	1	10	'A#'	1.7447	'D#'	0
3	'F'	1.3528	'A#'	0	11	'A#'	1.6163	'A#'	1
4	'Cm'	2.0058	'Cm'	1	12	'D#'	1.6398	'Cm'	0
5	'Gm'	1.9573	'G#'	0	13	'G#'	1.1542	'G#'	1
6	'D#'	1.3499	'D#'	1	14	'Em'	2.1104	'D#'	0
7	'Dm'	1.7141	'A#'	0	15	'F'	1.51	'A#'	0
8	'D#'	1.5321	'Cm'	0	16	'G#'	1.8577	'Cm'	0
								% Correct:	31.25%

Table 4.19: Chord recognition performance for ‘Good Time’, pure vocals mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	1.653	'F'	0	7	'C#'	1	'F'	0
2	'C'	1.7009	'G'	0	8	'Bm'	1.8669	'C'	0
3	'Am'	2.0612	'Am'	1	9	'C'	1.7333	'F'	0
4	'Gm'	2.3871	'C'	0	10	'D'	1.2902	'G'	0
5	'E'	1.3373	'Am'	0	11	'Bm'	1.4684	'Am'	0
6	'Dm'	1.58	'G'	0	12	'C'	1.6682	'C'	1
								% Correct:	16.67%

Table 4.20: Chord recognition performance for 'Tread The Water', pure vocals mix

This case achieved overall chord recognition accuracy of 17 out of 62 chords exactly recognised, or 27.42%. The results for each success metric are tabulated in Table 4.21. Relative and parallel identification scored at 38.71% and 29.03% accuracy respectively, a moderate increase for the relative identification case. Recognition performance for this case was very poor, generally around half that of the control case.

Metric	# Correctly Identified	Accuracy (%)
Exact Identification	17	27.42
Relative Identification	24	38.71
Parallel Identification	18	29.03

Table 4.21: Overall chord recognition performance for 'pure vocals' mixes

4.5 'Impure Vocals' Mixes

Below are the results achieved for the 'impure vocals' case for each song: that is, the mixes with primarily vocals left in, with instruments, bass and drums reduced to 25% of their original signal power.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	1.6337	'A'	1	10	'C#m'	1.9647	'C#m'	1
2	'Em'	1.2672	'E'	0	11	'G#m'	2.1737	'B'	0
3	'B'	2.2444	'B'	1	12	'C#m'	1.4855	'C#m'	1
4	'C#m'	2.4711	'C#m'	1	13	'B'	1.8249	'A'	0
5	'C#m'	1.4937	'C#m'	1	14	'E'	2.1899	'E'	1
6	'B'	1.2405	'E'	0	15	'Bm'	2.1605	'B'	0
7	'F#m'	1.8854	'A'	0	16	'C#m'	1.6655	'C#m'	1
8	'Bm'	1.7989	'B'	0	17	'G#m'	1.4632	'B'	0
9	'D'	2.3917	'F#m'	0	18	'B'	1.6932	'A'	0
								% Correct:	44.44%

Table 4.22: Chord recognition performance for 'Chelsea', impure vocals mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	1.5264	'C#'	1	**9**	'C#'	1.1488	'C#'	1
2	'C'	2.1366	'G#'	0	**10**	'Fm'	1.9989	'G#'	0
3	'G#'	1.9057	'D#'	0	**11**	'D#'	2.2161	'D#'	1
4	'C#'	1.8929	'Fm'	0	**12**	'C#'	1.5731	'Fm'	0
5	'F#'	1.877	'C#'	0	**13**	'C#'	2.0504	'C#'	1
6	'C#'	2.067	'G#'	0	**14**	'D#'	2.1583	'D#'	1
7	'G#'	2.1598	'D#'	0	**15**	'C#'	1.3342	'C#'	1
8	'A#'	1.6924	'Fm'	0	**16**	'C#m'	1.5358	'C#m'	1
								% Correct:	43.75%

Table 4.23: Chord recognition performance for 'Natalie Grey', impure vocals mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'Fm'	2.1966	'G#'	0	**9**	'Gm'	1.6664	'G#'	0
2	'D#'	2.6075	'D#'	1	**10**	'Cm'	2.7954	'D#'	0
3	'A#'	2.0365	'A#'	1	**11**	'A#'	1.9921	'A#'	1
4	'Cm'	2.3514	'Cm'	1	**12**	'Cm'	2.3673	'Cm'	1
5	'D#'	2.445	'G#'	0	**13**	'G#'	1.1078	'G#'	1
6	'D#'	1.7348	'D#'	1	**14**	'D#'	2.6301	'D#'	1
7	'Dm'	2.0217	'A#'	0	**15**	'A#'	1.9852	'A#'	1
8	'D#'	1.6024	'Cm'	0	**16**	'Cm'	2.1734	'Cm'	1
								% Correct:	62.50%

Table 4.24: Chord recognition performance for 'Good Time', impure vocals mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	1.6178	'F'	0	**7**	'F'	1.024	'F'	1
2	'C'	1.9177	'G'	0	**8**	'Am'	1.8648	'C'	0
3	'Am'	2.0598	'Am'	1	**9**	'C'	1.7768	'F'	0
4	'Dm'	2.5224	'C'	0	**10**	'G'	1.4078	'G'	1
5	'E'	1.3435	'Am'	0	**11**	'Bm'	1.4622	'Am'	0
6	'Bm'	1.7413	'G'	0	**12**	'C'	1.8582	'C'	1
								% Correct:	33.33%

Table 4.25: Chord recognition performance for 'Tread The Water', impure vocals mix

This case achieved overall chord recognition accuracy of 29 out of 62 chords exactly recognised, or 46.77%. The results for each success metric are tabulated in Table 4.26. Relative and parallel identification scored at 59.68% and 51.61% accuracy respectively, again indicating a slight improvement in performance with the relative identification metric. Overall recognition performance for this case was poor, being worse than the control case; however, these results are comparatively better than the 'pure vocals' case, implying that the presence of drums, bass and instrumentation improved performance somewhat.

Metric	# Correctly Identified	Accuracy (%)
Exact Identification	29	46.77
Relative Identification	37	59.68
Parallel Identification	32	51.61

Table 4.26: Overall chord recognition performance for ‘impure vocals’ mixes

4.6 ‘Pure Bass’ Mixes

Below are the results achieved for the ‘pure bass’ case for each song: that is, the mixes with only bass left in, and no vocals, instruments or drums.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	1.1831	'A'	1	10	'C#'	1.0547	'C#m'	0
2	'E'	1.0827	'E'	1	11	'B'	1.1338	'B'	1
3	'B'	1.0827	'B'	1	12	'C#'	1.0939	'C#m'	0
4	'C#'	1.1103	'C#m'	0	13	'A'	1.1373	'A'	1
5	'C#'	1.0019	'C#m'	0	14	'E'	1.1202	'E'	1
6	'E'	1.0171	'E'	1	15	'B'	1.107	'B'	1
7	'A'	1.1147	'A'	1	16	'C#'	1.0991	'C#m'	0
8	'B'	1.0774	'B'	1	17	'B'	1.0893	'B'	1
9	'F#'	1.2384	'F#m'	0	18	'A'	1.12	'A'	1
								% Correct:	66.67%

Table 4.27: Chord recognition performance for ‘Chelsea’, pure bass mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	1.0872	'C#'	1	9	'C#'	1.0891	'C#'	1
2	'G#'	1.0518	'G#'	1	10	'G#'	1.0511	'G#'	1
3	'D#'	1.0711	'D#'	1	11	'D#'	1.0626	'D#'	1
4	'F'	1.0631	'Fm'	0	12	'F'	1.0719	'Fm'	0
5	'C#'	1.089	'C#'	1	13	'C#'	1.1009	'C#'	1
6	'G#'	1.0528	'G#'	1	14	'D#'	1.0426	'D#'	1
7	'D#'	1.0722	'D#'	1	15	'C#'	1.0522	'C#'	1
8	'F'	1.065	'Fm'	0	16	'C#'	1.0422	'C#m'	0
								% Correct:	75.00%

Table 4.28: Chord recognition performance for ‘Natalie Grey’, pure bass mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'G#'	1.0974	'G#'	1	**9**	'G#'	1.0424	'G#'	1
2	'D#'	1.1799	'D#'	1	**10**	'D#m'	1.1944	'D#'	0
3	'A#'	1.2121	'A#'	1	**11**	'A#'	1.2284	'A#'	1
4	'C'	2.0939	'Cm'	0	**12**	'Fm'	1.9104	'Cm'	0
5	'F#'	1.968	'G#'	0	**13**	'C#'	1.1133	'G#'	0
6	'C'	0.99999	'D#'	0	**14**	'Cm'	1.1182	'D#'	0
7	'A#'	1.4481	'A#'	1	**15**	'D#'	1	'A#'	0
8	'C#'	1.1063	'Cm'	0	**16**	'Cm'	1.8971	'Cm'	1
								% Correct:	43.75%

Table 4.29: Chord recognition performance for 'Good Time', pure bass mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'F'	1.0754	'F'	1	**7**	'F'	1.0962	'F'	1
2	'C'	1	'G'	0	**8**	'C'	1.0523	'C'	1
3	'A'	1.0362	'Am'	0	**9**	'F'	1.0548	'F'	1
4	'F'	1.9025	'C'	0	**10**	'C'	1	'G'	0
5	'A'	1.1419	'Am'	0	**11**	'D'	1	'Am'	0
6	'B'	1.1295	'G'	0	**12**	'C'	1.7776	'C'	1
								% Correct:	41.67%

Table 4.30: Chord recognition performance for 'Tread The Water', pure bass mix

This case achieved overall chord recognition accuracy of 36 out of 62 chords exactly recognised, or 58.06%. The results for each success metric are tabulated in Table 4.31. Note that whilst the relative identification performance was only slightly better at 59.68%, the parallel identification metric scored 80.65% recognition – a significant increase (22.59%) over the base case.

Metric	# Correctly Identified	Accuracy (%)
Exact Identification	36	58.06
Relative Identification	37	59.68
Parallel Identification	50	80.65

Table 4.31: Overall chord recognition performance for 'pure bass' mixes

4.7 'Impure Bass' Mixes

Below are the results achieved for the 'impure bass' case for each song: that is, the mixes with primarily bass left in, with vocals, instruments and drums reduced to 25% of their original signal power.

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'A'	1.1774	'A'	1	**10**	'C#'	1.0607	'C#m'	0
2	'E'	1.0768	'E'	1	**11**	'B'	1.1218	'B'	1
3	'B'	1.1203	'B'	1	**12**	'C#m'	1.0855	'C#m'	1
4	'C#'	1.1122	'C#m'	0	**13**	'A'	1.1152	'A'	1
5	'C#'	1.0126	'C#m'	0	**14**	'E'	1.1093	'E'	1
6	'E'	1.0183	'E'	1	**15**	'B'	1.1407	'B'	1
7	'A'	1.1281	'A'	1	**16**	'C#'	1.1074	'C#m'	0
8	'B'	1.0744	'B'	1	**17**	'B'	1.0878	'B'	1
9	'F#'	1.2168	'F#m'	0	**18**	'A'	1.1586	'A'	1
								% Correct:	72.22%

Table 4.32: Chord recognition performance for 'Chelsea', impure bass mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'C#'	1.148	'C#'	1	**9**	'C#'	1.0829	'C#'	1
2	'G#'	1.0686	'G#'	1	**10**	'G#'	1.0628	'G#'	1
3	'D#'	1.0862	'D#'	1	**11**	'D#'	1.0982	'D#'	1
4	'F'	1.0693	'Fm'	0	**12**	'A#m'	1.1906	'Fm'	0
5	'C#'	1.0885	'C#'	1	**13**	'C#'	1.4965	'C#'	1
6	'G#'	1.0494	'G#'	1	**14**	'D#'	1.2117	'D#'	1
7	'D#'	1.0741	'D#'	1	**15**	'C#'	1.0456	'C#'	1
8	'F'	1.0736	'Fm'	0	**16**	'C#m'	1.2873	'C#m'	1
								% Correct:	81.25%

Table 4.33: Chord recognition performance for 'Natalie Grey', impure bass mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'G#m'	1.2688	'G#'	0	**9**	'G#m'	1.2331	'G#'	0
2	'D#'	1.5588	'D#'	1	**10**	'D#'	1.5406	'D#'	1
3	'A#'	1.466	'A#'	1	**11**	'A#'	1.7079	'A#'	1
4	'Cm'	2.5545	'Cm'	1	**12**	'Fm'	2.5206	'Cm'	0
5	'G#m'	2.5484	'G#'	0	**13**	'C#'	1.1025	'G#'	0
6	'D#'	2.3899	'D#'	1	**14**	'Cm'	1.1473	'D#'	0
7	'A#'	2.7997	'A#'	1	**15**	'A#'	1.018	'A#'	1
8	'G#'	2.8543	'Cm'	0	**16**	'Cm'	1.9503	'Cm'	1
								% Correct:	56.25%

Table 4.34: Chord recognition performance for 'Good Time', impure bass mix

#	Predicted Chord	Likelihood Score	Actual Chord	Correct?	#	Predicted Chord	Likelihood Score	Actual Chord	Correct?
1	'F'	1.0347	'F'	1	**7**	'F'	1.0955	'F'	1
2	'G'	1.0063	'G'	1	**8**	'C'	1.0614	'C'	1
3	'A'	1.0232	'Am'	0	**9**	'F'	1.0739	'F'	1
4	'F'	1.9274	'C'	0	**10**	'C'	1	'G'	0
5	'A'	1.1242	'Am'	0	**11**	'A'	1.0023	'Am'	0
6	'Bm'	1.1271	'G'	0	**12**	'C'	1.856	'C'	1
								% Correct:	50.00%

Table 4.35: Chord recognition performance for 'Tread The Water', impure bass mix

This case achieved overall chord recognition accuracy of 41 out of 62 chords exactly recognised, or 66.13%. The results for each success metric are tabulated in Table 4.36. Note that, as in the pure bass case, the relative identification metric was similar to the exact identification metric (67.74%) but the parallel identification metric strongly outperformed both at 87.10%, a 20.97% increase over the base case.

Metric	**# Correctly Identified**	**Accuracy (%)**
Exact Identification	41	66.13
Relative Identification	42	67.74
Parallel Identification	54	87.10

Table 4.36: Overall chord recognition performance for 'impure bass' mixes

4.8 Summary of Results

In total, 434 files were analysed and tested against the three chord recognition metrics. A graphical representation of the chord recognition performance for each case and for each of the four songs tested is shown in Figure 4.1 as measured by the exact identification metric. This plot exhibits several interesting features.

It can be seen that the vocals are a poor predictor of the chord: this is evident in that in only one case did the vocals (pure or impure) match the control case performance – for the song 'Good Time' – and underperformed relative to the control case in all other instances. Bass was a more effective predictor of chord, but was inconsistent: the pure bass case exhibited a large 33.33% variation in effectiveness between the four songs tested, and for some songs scored higher than the no separation case whilst scoring lower in others. Most striking, however, is the success of the instruments stream in predicting chord. Both the pure and impure instruments streams outperformed the control case in every instance, and achieved perfect chord recognition over the sample set multiple times. This indicates that the instruments are the best stream-wise predictor of the chord. Interestingly, in all cases, impure streams performed better than or as well as their pure counterparts.

To illustrate whether relative or parallel chord confusion (i.e. classifying chords as their relative or parallel majors/minors) is significant in any case, the chord recognition success for each stream as measured by each of the three chord recognition success metrics is given in Figure 4.2. Here it can be seen that relative chord confusion accounts for a moderate proportion of errors only in the vocals mixes, where the relative identification metric outperforms the exact identification one between ten and fifteen per cent. Since the vocals are evidently the worst stream for predicting the chord, however, this is not a significant result. More useful, as we have already identified in this chapter, is the strong performance of the bass for chord recognition when parallel majors and minors are treated equally. This is indicated by the parallel identification metric scoring much higher than the exact identification metric in the bass cases. In each of these cases, the resultant chord recognition performance is better than the control case – a result that could allow for improvement upon chord transcription performance. We explore this prospect in Section 5.

4.9 Conclusion

In this chapter we have presented the results of the experiment designed in Section 3. The raw chord identification data were tabulated and then presented graphically to assist identification of relevant trends. We also provided some commentary upon the most prevalent features of these data. In the next section, we deal much more thoroughly with the results of this experiment, and identify not only the possible factors influencing them, but also their significance and consequences both for this project and beyond it.

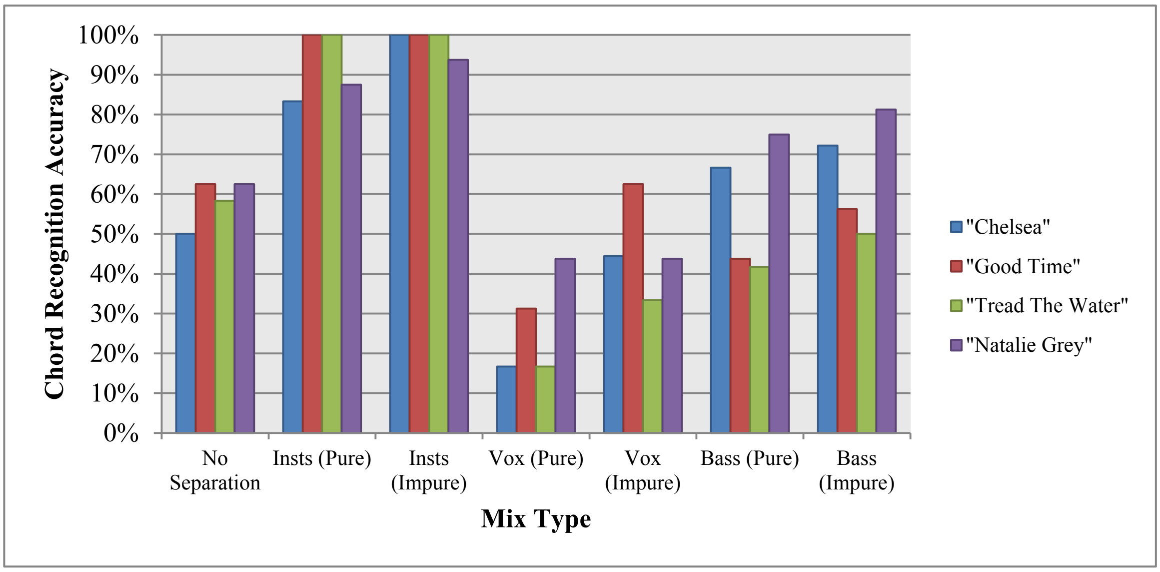

Figure 4.1: Exact chord recognition accuracy for different mixes by song

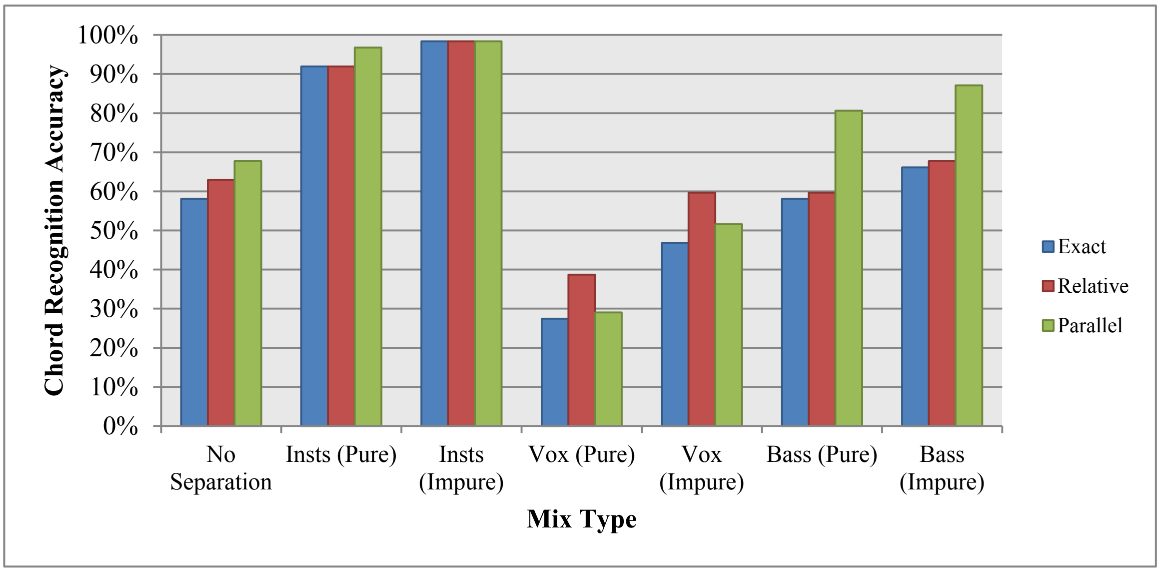

Figure 4.2: Chord recognition accuracy for different mixes by metric

5. DISCUSSION

In this section we analyse the results presented in the previous chapter in more depth. We comment upon features and trends present in the data and their importance and probable causes. We then discuss the significance of our results and its applications for the field of automatic chord extraction and outside it. We conclude the chapter by considering the various limitations of our study and the assumptions we have made, and defending these where appropriate.

5.1 Analysis of Results

We open the discussion of our results by here presenting a detailed analysis of them. The outcomes of the vocals, bass and instrument cases are considered and commented upon separately, and then general comment is offered upon items not specific to any of these.

5.1.1 Vocals

The vocals in this experiment were the poorest performing of all the harmonic streams. The results show only one instance in which the vocals did not underperform the base case – for the impure vocals mix in the song “Good Time” – and even in this instance the vocals only matched the base case performance, rather than beating it. Because of this, the vocals are of little use in a direct implementation of chord extraction. This is not to say they are useless in any chord extraction technique, though: analysis of the vocals could assist in applications such as key detection, since a vocal line will almost always be constrained to the key of the song. The key is useful information for contextual methods of chord transcription, and so extraction of vocals is not necessarily a fruitless endeavour.

The main factor behind the poor performance of the vocals in predicting the chord is probably that unlike the instrumentation, the vocals in a song are rarely limited to chord tones only. One of the features of melody is that it is not constrained to the three basic notes that make up the chord playing at any given time, but rather can play any note within the current key (and often can even break the key). Furthermore, notes in a vocal melody usually change far more rapidly than the chords in the song do, and so there may be several different notes that are sung during one chord. This naturally makes it difficult for a chord estimation method to determine the current chord by considering the only notes of the vocal melody.

As an aside, it can be seen that the impure vocals case classified more chords correctly than the pure vocals case. A simple explanation for this is perhaps that since each of the other harmonic streams was a better predictor of the chord, their addition increased chord recognition performance.

5.1.2 Bass

We noted at the end of the previous chapter that the bass scored much more highly on the parallel identification metric than on the exact identification metric: a 22.59% increase in the pure case and 20.97% in the impure case. This means that over one-fifth of the chords in either case were classified incorrectly by tonality (major or minor quality): that is, the identification method selected, say, B minor instead of B major or vice versa. Considering the relatively small increase in other cases, this is a significant proportion of the sample set. In Section 3.4 one of our hypotheses was that the parallel identification performance of the bass would outperform the exact identification performance of the control case; this therefore indeed proved to be true. (What did not prove to be true was the prediction that the pure stream would perform better than the impure stream, as Figure 4.2 shows. We discuss this issue shortly.)

The reason that the bass performs so much better in parallel identification than in exact identification is most likely that it is a single tone. Basses rarely, if ever, play chords, and instead usually play a monophonic line that heavily features the root and fifth of the chord. Because of this, the third – which, it may be recalled from Section 2.1.2, determines whether the chord is major or minor – is seldom played in the bass, and therefore the bass stream on its own is much more likely to make a parallel classification error since it does not have this information. We find support for this notion in that the impure bass stream performed better than the pure bass stream against all metrics: the presence of the vocals and the instruments in these cases would have supplied the mix with information about the chord tonality, boosting its performance.

Parallel classification errors are more easily corrected than relative classification errors, because a parallel major or minor will not be in the home key of the song, whereas a relative major or minor will. For example, consider the chord D# major in a song also in D# major: its *relative* minor, C minor, is also part of this key, but its *parallel* minor, D# minor, is not. A chord detection method would therefore easily be able to identify D# minor as an erroneous detection if it had knowledge of the current key of the piece, but could make no such decision about the C minor if it were detected. Because of this, the parallel classification errors made by the bass cases would likely prove to be only minor issues in a practical system.

One major issue with the bass, however, is its inconsistency in performance. Over the four songs tested, it was noted that the bass exhibited a 33.33% swing in its chord prediction ability, and both bass mixes underperformed the control case for the songs “Tread The Water” and “Good Time”. It is not inherently obvious what caused this discrepancy as there is no clear difference between the bass streams from these songs and those from the other two songs, in terms of playing style or instrumentation: “Chelsea” also uses an electric bass like both of these songs, but it suffers no such inaccuracy. As a basis for a chord detection method, this inconsistency would be unacceptable, and so further investigation is required into developing more reliable chord detection from the bass before it is used in a real implementation.

5.1.3 Instruments

The instruments stream, as we noted in the last chapter, was the best predictor of the chord in this experiment, achieving 98.39% and 91.94% exact detection accuracy for the impure and pure cases respectively. Compared to the ‘no separation’ case, which scored 58.06% for the same metric, these are remarkable increases, at 40.33% and 33.88% respectively. The magnitude of this result validates the second of our hypotheses: that the instruments would exhibit better exact chord recognition performance than the base case. This was indeed true over every instance tested. Again, the prediction that the pure case would perform better was incorrect, as the impure case actually achieved better results. However, both the impure and pure cases outperformed the control case significantly.

The results from the instruments cases were also less variable than any others except for the no separation case: across the four songs tested, the pure instruments case exhibited a variation of 16.67%, and the impure case a marginal 6.25%. This consistency would be a welcome feature to automatic chord transcription methods using this stream. Furthermore, because even the lowest recognition score outperformed the best from the base case, this inconsistency is hardly important: even the instruments in their worst case provided a better estimate of the chord than the overall song in the best case.

The fact that the impure case outperformed the pure case is somewhat enigmatic, and seems counter-intuitive: if the instruments are the best predictor of the chord, why do we achieve better performance when they are ‘diluted’ by other streams? Indeed, even if the bass and vocals had had no effect upon chord recognition performance, at least the introduction of the drums into the mix should have had a detrimental effect upon the chord recognition performance due to its wideband inharmonic noise. That this is not the case is puzzling indeed, for it implies that the addition of the bass and vocals was so beneficial as to not only counteract the effect of the drums, but also to further improve the performance. Our best guess is that the presence of the bass boosted the power levels of the root note sufficiently that it enabled some erroneous chord detections to be corrected.

There is a pleasing and useful caveat, however, to the fact that the impure case performed better in both recognition score and consistency. No noise reduction technique from audio is known to be perfect, and signal extraction invariably leaves behind traces of the signal to be extracted. We may therefore confidently assume that any practical implementation of stream separation would leave traces of the bass, vocals and drums in the instrument stream – and from our experiment we see that this would actually have a favourable effect upon the performance of such a method! This is an unexpected benefit arising from the results we have gathered, and may prove useful to others attempting a practical implementation.

5.1.4 General Comments

It could be remarked, by cursorily considering only the result that the instruments were the best predictor of the chord, that this endeavour has been somewhat trivial in nature: is it not obvious that the instruments, which are used to create the chords in a song, should prove the best predictor of them? Such a judgment would, however, be highly inaccurate for a number of reasons. Primarily, it is quite unscientific to claim that a result like this is obvious and not requiring empirical justification. There are numerous potential factors that could have resulted in the instruments not proving the best predictor of the chord. What if the impression of a ‘chord’ that listeners formed when listening to a song was created not merely by a subset of the instrumentation, but by some complex interplay between all the streams that formed the music? Had we discovered that none of the individual streams on their own predicted the chord more accurately than the base case, this would have been our conclusion. We have also been careful to note that our results apply to our assumed model of human perception, and do not necessarily apply to human perception itself: we could possibly have found that, with our method, the base case was a better predictor of chord than any individual stream, and if this were so then it may have been that our model was in fact incorrect. Our results are therefore useful in showing some alternative theories to be false.

Furthermore, the results are not as obvious as expected. Taking the naïve viewpoint of the previous paragraph would have led us to conclude that the pure instruments stream would have performed better than the impure one; as we saw, this was not so. We also discovered that the bass exhibited useful chord prediction performance in some cases in which it outperformed the base case. Such findings easily dispel the notion that our results are in any way obvious or trivial, or were not worth investigating.

One of the conclusions we can draw from the data, then, is that the hypotheses offered earlier in this dissertation have proven mostly correct. As hypothesised, the bass offered better relative chord prediction performance than the base case, and the instruments offered better exact chord prediction performance. The impure streams performed better than the pure streams in each case, which was counter to what was originally predicted: however, this was a minor outcome, and as was noted above, proved to be a very useful one in the instruments case.

A practical consideration to make is that, whilst the bass may offer a worse prediction of the chord compared to the instruments, the extraction of bass from audio is a much simpler problem than the ‘extraction’ of instruments. This follows simply from the fact that ‘extraction’ of the instruments is actually achieved by the removal of everything that is *not* the instruments, meaning the vocals, bass and drums. Obviously, extracting the bass alone is simpler, more efficient and less computationally intensive than extracting it and two other streams. Practical methods for automatic chord transcription must therefore account for this and decide whether the cost of separating the instruments stream is worth the performance improvement it yields.

5.1.5 Conclusion

Having shown our hypotheses to be fundamentally true, we may therefore state that this project has been successful in that it has achieved its original aim (as in Section 1.3) to assess the effect of stream separation upon automatic chord transcription, since we have shown that it is indeed possible to improve automatic chord estimation performance by separating streams. In doing so, we have laid the groundwork for future efforts in this area toward developing more effective and more efficient methods of transcribing chords from musical audio. We elaborate upon possible future directions for this work in Section 6.2.

5.2 Significance & Applications

Our project has significant ramifications not only in the area of automatic chord estimation, but beyond this field also. The primary application of our results, however, is in relation to this project and the problem of transcribing chords from audio. Stream separation has not previously been applied to chord extraction, and hence this is a pioneering work in this regard. The outcomes of our experiment demonstrate that it is a valid and useful technique for improving chord transcription methods, and also provide insight into how best to make use of stream separation to improve chord recognition performance. We have also enabled any future implementations that use stream separation to effectively utilise the information it yields, such as the success of the bass in parallel chord detection.

Another consequence of our study relates to our theoretical motivation for undertaking this project (see Section 1.2), well beyond the scope of merely music information retrieval: namely, that our success or failure in demonstrating the effectiveness of stream separation to assist chord recognition would provide insight into the processes behind human perception. By demonstrating that separating streams assisted chord transcription performance even using a crude method of chord recognition – that is to say, one which does not use musical context at all, as the brain would – we can state that it is highly likely that the brain avails itself of its ability to separate streams when detecting chords in music. In other words, the effectiveness of stream separation in achieving near-perfect chord recognition performance in some cases is good evidence that the brain uses a similar process to achieve similar results.

Finally, the fact that in one case we were able to achieve such excellent chord transcription performance (over 98%) hints at an exciting prospect: perhaps stream separation may one day allow for automatic methods of chord transcription to match or even exceed the performance of the human brain at recognising chords. It is probably not fair to claim that we outperformed the brain here, given the relatively small sample set of 62 chords: however, as computing capabilities increase with faster and more powerful processors, and as new and more complex methods of chord extraction are developed, the integration of stream separation into these may well prove to be a crucial element in the creation of an automatic method that can transcribe chords better than humans.

5.3 Assumptions & Limitations

As useful as our results may seem, various assumptions have been made in order to achieve them, and furthermore there are various ways in which our method falls short of the actual human case it aims to imitate or is otherwise lacking. Here we acknowledge these assumptions and limitations, and where applicable, explain why they are reasonable.

5.3.1 'Control Case' Misleading

We are careful to highlight that the 'no separation' case in our experiment does *not* reflect the accuracy of human perception, lest its label as the 'control case' should imply otherwise. It is the control case for this method of chord estimation only: that is, the evaluation of a matrix-vector product and the subsequent selection of the highest likelihood. The assumption inherent in this is that the incorporation of more complex techniques of chord estimation – say, incorporating musical theory or models of musical context (see Section 2.2) – would improve all our cases of stream separation equally; therefore there is no benefit in using a more complicated method than that described in Section 3.3.4 since the relative difference between cases would not significantly be affected. This is one of the likeliest criticisms that could be levelled at this experiment, and so this explanation is of great importance.

5.3.2 Stream Separation Not Implemented

It is also significant to note that this project has not attempted the automatic separation of streams from musical audio (although we dealt briefly with prior attempts at achieving it in Section 2.4); we have merely simulated this technique. It could well be argued, as we have, that this allowed for a fairer and more accurate experiment; however, the fact remains that automatic stream separation has neither been performed nor developed here. Consequently it must be noted that the results presented herein cannot be claimed as a sound demonstration that stream separation is an effective chord recognition technique, as the process has not been actually used here. Nevertheless, the results here strongly indicate that stream separation, if implemented automatically, would indeed improve chord transcription performance, and so this limitation is mainly technical, rather than a tangible barrier to the effectiveness of stream separation.

5.3.3 Only Single Streams Considered

In conducting our experiment upon chord transcription performance, we assumed that the best results offered by stream separation would occur from trying to single out one stream alone. This is manifest in that the stream separation cases considered were classed by individual stream – e.g. 'pure instruments' or 'impure bass' – and not by any combination, such as 'instruments and bass'. However, the fact that the impure instrument mix achieved better results than the pure instrument mix suggests that perhaps the presence of other streams can assist accuracy. The possibility of using different combinations of streams – or perhaps even different features from each stream – has not been investigated. This is a shortfall of the research conducted here, and one that we suggest as future work in Section 6.2.2.

5.3.4 Noncontiguous Streams

Another of the inherent assumptions of our experiment was that in each chord segment each stream would be present. In practice, this is not the case: songs frequently have segments where drums, vocals, bass, or certain instruments are omitted, usually for musical effect. In such cases, a stream separation technique might become confused, and this was not simulated in our experiment since each of the audio files used was manually chosen to have all streams. We therefore note that in some sense, the results we achieved are perhaps a best case for performance for stream separation, unless compensation for this issue is implemented. This is also suggested as future work in Section 6.2.3.

5.3.5 Limited Sample Size

With a sample set of 62 chord segments, in the best case it would only be possible to analyse all of the 24 major/minor triads a minimum of twice each. However, since we only used four songs in this experiment, we drew chords from only four keys and therefore there were some chords we did not analyse at all. The issue here is not the lack of chords being tested: in theory, since chords are invariant under pitch translation, one chord should be representative of all others with negligible differences. In actuality, the problem is the small number of songs from which these chords were taken. Drawing our 62 chords from, say, 20 songs would have yielded a much better representation of the chord population than taking them from four songs, as inherently every song will have different musical qualities: different vocalists, different instrumentation, different tempo and so on. Unfortunately, as described in Section 3.3.1, this experiment has been constrained to songs for which stem files (i.e. files corresponding to each individual stream) are available; without access to a large database of stem files, it is difficult to see how this challenge could be overcome. Nonetheless, to theoretically achieve more useful results, a similar experiment should utilise a sample size much larger than 62 chords, from more than four songs and from more than three artists.

5.3.6 Lack of Musical Knowledge

Admittedly, comparison to the human brain is not a fair comparison for our method, since the human brain has many advantages. One is that the brain of a trained human transcriber (and even that of the casual listener) has years of experience in recognising instruments such as drums or vocals in audio. Such training means that the task of stream separation is far easier for humans as they are preconditioned to identify different instruments in a mix. We have not specified at all exactly how a system using stream separation is to recognise instruments in order to separate them, and perhaps it is too optimistic to assume that the system could even have as good an idea at what constitutes a certain stream as do humans. This would obviously be another source of error, as not only would the separation be imperfect, but it would also be based on goals that are not clearly defined. In a theoretical discussion such as ours, such issues cannot and need not be addressed in order to provide a useful result, but they may prove an issue to any practical implementation in future.

5.4 Conclusion

In this chapter we have presented an analysis of the results obtained in Section 4. We examined the chord accuracy data for each of the various cases tested and concluded that whilst the bass offered an improvement on regular chord extraction performance when parallel chord identification errors were forgiven, the instruments (and especially the impure stream) offered much better results even for exact chord recognition. We noted that these results have applications not only to future practical chord transcription implementations, but also to fields such as psychoacoustics. Finally, we identified some of the main limitations and assumptions of our study, and justified these where applicable.

In the final chapter we conclude this thesis with a summary of our project and recommendations for future work.

6. CONCLUSION

In this final section we summarise the main points of this thesis along with the key results from our work. We conclude with several topics that we believe show great promise for the field of automatic chord transcription and that may prove fruitful avenues for further investigation to future researchers.

6.1 Summary

In this thesis we have examined the psychoacoustically motivated technique of stream separation and its application to the process of automatic chord extraction from musical audio to determine its effect upon chord recognition performance. This was done with the ultimate aim of improving current chord transcription techniques and assessing whether stream separation is capable of doing so.

We opened by presenting a brief study of the theoretical background for the project in Section 2: music theory, current techniques for automatically transcribing chords from audio and the principles behind stream separation. The possibility of applying stream separation to automatic chord extraction was discussed and its feasibility defended with a study of various processes that either constitute or contribute to stream separation as a whole. We concluded from this that stream separation is a fundamentally realisable technique with current technology and that therefore to assess its effect upon chord transcription is both relevant and useful.

In Section 3 we presented a simple model of human perception. This assumed that the brain employs stream separation in order to detect chords by separating musical audio input into four 'streams': bass, vocals, instrumentation and drums. We then designed and conducted an experiment based upon this model to simulate this process and see if it yielded improved chord recognition performance.

Four pop songs – "Chelsea" and "Natalie Grey" by Summertime's End, "Good Time" by Owl City and "Tread The Water" by Malachi Joshua – were selected for this experiment, and 62 chord samples taken from among them. For each chord sample, seven different 'mixes' were made. The first of these was denoted the "base case" or "control case" and was simply a segment of the song. For each of the bass, vocals and instruments, two mixes were made: one containing only the stream in question, and one containing this stream with the other three streams mixed in at 25% of full power. This resulted in 434 .wav files tested.

MATLAB code was created to take each .wav file and use a function from the MIRtoolbox function set to extract chroma data from it. The inner product of this chroma data with a predefined chord template matrix was evaluated to yield chord likelihood scores for a set of 24 of the most common triad chords in pop music. The chord with the highest likelihood score for each file was selected and recorded as the predicted chord for that file. We measured the results in terms of an 'exact identification' metric, which simply measured the correct identifications; a 'relative identification' metric, which also considered relative major/minor chords as correct identifications; and a 'parallel identification' metric, which considered parallel major/minor chords as correct identifications.

In Sections 4 and 5 the results obtained from this experiment were presented and analysed. The aforementioned chord detection method was shown to exactly identify the correct chord 58.06% of the time in the base case. It was found that using the instruments identified the correct chord 91.94% of the time in the 'pure' case and 98.39% of the time in the 'impure' case. The bass was also shown to predict either the correct chord or its parallel major or minor 87.10% of the time for the 'impure bass' case.

We concluded that the success of the instruments stream showed that it was possible to use stream separation to improve automatic chord extraction performance, and that the instruments stream was best suited to do so, although the bass may prove a more practical source due to its relative ease of extraction. In doing so, the overall aim for this thesis – to assess the usefulness of stream separation in automatic chord extraction – was achieved by demonstrating that stream separation does indeed allow for improvement to current chord estimation accuracy.

6.2 Future Work

Although we have achieved our objectives in this project, there is still much to be done in this field of study; we have merely laid the groundwork for continuing work on this topic. There are also limitations, such as those we have identified, which must be addressed. Here we describe some possible directions for future efforts in this field.

6.2.1 Automatic Implementation of Stream Separation

Perhaps the most obvious extension of the work presented herein is the automatic implementation of a stream separation technique. As noted in Section 5.3.2, this was not done in this project. Consequently, the most obvious progression of this work would be to create a method that does so. Such a method should be able to accept audio files as an input and extract various streams from this audio. It is not necessary to use the delineations used for 'streams' in this thesis – namely, drums, instruments, bass and vocals – in any implementation of stream separation; these are merely one suggested set of streams. However, based upon the results demonstrated in this project, it is envisioned that a stream separation method capable of extracting instrumentation (as opposed to bass, vocals or drums) in some sense would be most successful.

6.2.2 Using Multiple Streams for Analysis

Another natural extension of this work is the investigation of using other kinds of stream mixes to assess chord transcription performance. We found that instruments with some other of the remaining streams left in (i.e. the 'impure instruments' mix) yielded the best chord recognition performance – even better than the pure instrument stream. This suggests that the contribution of these other streams improves the chord estimation performance somewhat. What if we were to mix, say, bass with instruments, and then evaluated the most likely chord from this? Or further, what if we used the bass to determine the root note, the drums to conduct a rhythmic analysis (see Section 3.1) and thus segment the chords, the vocals to check the key, and then used all of this along with the instruments to guide the chord detection? The investigation of such questions is an interesting prospect for future studies.

6.2.3 Tolerance for Stream Discontinuity

Following from the discussion in Section 5.3.4 about streams not always playing continuously through a song, a future improvement to make to this experiment (or process, if it is implemented) is to develop a tolerance for streams playing intermittently, such as the bass dropping out of the song for a section. Such occurrences are common in music and being able to tolerate these in a stream separation method (and hence reduce false stream classification which may impair performance) would make the technique far more robust. One possibility might be the implementation of a step which tests to see which streams are present at any time in a song and adjusts its calculations accordingly.

6.2.4 Application to Other Genres

This thesis has focused primarily on pop music, due to one of the primary motivations behind the project being the development of a tool that could provide musicians with chord information; such information is most useful to pop musicians (since classical musicians generally work from sheet music rather than just chord labels). There were therefore 24 chords used in this experiment: the 12 major and minor triads. However, this is nowhere near an exhaustive list of the chords that can appear in music, and especially in genres such as jazz more complex chords are frequent. Furthermore, different genres have different playing styles and different standard instrumentation: instrumental music obviously lacks vocals; acoustic music may lack drums; and so on. The first issue is easily dealt with by the expansion of the chord template matrix $\boldsymbol{T}$ to include bit masks for non-triad chords. For example, the Cmaj7 chord, which contains the notes C, E, G and B, could be added as a column in $\boldsymbol{T}$ with ones in elements 1, 5, 8 and 12, and likewise for other complex chords. The second problem is more difficult; models of the difference in playing styles need to be developed and translated into a computationally useful form. However, doing so would prove useful in rendering the technique of stream separation applicable to a more general subset of music, not merely pop.

6.2.5 Integration into Models of Musical Context

The experiment conducted in this project to assess chord recognition performance on various streams used the method first proposed by Fujishima to evaluate chords: taking the inner product of the chroma vector with a chord template matrix, and then the selection of the chord that consequently yielded the highest likelihood score (see Sections 2.2 and 3.3.4). This decision was justified as it was argued that the consistency of the chord estimation method throughout the experiment was more important to a useful result than was the accuracy of this method. However, after demonstrating (at least theoretically) that stream separation is a useful technique for assisting chord transcription, the next step is naturally to integrate this technique into another chord estimation method more advanced than that of Fujishima. In Section 2.2 were given several such methods. The most obvious of these to integrate stream separation into would be the model of musical context proposed by Mauch: this is currently the most advanced and most accurate method in the field and consequently the best platform from which to attempt improvements. Noting that stream separation as a technique is somewhat modular in nature – that is, that it can be 'inserted' into an existing chord recognition technique to extract the most effective streams for analysis – it can be seen that this would likely improve the accuracy of this computational model of musical context even further.

The development of new and more powerful technologies and algorithms in the field of automatic chord transcription is an exciting prospect. The implementation of psychoacoustically motivated techniques such as stream separation alongside such developments will hopefully one day allow machines to match the chord transcription performance of the human brain.

7. BIBLIOGRAPHY

[1] Helene Papadopoulos and Geoffroy Peeters, "Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM," in *Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop on.*, Bordeaux, 2007, pp. 53-60.

[2] Anssi Klapuri, *Introduction to Music Transcription*.: Springer Science + Business Media, 2006.

[3] E. Bruce Goldstein, *Blackwell handbook of perception*, 4th ed.: Wiley-Blackwell, 2001.

[4] David A. Schwartz and Dale Purves, "Pitch is determined by naturally occurring periodic sounds," vol. 194, no. 1, pp. 31-46, 2004.

[5] Gregory H. Wakefield, "Mathematical representation of joint time-chroma distributions," in *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, Chicago, 1999, pp. 637-645.

[6] Arnold Whitall, *The Cambridge Introduction to Serialism*. New York: Cambridge University Press, 2008.

[7] J. F. Schouten, "The perception of subjective tones," in *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen*, vol. 41, 1938, pp. 1086-1093.

[8] Harvey Fletcher, "The physical criterion for determining the pitch of a musical tone," *Physical review*, vol. 23, no. 3, p. 427, 1924.

[9] *Oxford Dictionary of English*, 3rd ed.: Oxford University Press, 2010.

[10] Tao Li, Mitsunori Ogihara, and Qi Li, "A comparative study on content-based music genre classification.," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 282-289.

[11] Walter Piston, *Harmony*, 5th ed.: Norton, W. W. & Company, Inc. , 1987.

[12] Amélie Anglade, Rafael Ramirez, and Simon Dixon, "Genre classification using harmony rules induced from automatic chord transcriptions," in *ISMIR*, 2009, pp. 669-674.

[13] W. Everett, Ed., *Expression in Pop-rock Music: A Collection of Critical and Analytical Essays*. London: Taylor & Francis, 2000.

[14] J. Shepherd, Ed., *Continuum Encyclopedia of Popular Music of the World: Performance and production*.: Continuum, 2003.

[15] J. Serrà, Á. Corral, M. Boguñá, M. Haro, and J. L. Arcos, "Measuring the evolution of

contemporary western popular music," *Scientific reports*, vol. 2, 2012.

[16] Apple Inc. (2010, February) iTunes Store Tops 10 Billion Songs Sold. [Online]. http://www.apple.com/pr/library/2010/02/25iTunes-Store-Tops-10-Billion-Songs-Sold.html

[17] Apple Inc. iTunes Preview. [Online]. https://itunes.apple.com/en/genre/music/id34

[18] Matthias Mauch, "Automatic Chord Transcription from Audio Using Computational Models of Musical Context ," Queen Mary, University of London, London, PhD Thesis 2010.

[19] Takuya Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music. ," in *International Computer Music Conference*, Beijing, China, 1999, pp. 464-467.

[20] Gregory H. Wakefield, "Mathematical representation of joint time-chroma distributions," in *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, 1999, pp. 637-645.

[21] Alexander Sheh and Daniel P.W. Ellis, "Chord segmentation and recognition using EM-trained HMM," in *ISMIR*, Baltimore, MD, 2003, pp. 183-189.

[22] Christopher Harte and Mark Sandler, "Automatic chord identifcation using a quantised chromagram," in *Audio Engineering Society Convention 118*, 2005.

[23] Juan Pablo Bello and Jeremy Pickens, "A Robust Mid-Level Representation for Harmonic Content in Music Signals," in *ISMIR*, vol. 5, 2005, pp. 304-311.

[24] Yasushi Konoki, Norio Emura, and Masanobu Miura, "Chord estimation using chromatic profiles of sounds played by an electric guitar," in *Proceedings of the International Conference on Music Perception & Cognition (ICMPC)*, pp. 734-737.

[25] Yushi Ueda, Yuuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama, "HMM-based approach for automatic chord detection using refined acoustic features," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 5518-5521.

[26] Kyogu Lee, "Automatic chord recognition from audio using enhanced pitch class profile," in *Proc. of the Intern. Computer Music Conference (ICMC)*, New Orleans, USA, 2006.

[27] Masataka Goto and Yoichi Muraoka, "Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions," *Speech Communication*, vol. 27, no. 3, pp. 311-335, 1999.

[28] A.S. Bregman, *Auditory scene analysis: the perceptual organization of sound*, 2nd ed.: MIT Press, 1994.

[29] Christian Uhle, Christian Dittmar, and Thomas Sporer, "Extraction of drum tracks from polyphonic music using independent subspace analysis," in *Proc. ICA*, 2003, pp. 843-847.

[30] Marko Helén and Tuomas Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine.," in *Proc. EUSIPCO*, 2005.

[31] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Integration and adaptation of harmonic and inharmonic models for separating polyphonic musical signals," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, 2007, pp. I-57.

[32] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*.: Wiley & Sons, 2001.

[33] Nobutaka Ono, Kenichi Miyamoto, Jonathan Le Roux, Hirokazu Kameoka, and Shigeki Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram.," in *Proc. EUSIPCO*, 2008.

[34] Derry FitzGerald, "Harmonic/percussive separation using median filtering," in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010.

[35] Vishweshwara Rao and Preeti Rao, "Vocal melody extraction in the presence of pitched accompaniment in polyphonic music.," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 8, pp. 2145-2154, 2010.

[36] Masataka Goto, "A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals ," *Speech Communication*, vol. 43, no. 4, pp. 311-329, 2004.

[37] Matti P Ryynänen and Anssi P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72-86, 2008.

[38] Beiming Wang and Mark D. Plumbley, "Musical audio stream separation by non-negative matrix factorization," in *Proc. DMRN summer conf.*, 2005, pp. 23-24.

[39] Alexey Ozerov, Emmanuel Vincent, and Frédéric Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1118-1133,

2012.

[40] Juan J. Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera, "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals," *ISMIR*, pp. 559-564, 2012.

[41] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen, "Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation," *ISMIR*, pp. 327-332, 2009.

[42] Norman Hollyn, *The Film Editing Room Handbook: How to Tame the Chaos of the Editing Room*, 4th ed.: Peachpit Press, 2009.

[43] C. E. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 10-21, Jan 1949.

[44] University of Jyväskylä. (2013, August) MIRtoolbox. [Online]. https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

8. APPENDICES

8.1 MATLAB Code

The MATLAB codes used in this experiment and described in Section 3.3.4.5 are reproduced here.

8.1.1 Chroma Extraction: chroma_extract.m

This code was used to feed the audio into 'mirchromagram.m' for chroma analysis, and to retrieve the output data from this function (the chroma vector ***c***) to be used later.

```
function [pitch_class_profiles] = chroma_extract(song,inst,sep,num_files)
%chroma_extract Uses mirchromagram.m to extract chromagram data from a list
%of audio files

for i=1:1:num_files
    %Concatenate filename string
    filename =
strcat(song,'_',inst,'_',sprintf('%03d',sep),'_',sprintf('%02d',i));
    %Extract chroma data using mirchromagram from 20 Hz to 20 kHz
    chromagram = mirchromagram(filename,'Min',20,'Max',20000);
    %Retrieve data from mirchromagram into matrix format
    pitch_class_profiles(i,1:12) = mirgetdata(chromagram)';

end
```

8.1.2 Maximum Likelihood Chord Detection: ml_func.m

This function defined the chord template matrix ***T*** and evaluated the likelihood array ***l*** by taking the inner product of ***c*** with ***T***.

```
function [chord_scores, m_likelihoods, ml_chords] =
ml_func(pitch_class_profiles)
%ml_func Performs maximum likelihood chord detection based on a simple
%binary bitmask for 12 major and 12 minor triads. Input is a series of
%pitch class profile vectors; output is an array of corresponding most
%likely chords.

%Define chord template matrix
chord_templates = [1    0    0    0    0    1    0    0    1    0    0    0    1    0
0    0    0    1    0    0    0    1    0    0
0    1    0    0    0    0    1    0    0    1    0    0    0    1    0    0    0    0    1
0    0    0    1    0
0    0    1    0    0    0    0    1    0    0    1    0    0    0    1    0    0    0    0
1    0    0    0    1
0    0    0    1    0    0    0    0    1    0    0    1    1    0    0    1    0    0    0
0    1    0    0    0
1    0    0    0    1    0    0    0    0    1    0    0    0    1    0    0    1    0    0
0    0    1    0    0
0    1    0    0    0    1    0    0    0    0    1    0    0    0    1    0    0    1    0
0    0    0    1    0
0    0    1    0    0    0    1    0    0    0    0    1    0    0    0    1    0    0    1
0    0    0    0    1
1    0    0    1    0    0    0    1    0    0    0    0    1    0    0    0    1    0    0
1    0    0    0    0
0    1    0    0    1    0    0    0    1    0    0    0    0    1    0    0    0    1    0
0    1    0    0    0
```

```
0    0    1    0    0    1    0    0    0    1    0    0    0    0    1    0    0    0    1
0    0    1    0    0
0    0    0    1    0    0    1    0    0    0    1    0    0    0    0    1    0    0    0
1    0    0    1    0
0    0    0    0    1    0    0    1    0    0    0    1    0    0    0    0    1    0    0
0    1    0    0    1];

%Define chord alphabet
chord_list =
{'C','C#','D','D#','E','F','F#','G','G#','A','A#','B','Cm','C#m','Dm','D#m'
,'Em','Fm','F#m','Gm','G#m','Am','A#m','Bm'};

%Take product of PCP vectors with chord template matrix
chord_scores = pitch_class_profiles * chord_templates;
%Select largest likelihood product for each file
[m_likelihoods, idx] = max(chord_scores,[],2);
%Retrieve chord labels corresponding to maximum likelihoods
ml_chords = {chord_list{idx}}.';
%Convert likelihood products to string and concatenate with labels
%ml_data = [ml_chords arrayfun(@num2str,m_likelihoods,'unif',0)];

end
```

8.1.3 Writing to File: thesis_xls_write.m

This code was used to co-ordinate the entire process and to export the data retrieved into Microsoft Excel for analysis.

```
function [ml_chords] = thesis_xls_write(song,inst,sep,num_files)
%thesis_xls_write Automatically performs chromagram extraction and maximum
%likelihood chord detection. Writes the results to csv files and returns
%chord labels as a cell array (since OS X does not allow automatic writing
%of cell arrays to csv files).

%Extract chromagram and perform maximum likelihood chord detection
[PCPs] = chroma_extract(song,inst,sep,num_files);
[chord_scores, m_likelihoods, ml_chords] = ml_func(PCPs);

%Pad likelihood array with zeroes to be replaced with labels (manually)
padded_mls = [zeros(size(m_likelihoods)) m_likelihoods];

%Create filename strings for writing to files
cs_filename =
strcat('CHORDSCORES_',song,'_',inst,'_',sprintf('%03d',sep),'.xls');
ml_filename =
strcat('MLDATA_',song,'_',inst,'_',sprintf('%03d',sep),'.xls');
%mlc_filename =
strcat('MLCHORDS_',song,'_',inst,'_',sprintf('%03d',sep),'_',sprintf('%02d'
,i),'.xls');

%Write data to csv files
xlswrite(cs_filename, chord_scores, 'Sheet1', 'A1');
xlswrite(ml_filename, padded_mls, 'Sheet1', 'A1');
%xlswrite(mlc_filename, strvcat(ml_chords), 'Sheet1', 'A1');

end
```

8.2 Chord Likelihood Data

Given below is the chord likelihood array ***l*** for each song, stream and separation level. The column headers represent chords, and the rows represent the various chord samples taken from each song. For clarity and brevity, values have been truncated to three decimal places for reproduction here.

8.2.1 'No Separation' Case

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	1.063	1.109	1.745	0.89	1.422	1.623	1.143	1.186	1.103	2.137	1.034	1.303	0.955	1.417	1.733	1.027	1.274	0.903	1.842	0.91	1.315	1.931	1.13	1.323
2	1.184	0.461	0.678	0.289	1.256	0.383	0.353	0.846	0.467	1.262	0.796	0.625	0.467	1.262	0.79	0.375	1.256	0.383	0.348	0.597	0.539	1.184	0.466	0.933
3	0.697	0.547	0.895	0.754	1.404	0.441	0.98	1.493	0.717	0.726	0.817	1.98	0.794	0.726	0.64	1.157	1.481	0.441	0.802	0.67	1.5	0.62	0.724	1.895
4	0.71	1.389	0.843	0.52	0.751	0.496	1.134	1.062	0.817	1.357	0.943	0.772	0.767	1.505	0.946	0.535	0.701	0.644	1.138	0.824	0.807	0.612	1.237	1.077
5	0.653	1.355	0.352	0.256	0.697	0.434	1.049	0.428	0.517	1.483	0.574	0.411	0.421	1.577	0.569	0.298	0.601	0.528	1.044	0.315	0.465	0.655	1.266	0.47
6	1.134	0.097	0.553	0.308	1.065	0.112	0.289	0.449	0.376	1.086	0.361	0.543	0.409	1.087	0.361	0.478	1.098	0.112	0.289	0.383	0.341	1.101	0.097	0.618
7	0.995	0.707	2.027	0.65	0.698	1.359	1.005	0.761	0.881	2.136	0.654	1.01	0.937	1.211	1.654	0.963	0.755	0.434	2.005	0.714	0.641	1.863	0.632	1.074
8	0.972	0.561	1.064	0.597	1.692	0.4	0.833	1.701	0.667	0.891	0.729	1.909	0.758	0.966	0.723	0.938	1.783	0.475	0.827	0.73	1.477	0.805	0.492	2.041
9	0.556	0.637	2.221	0.527	0.404	0.69	1.51	1.037	0.778	1.105	0.969	1.595	0.785	0.8	1.311	1.482	0.411	0.385	1.852	0.925	0.633	0.854	0.599	1.992
10	0.761	1.211	0.894	0.575	0.692	0.326	1.366	0.665	0.943	1.462	0.669	0.897	0.806	1.623	0.624	0.873	0.554	0.487	1.321	0.642	0.737	0.738	1.096	0.963
11	0.25	0.186	0.573	0.343	1.097	0.119	0.351	1.471	0.4	0.224	0.434	1.418	0.434	0.253	0.406	0.445	1.131	0.148	0.323	0.499	1.281	0.186	0.185	1.573
12	0.943	1.137	0.736	0.945	0.999	0.316	1.213	0.785	1.291	1.66	0.569	1.392	1.214	1.751	0.569	1.144	0.923	0.407	1.213	0.537	1.27	0.929	1.046	0.984
13	0.735	0.444	1.749	0.608	0.919	1.307	0.446	1.236	0.733	1.68	0.757	1.025	0.796	0.745	1.719	0.591	0.981	0.372	1.408	0.803	0.979	1.608	0.417	1.22
14	1.324	0.669	0.712	0.561	1.588	0.401	0.625	1.044	0.968	1.469	0.643	1.18	0.871	1.579	0.644	0.705	1.49	0.512	0.626	0.568	1.134	1.311	0.557	1.187
15	0.498	0.503	0.965	0.562	1.282	0.339	0.752	1.627	0.706	0.471	0.65	1.842	0.725	0.557	0.64	0.875	1.301	0.425	0.742	0.661	1.509	0.393	0.427	1.941
16	0.858	1.342	0.752	0.784	0.934	0.455	1.183	0.839	1.262	1.502	0.708	1.157	1.1	1.705	0.708	0.927	0.772	0.657	1.183	0.609	1.176	0.818	1.139	0.982
17	0.382	0.563	0.551	0.439	1.196	0.226	0.624	1.464	0.52	0.609	0.477	1.509	0.566	0.641	0.468	0.522	1.242	0.258	0.615	0.478	1.38	0.304	0.541	1.547
18	0.747	0.67	1.786	0.451	1.097	1.442	0.537	1.242	0.637	1.837	0.899	1.056	0.628	0.911	1.872	0.54	1.087	0.516	1.51	0.725	0.977	1.683	0.623	1.33

Table 8.1: Chord likelihoods for 'Chelsea', no separation mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.878	1.669	0.91	0.84	0.77	0.602	1.435	0.945	1.046	1.347	0.497	0.672	1.004	1.667	0.66	0.657	0.728	0.922	1.599	0.93	0.897	0.6	1.186	0.761
2	1.607	1.736	1.548	1.734	1.456	1.397	1.608	1.255	1.854	1.317	1.187	1.161	1.663	1.727	1.226	1.575	1.265	1.807	1.648	1.668	1.512	1.388	1.287	1.095
3	1.639	1.769	1.533	2.487	1.546	1.186	1.594	1.606	2.252	1.408	1.253	1.815	2.274	1.805	1.283	2.1	1.568	1.582	1.623	1.89	2.181	1.221	1.344	1.219
4	1.524	1.935	1.359	1.961	1.255	1.592	1.414	1.559	1.391	1.079	1.956	1.35	1.667	1.335	1.79	1.639	1.53	1.848	1.247	1.848	1.398	0.992	1.844	1.237
5	1.033	1.742	1.013	1.532	0.929	0.917	2.07	0.895	1.27	1.58	0.992	0.936	1.216	1.746	0.756	1.411	0.874	1.084	1.833	1.37	1.111	0.921	1.812	0.775
6	0.586	1.469	0.658	0.706	1.2	0.57	0.685	0.58	1.395	0.691	0.46	0.559	0.718	1.427	0.583	0.61	0.523	1.305	0.808	0.631	1.332	0.527	0.61	0.484
7	1.137	1.034	1.265	2.077	0.932	0.765	1.22	1.209	1.757	0.867	1.134	1.618	1.856	1.087	0.996	1.944	1.032	0.985	1.082	1.535	1.651	0.818	0.952	1.076
8	1.176	1.822	1.338	1.645	0.901	1.536	1.424	1.39	1.08	0.933	1.99	1.151	1.341	1.067	1.796	1.469	1.162	1.67	1.23	1.708	1.066	0.781	1.882	1.214
9	0.722	1.385	0.607	0.598	0.398	0.589	1.33	0.511	0.819	1.267	0.334	0.436	0.832	1.386	0.438	0.513	0.411	0.708	1.435	0.588	0.508	0.59	1.161	0.426
10	1.068	1.693	1.217	1.197	1.355	1.081	1.248	0.941	1.532	1.004	0.978	0.916	1.097	1.559	1.042	1.162	0.921	1.636	1.313	1.187	1.385	0.947	1.073	0.906
11	1.296	1.287	1.232	2.39	0.924	0.916	1.582	1.193	2.001	0.917	1.284	1.605	2.091	1.247	0.945	2.186	1.014	1.246	1.244	1.774	1.719	0.876	1.296	0.99
12	1.322	1.983	1.124	1.417	1.134	1.797	1.111	1.309	1.454	1.049	1.688	1.142	1.521	1.245	1.744	1.159	1.201	1.993	1.167	1.326	1.333	1.059	1.73	1.051
13	0.921	2.03	1.138	0.906	0.929	1.166	1.75	0.924	1.022	1.577	0.895	0.725	0.906	1.739	1.076	0.833	0.814	1.328	1.931	1.031	0.915	0.874	1.687	0.85
14	1.403	1.351	1.241	2.193	1.063	1.093	1.541	1.233	1.995	1.06	1.342	1.719	2.074	1.321	1.066	2.098	1.142	1.354	1.265	1.612	1.735	1.063	1.366	1.138
15	0.99	1.748	0.923	1.02	0.835	0.716	1.684	0.924	1.021	1.426	0.851	0.706	0.998	1.736	0.764	0.906	0.812	1.026	1.597	1.124	0.843	0.703	1.525	0.81
16	0.287	1.191	0	0.01	0.546	0.001	1	0.069	0.202	1.286	0.000	0.08	0.011	1.477	0.000	0.01	0.356	0.191	1	0	0.27	0.287	1	0.069

Table 8.2: Chord likelihoods for ‘Natalie Grey’, no separation mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	1.644	1.907	1.409	1.667	2.036	1.496	1.322	1.605	2.065	1.499	1.438	1.467	1.68	1.97	1.481	1.448	1.65	1.967	1.364	1.586	2.072	1.558	1.393	1.386
2	2.192	2.051	2.091	2.879	2.286	2.092	2.112	2.379	2.446	2.028	2.379	2.317	2.499	2.063	2.187	2.55	2.339	2.128	1.92	2.612	2.593	2.104	2.208	2.05
3	1.923	2.105	2.013	2.359	1.989	2.235	2.061	1.941	2.015	2.003	2.57	1.835	1.95	1.931	2.35	2.243	1.924	2.163	1.841	2.348	2.016	2.06	2.398	1.825
4	2.516	1.954	1.879	2.612	2.186	2.335	1.761	2.19	2.682	1.966	2.026	2.221	2.794	1.994	2.035	2.257	2.298	2.363	1.77	2.226	2.464	2.375	1.917	1.835
5	2.301	2.362	2.171	2.506	2.434	2.228	2.113	2.292	2.532	2.176	2.197	2.219	2.415	2.364	2.215	2.293	2.317	2.416	2.131	2.366	2.549	2.23	2.156	2.079
6	1.841	2.054	1.865	2.39	2.175	1.96	1.949	1.89	2.254	1.883	2.072	1.802	1.981	2.045	1.94	2.15	1.902	2.123	1.817	2.237	2.315	1.951	2.024	1.649
7	2.3	2.428	2.572	2.575	2.411	2.571	2.316	2.565	2.481	2.388	2.8	2.173	2.363	2.331	2.77	2.392	2.293	2.515	2.286	2.784	2.473	2.474	2.514	2.382
8	2.481	2.499	2.329	2.806	2.422	2.675	2.288	2.474	2.854	2.347	2.532	2.433	2.803	2.351	2.534	2.546	2.371	2.68	2.29	2.587	2.745	2.528	2.492	2.214
9	1.56	1.729	1.445	1.941	1.964	1.49	1.296	1.675	2.15	1.387	1.574	1.491	1.778	1.78	1.53	1.623	1.592	1.882	1.252	1.807	2.182	1.541	1.38	1.357

10	2.197	1.885	2.149	2.781	2.032	2.086	1.855	2.466	2.43	1.763	2.287	2.294	2.63	1.78	2.225	2.41	2.232	2.102	1.793	2.582	2.465	1.98	1.931	2.095
11	1.832	1.771	1.804	2.288	1.897	2.01	1.879	1.966	1.978	1.655	2.425	1.83	1.936	1.645	2.066	2.158	1.856	1.999	1.52	2.293	2.001	1.884	2.141	1.836
12	2.561	2.219	2.368	2.798	2.252	2.589	2.13	2.572	2.709	2.022	2.44	2.575	2.907	2.007	2.456	2.577	2.45	2.575	2.146	2.573	2.598	2.377	2.218	2.35
13	0.789	1.282	0.748	1.013	1.28	0.762	0.586	0.677	1.369	0.581	0.663	0.603	0.845	1.208	0.74	0.805	0.756	1.388	0.663	0.879	1.336	0.687	0.579	0.469
14	0.48	0.403	0.429	1.473	0.467	0.488	0.511	0.575	1.576	0.275	0.608	1.249	1.461	0.393	0.406	1.389	0.352	0.606	0.309	0.715	1.448	0.478	0.487	0.491
15	0.147	0.23	0.134	1.021	0.156	0.477	1	0.189	0.168	0.101	1.263	0.176	0.168	0.001	0.364	1.021	0.155	0.377	0.101	1.034	0.177	0.248	1.229	0.189
16	1.892	0.736	0.803	1.703	1.006	1.332	0.812	1.656	1.652	0.58	0.644	1.062	2.171	0.678	0.637	0.96	1.525	1.43	0.804	1.554	1.286	1.274	0.645	0.913

Table 8.3: Chord likelihoods for ‘Good Time’, no separation mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.217	1.027	0.28	0.008	0.194	1.338	0.008	0.008	0.085	0.44	1.008	0.008	0.058	0.186	1.28	0.008	0.168	1.085	0.281	0.008	0.035	0.498	1.008	0.008
2	1.195	0.015	0.202	1.002	0.218	0.297	0.002	1.322	0.19	0.098	0.126	0.212	1.19	0.006	0.217	0.002	1.218	0.204	0.092	1.112	0.212	0.288	0.016	0.322
3	0.95	0.677	1.079	0.354	0.858	1.827	0.443	0.366	0.745	1.493	0.617	0.321	0.592	0.723	1.35	0.327	0.706	1.057	1.176	0.373	0.5	1.873	0.714	0.34
4	1.819	1.625	1.435	0.778	1.322	2.443	0.79	1.483	1.434	1.464	2.048	0.822	1.434	1.232	2.274	0.662	1.323	2.212	1.017	1.324	0.937	2.05	1.63	1.368
5	0.932	0.087	1	0.142	1.153	1.094	0.032	0.229	0.117	1.925	0.119	0.339	0.117	0.925	1.087	0.142	1.153	0.094	1	0.032	0.339	1.932	0.119	0.229
6	0.155	0.242	0.567	0.14	1.128	0.283	0.264	1.337	0.202	0.28	0.493	1.215	0.16	0.218	0.55	0.264	1.086	0.221	0.321	0.386	1.133	0.259	0.247	1.461
7	0.305	1.009	0.148	0.024	0.317	1.145	0.087	0.074	0.084	0.318	1.013	0.155	0.079	0.247	1.077	0.094	0.311	1.073	0.152	0.013	0.09	0.382	1.017	0.144
8	1.517	0.442	0.59	0.375	0.747	1.445	0.227	0.696	1.25	0.66	0.762	0.35	1.231	0.596	0.815	0.283	0.728	1.381	0.28	0.629	0.461	1.599	0.451	0.604
9	0.596	1.115	0.4	0.147	0.306	1.697	0.057	0.223	0.38	0.58	1.006	0.093	0.447	0.264	1.39	0.016	0.373	1.38	0.441	0.147	0.157	0.846	1.047	0.093
10	1.093	0.024	0.431	1.003	0.06	0.205	0	1.362	0.075	0.129	0.347	0.042	1.075	0.021	0.455	0.003	1.06	0.097	0.108	1.323	0.042	0.201	0.024	0.362
11	0.512	0.111	1.068	0.011	0.42	1.521	0.119	0.361	0.457	1.102	0.075	0.433	0.457	0.102	1.068	0.079	0.42	0.521	1.111	0.007	0.365	1.512	0.118	0.429
12	2.191	1.083	0.823	0.547	1.312	2.094	0.464	0.812	1.284	1.462	1.389	0.48	1.365	1.281	1.506	0.461	1.393	1.913	0.581	0.793	0.486	2.292	1.147	0.726

Table 8.4: Chord likelihoods for ‘Tread The Water’, no separation mixes

8.2.2 ‘Pure Instruments’ Case

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.682	0.377	1	0.006	0.893	1	0.253	0.1	0.124	1.929	0	0.094	0.006	1.052	1	0	0.776	0.124	1.253	0.006	0.218	1.676	0.253	0.094
2	1.008	0.241	0.118	0.06	2.187	0.062	0.065	0.958	0.296	1.054	0.025	1.045	0.055	1.241	0.065	0.112	1.946	0.249	0.106	0.025	1.234	1.062	0.013	1.01
3	0.05	0.009	0.31	0.579	1.028	0.002	0.485	1.022	0.389	0.038	0.168	1.698	0.411	0.037	0.002	0.866	1.05	0	0.319	0.19	1.389	0.03	0.177	1.309

4	0.965	1.445	0.01	0.05	2.027	0.009	0.456	0.074	1.038	1.41	0.008	0.108	0.043	2.403	0.007	0.047	1.032	1.002	0.455	0.013	1.105	0.967	0.453	0.072
5	0.461	1.763	0.038	0.18	1.023	0.278	1.031	0.09	0.772	1.403	0.216	0.28	0.235	1.943	0.223	0.201	0.486	0.818	1.038	0.01	0.797	0.458	1.216	0.111
6	1.002	0.267	0.106	0.036	1.596	0.02	0.081	0.364	0.292	1.004	0.036	0.458	0.038	1.254	0.039	0.116	1.342	0.27	0.084	0.022	0.632	1.007	0.014	0.444
7	0.96	0.806	1.012	0.146	1.139	1.012	0.619	0.298	0.18	2.432	0.02	0.15	0.146	1.61	1.02	0.005	1.105	0.19	1.619	0.153	0.325	1.816	0.628	0.157
8	0.025	0.047	1.036	0.629	0.813	0.049	1.059	0.798	0.6	0.09	0.038	2.393	0.604	0.054	0.049	1.624	0.818	0.013	1.07	0.029	1.393	0.056	0.072	1.793
9	0.065	0.376	1.057	0.03	0.194	0.117	1.226	0.009	0.119	0.319	0.089	1.01	0	0.381	0.117	1.03	0.075	0.179	1.254	0.03	0.129	0.122	0.286	1.01
10	0.244	1.836	0.106	0.127	0.881	0.335	1.078	0.148	0.678	1.232	0.292	0.305	0.145	1.755	0.316	0.183	0.348	0.858	1.102	0.025	0.782	0.254	1.289	0.204
11	0.225	0.08	0.801	0.663	1.254	0.094	0.805	1.006	0.639	0.331	0.066	2.325	0.61	0.285	0.094	1.378	1.225	0.048	0.833	0.058	1.639	0.3	0.098	1.721
12	0.184	1.143	0.064	0.327	1.542	0.146	0.698	1.005	0.685	0.853	0.126	1.366	0.327	1.196	0.146	0.366	1.184	0.489	0.718	0.005	1.685	0.199	0.78	1.044
13	0.936	0.262	1.034	0.151	1.468	1.001	0.224	0.687	0.11	2.012	0.001	0.648	0.151	1.083	1.001	0.073	1.51	0.072	1.224	0.113	0.684	1.823	0.19	0.609
14	1.001	0.175	0.112	0.031	1.579	0.01	0.11	0.413	0.198	1.001	0.009	0.554	0.032	1.166	0.01	0.141	1.413	0.175	0.112	0.001	0.61	1.002	0.008	0.523
15	0.111	0.216	1.032	0.814	1.073	0.11	1.286	1.035	0.651	0.313	0.178	2.599	0.679	0.251	0.102	1.744	1.101	0.047	1.209	0.179	1.641	0.145	0.356	1.964
16	0.496	2.127	0.045	0.404	1.268	0.473	1.023	0.193	1.125	1.423	0.404	0.523	0.45	2.124	0.427	0.379	0.594	1.173	1.045	0.048	1.223	0.47	1.404	0.167
17	0.123	0.171	1.07	0.694	1.05	0.101	1.28	1.054	0.501	0.262	0.171	2.483	0.559	0.198	0.098	1.633	1.108	0.037	1.206	0.205	1.486	0.129	0.308	1.993
18	1.233	0.715	0.837	0.245	1.435	0.842	0.481	0.444	0.251	2.307	0.018	0.226	0.251	1.707	0.845	0.019	1.435	0.242	1.308	0.236	0.453	1.834	0.489	0.217

Table 8.5: Chord likelihoods for 'Chelsea', pure instruments mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.608	2.202	0.083	0.75	1.103	0.359	1.094	0.616	1.324	1.03	0.343	0.296	0.793	2.006	0.279	0.289	0.572	1.335	1.03	0.609	1.288	0.163	1.29	0.155
2	0.389	1.028	0.242	0.872	1.007	0.238	0.181	0.338	1.478	0.177	0.13	0.506	0.815	1.01	0.193	0.609	0.343	1.071	0.244	0.442	1.432	0.22	0.132	0.075
3	0.118	0.256	0.319	1.896	0.041	0.517	0.786	0.135	1.037	0.301	1.032	1	1.112	0.042	0.54	1.786	0.116	0.258	0.294	0.921	1.035	0.303	1.007	0.025
4	1.163	1.012	0.166	0.31	0.251	1.888	0.093	0.32	0.788	0.308	1.045	0.149	1.024	0.179	1.129	0.111	0.488	1.759	0.177	0.281	0.113	1.055	1.056	0.12
5	0.229	1.524	0.023	0.459	0.424	0.315	1.303	0.066	0.701	1.016	0.415	0.146	0.346	1.405	0.135	0.421	0.068	0.704	1.023	0.341	0.54	0.196	1.415	0.028
6	0.612	1.282	0.007	1.011	1	0.533	0.435	0.096	2.268	0.265	0.183	0.757	1.363	1.265	0.018	0.922	0.095	1.533	0.27	0.261	1.752	0.516	0.446	0.007
7	0.355	0.525	0.034	1.684	0.517	0.312	0.591	0.131	1.753	0.024	0.63	1.023	1.355	0.496	0.064	1.588	0.119	0.783	0.024	0.696	1.517	0.282	0.621	0.035
8	0.581	1.274	0.005	0.296	0.291	1.417	0	0.185	0.814	0.005	1	0.144	0.708	0.274	1.005	0.127	0.185	1.686	0.005	0.168	0.418	0.417	1	0.017
9	0.168	1.473	0.011	0.184	0.304	0.325	1.022	0.023	0.587	1.011	0.192	0.139	0.306	1.304	0.181	0.161	0.023	0.618	1.011	0.045	0.442	0.156	1.192	0
10	0.271	1.133	0	1.103	1	0.191	0.546	0.092	1.766	0.122	0.435	0.586	0.858	1.122	0.011	1.011	0.092	1.191	0.122	0.516	1.586	0.18	0.557	0
11	0.445	0.187	0.01	1.538	0.188	0.309	0.389	0.174	1.469	0	0.412	1.015	1.445	0.174	0.023	1.389	0.164	0.482	0	0.548	1.188	0.295	0.403	0.024

12	0.779	1.2	0.02	0.024	0.2	1.776	0	0.024	0.955	0.02	1	0	0.779	0.2	1.02	0	0.024	1.955	0.02	0.024	0.2	0.776	1	0
13	0.064	1.565	0.012	0.079	0.54	0.127	1	0.025	0.658	1.012	0.05	0.104	0.143	1.515	0.062	0.079	0.025	0.63	1.012	0	0.619	0.077	1.05	0.025
14	0.252	0.353	0	2.201	0.148	0.211	1	0.258	1.091	0	1.211	0.956	1.201	0.141	0.211	1.95	0.258	0.353	0	1.252	1.097	0	1.211	0.006
15	0.431	1.995	0.056	0.62	0.839	0.398	1.07	0.407	1.119	1.035	0.299	0.366	0.686	1.748	0.284	0.325	0.405	1.111	1.054	0.366	1.094	0.151	1.298	0.112
16	0.438	1.774	0.054	0.114	1.197	0.117	1.045	0.112	0.905	1.371	0.059	0.262	0.182	2.095	0.059	0.159	0.474	0.841	1.045	0.009	0.941	0.438	1.05	0.157

Table 8.6: Chord likelihoods for ‘Natalie Grey’, pure instruments mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.119	1.6	0.117	1.436	1.001	0.776	0.568	0.046	1.932	0.095	1.164	0.86	0.966	1.002	0.703	1.414	0.035	1.682	0.107	0.6	1.848	0.178	1.155	0.024
2	0.787	0.18	0.144	2.214	0.035	0.631	0.909	0.465	1.456	0.048	1.179	1.018	1.773	0.033	0.305	1.893	0.353	0.616	0.035	1.34	1.022	0.484	1.069	0.144
3	0.366	0.158	0.313	1.029	0.07	0.6	1.015	0.286	0.356	0.13	1.342	0.078	0.374	0.017	0.455	1.01	0.088	0.488	0.128	1.219	0.078	0.46	1.157	0.267
4	1.126	0.237	0.095	0.586	0.006	1.167	0.282	0.222	1.25	0.072	0.474	0.254	1.373	0.072	0.262	0.463	0.129	1.167	0.07	0.431	0.254	1.002	0.449	0.099
5	0.094	1.257	0.059	1.107	1	0.371	0.401	0.031	1.745	0.051	0.651	0.689	0.776	1	0.308	1.083	0.031	1.32	0.058	0.425	1.682	0.114	0.651	0.008
6	0.368	0.128	0.026	1.788	0.079	0.199	0.576	0.241	1.232	0.003	0.645	1	1.368	0.082	0.072	1.573	0.215	0.278	0.003	0.814	1.079	0.153	0.623	0.026
7	0.286	0.229	0.584	1.052	0.084	0.593	1.004	0.528	0.326	0.129	1.668	0.098	0.318	0.03	0.788	1.043	0.076	0.494	0.124	1.473	0.116	0.394	1.208	0.519
8	1.317	0.813	0.133	1.289	0.105	1.712	0.598	0.306	1.619	0.205	1.264	0.658	1.831	0.205	0.806	1.116	0.317	1.712	0.14	0.763	0.619	1.105	1.271	0.133
9	0.481	1.322	0.024	1.036	1.003	0.674	0.305	0.151	1.908	0.027	0.627	0.581	1.059	1.003	0.346	0.885	0.154	1.65	0.024	0.455	1.581	0.355	0.627	0
10	1.367	0.191	0.287	2.792	0.19	0.522	1.051	1.273	1.309	0.058	1.137	1.185	2.269	0.067	0.278	1.925	1.15	0.531	0.191	2.014	1.091	0.398	1.041	0.406
11	0.452	0.374	0.537	1.013	0.096	0.92	1.015	0.475	0.463	0.131	1.759	0.096	0.452	0.024	0.88	1.024	0.085	0.814	0.137	1.403	0.096	0.57	1.359	0.487
12	1.45	0.484	0.21	1.282	0.13	1.32	0.588	0.558	1.545	0.255	0.863	0.618	1.9	0.258	0.484	0.966	0.484	1.322	0.208	0.906	0.58	1.093	0.861	0.242
13	0.109	1.494	0.09	1.068	1	0.654	0.401	0.028	1.738	0.077	0.888	0.663	0.763	1.001	0.573	1.054	0.024	1.578	0.087	0.419	1.653	0.161	0.884	0.014
14	0.655	0.169	0.084	1.949	0.026	0.568	0.713	0.31	1.395	0.042	0.939	1	1.633	0.031	0.245	1.709	0.264	0.556	0.019	1.019	1.003	0.429	0.875	0.069
15	0.38	0.184	0.345	1.027	0.019	0.666	1	0.259	0.362	0.126	1.403	0.023	0.384	0.001	0.528	1.005	0.04	0.541	0.125	1.241	0.023	0.483	1.184	0.237
16	1.148	0.283	0.113	0.628	0.009	1.206	0.264	0.263	1.297	0.081	0.506	0.302	1.441	0.081	0.319	0.484	0.153	1.206	0.077	0.445	0.302	1.004	0.47	0.119

Table 8.7: Chord likelihoods for ‘Good Time’, pure instruments mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.973	1.005	0.405	0.115	0.249	2.072	0.03	0.032	0.796	0.624	1.013	0.117	0.816	0.254	1.381	0.118	0.27	1.703	0.398	0.034	0.093	1.322	1.006	0.035
2	0.411	0.226	1.058	0.444	0.394	0.175	0.11	1.745	0.062	0.131	1.15	0.454	0.442	0.107	1.174	0.094	0.774	0.151	0.134	1.384	0.425	0.055	0.226	1.394
3	1.29	0.087	0.338	0.105	1.007	0.7	0.009	0	0.402	1.33	0.08	0.115	0.396	1.007	0.41	0.114	1	0.377	0.338	0	0.113	1.62	0.08	0.009

4	2.071	0.377	0.191	0.671	1.061	0.85	0.099	0.741	0.497	1.032	0.432	0.2	1.117	1.005	0.464	0.144	1.681	0.823	0.13	0.685	0.107	1.478	0.372	0.215
5	1.222	0.161	0.817	0.029	1.05	1.174	0.007	0.047	0.245	1.823	0.144	0.06	0.243	1.017	0.961	0.021	1.047	0.368	0.823	0.008	0.071	2.03	0.151	0.039
6	0.364	0.003	1.083	0.364	0.695	0.053	0.03	2.056	0.003	0.053	1	0.722	0.364	0.003	1.053	0.03	1.056	0.003	0.083	1.364	0.695	0.053	0	1.722
7	0.399	1	0.244	0.06	0.12	1.447	0.016	0.06	0.219	0.348	1	0.016	0.279	0.12	1.228	0.016	0.18	1.219	0.244	0.06	0	0.567	1	0.016
8	2.348	0.141	0.055	0.959	1.052	0.812	0.024	0.754	0.962	1	0.169	0.356	1.632	1.004	0.169	0.308	1.723	0.815	0.024	0.706	0.336	1.674	0.138	0.103
9	0.59	1	0.399	0.004	0.023	1.972	0	0.013	0.572	0.413	1	0.01	0.576	0.013	1.399	0	0.027	1.573	0.399	0.004	0.01	0.985	1	0.01
10	1	0	0.449	1	0.425	0.054	0	1.82	0	0.054	0.395	0.425	1	0	0.449	0	1.425	0	0.054	1.395	0.425	0.054	0	0.82
11	1.339	0.003	1	0.039	0.661	1.668	0.002	0.067	0.667	1.635	0.002	0.028	0.706	0.635	1.002	0	0.7	0.668	1.002	0.039	0.028	2.3	0.003	0.028
12	2.229	0.039	0.07	0.714	0.74	1.024	0.015	0.964	1.017	0.528	0.097	0.2	1.703	0.543	0.088	0.015	1.426	1.039	0.006	0.779	0.214	1.528	0.032	0.265

Table 8.8: Chord likelihoods for 'Tread The Water', pure instruments mixes

8.2.3 'Impure Instruments' Case

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.734	0.758	1.49	0.379	0.924	1.275	0.661	0.53	0.591	2.046	0.513	0.67	0.463	1.196	1.475	0.523	0.796	0.424	1.623	0.383	0.653	1.712	0.647	0.674
2	1.162	0.458	0.575	0.304	1.619	0.334	0.345	1.015	0.487	1.229	0.669	0.89	0.418	1.292	0.647	0.388	1.549	0.397	0.323	0.513	0.875	1.168	0.417	1.099
3	0.195	0.188	0.478	0.609	1.158	0.088	0.496	1.262	0.484	0.285	0.455	1.684	0.494	0.285	0.313	0.826	1.168	0.088	0.354	0.404	1.457	0.185	0.331	1.478
4	1.036	1.752	0.76	0.533	1.449	0.462	1.136	0.973	1.137	1.712	0.801	0.726	0.738	2.218	0.835	0.506	1.05	0.968	1.17	0.753	1.152	0.928	1.212	0.946
5	0.781	1.788	0.329	0.321	1.184	0.475	1.039	0.384	0.888	1.652	0.628	0.445	0.449	2.093	0.632	0.354	0.745	0.915	1.044	0.292	0.852	0.779	1.343	0.416
6	1.052	0.167	0.295	0.127	1.325	0.06	0.157	0.388	0.285	1.037	0.187	0.449	0.174	1.154	0.188	0.241	1.214	0.177	0.157	0.18	0.447	1.047	0.05	0.502
7	0.931	0.83	1.525	0.42	0.931	1.12	0.858	0.581	0.524	2.362	0.358	0.567	0.523	1.502	1.358	0.466	0.93	0.26	1.858	0.48	0.523	1.792	0.69	0.627
8	0.263	0.106	0.896	0.467	1.188	0.095	0.746	1.251	0.511	0.293	0.258	2.078	0.52	0.279	0.264	1.093	1.196	0.082	0.752	0.265	1.445	0.268	0.114	1.876
9	0.129	0.438	1.387	0.128	0.201	0.21	1.304	0.282	0.253	0.478	0.347	1.11	0.147	0.471	0.438	1.126	0.094	0.203	1.394	0.298	0.218	0.242	0.355	1.28
10	0.511	1.692	0.644	0.455	0.914	0.369	1.236	0.501	1.061	1.365	0.644	0.719	0.585	1.837	0.636	0.662	0.438	0.842	1.228	0.444	0.988	0.514	1.228	0.708
11	0.135	0.021	0.452	0.346	1.123	0.033	0.371	1.094	0.355	0.151	0.108	1.675	0.35	0.14	0.111	0.688	1.118	0.022	0.374	0.108	1.338	0.151	0.03	1.436
12	0.184	1.241	0.255	0.426	0.82	0.067	1.06	0.645	0.607	1.19	0.242	0.938	0.426	1.364	0.252	0.482	0.64	0.241	1.07	0.189	1.063	0.19	1.057	0.701
13	0.834	0.287	1.393	0.358	1.171	1.118	0.268	0.936	0.435	1.858	0.358	0.794	0.473	0.912	1.358	0.304	1.209	0.172	1.268	0.447	0.81	1.742	0.233	0.883
14	1.078	0.326	0.362	0.261	1.482	0.085	0.296	0.647	0.449	1.212	0.281	0.719	0.339	1.321	0.281	0.346	1.372	0.194	0.297	0.275	0.743	1.079	0.216	0.733
15	0.296	0.333	0.992	0.641	1.14	0.215	0.924	1.418	0.651	0.371	0.48	2.107	0.687	0.374	0.439	1.182	1.175	0.219	0.883	0.493	1.531	0.257	0.371	1.959
16	0.562	1.547	0.445	0.616	0.843	0.353	1.035	0.518	1.146	1.378	0.574	0.727	0.789	1.746	0.582	0.631	0.485	0.72	1.043	0.422	1.07	0.552	1.171	0.534

17	0.166	0.278	0.689	0.439	1.074	0.112	0.713	1.275	0.472	0.322	0.317	1.801	0.48	0.322	0.285	0.843	1.083	0.113	0.682	0.317	1.389	0.156	0.31	1.678
18	0.959	0.643	1.459	0.345	1.142	1.202	0.447	0.8	0.469	2.175	0.533	0.532	0.447	1.298	1.532	0.268	1.12	0.325	1.446	0.536	0.63	1.857	0.52	0.723

Table 8.9: Chord likelihoods for ‘Chelsea’, impure instruments mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.966	1.952	0.887	0.991	1.011	0.678	1.455	1.014	1.34	1.347	0.57	0.76	1.151	1.888	0.707	0.744	0.822	1.218	1.591	0.998	1.196	0.613	1.275	0.767
2	1.14	1.475	1.042	1.435	1.316	0.972	1.071	0.892	1.741	0.911	0.834	0.922	1.34	1.477	0.857	1.242	0.915	1.538	1.093	1.212	1.515	0.974	0.886	0.699
3	1.43	1.547	1.379	2.502	1.302	1.207	1.531	1.357	2.043	1.325	1.344	1.654	2.101	1.518	1.273	2.125	1.36	1.401	1.46	1.827	1.973	1.179	1.424	0.98
4	1.45	1.756	1.151	1.683	1.107	1.633	1.195	1.41	1.251	0.91	1.818	1.174	1.539	1.097	1.671	1.372	1.396	1.82	1.048	1.608	1.196	0.974	1.715	1.1
5	0.664	1.684	0.502	0.935	0.693	0.643	1.645	0.46	1.03	1.282	0.619	0.493	0.796	1.61	0.415	0.837	0.459	0.971	1.442	0.804	0.825	0.569	1.559	0.362
6	0.641	1.412	0.507	0.846	1.132	0.616	0.627	0.466	1.762	0.592	0.367	0.677	1.023	1.384	0.434	0.749	0.392	1.408	0.694	0.538	1.514	0.588	0.554	0.369
7	0.774	0.802	0.71	1.867	0.769	0.561	0.889	0.709	1.783	0.485	0.852	1.345	1.625	0.829	0.566	1.75	0.611	0.905	0.602	1.114	1.62	0.588	0.744	0.592
8	0.722	1.505	0.562	0.675	0.448	1.385	0.469	0.657	0.748	0.344	1.395	0.324	0.781	0.577	1.38	0.46	0.48	1.618	0.454	0.792	0.507	0.456	1.287	0.441
9	0.52	1.476	0.331	0.403	0.347	0.566	1.15	0.292	0.774	1.151	0.269	0.292	0.662	1.347	0.365	0.32	0.234	0.762	1.246	0.319	0.488	0.437	1.184	0.208
10	0.806	1.478	0.767	1.275	1.224	0.748	1.046	0.652	1.671	0.684	0.804	0.84	1.086	1.403	0.643	1.196	0.64	1.466	0.884	1.008	1.504	0.673	0.922	0.573
11	0.845	0.689	0.575	1.947	0.521	0.588	0.943	0.658	1.707	0.418	0.826	1.287	1.751	0.66	0.453	1.762	0.565	0.83	0.57	1.134	1.427	0.559	0.821	0.473
12	0.755	1.514	0.046	0.576	0.226	1.342	0.403	0.373	0.684	0.468	1.066	0.211	0.88	0.6	1.046	0.24	0.422	1.474	0.383	0.402	0.351	0.427	1.403	0.036
13	0.779	1.99	1.078	0.764	0.775	0.973	1.575	0.808	0.896	1.509	0.544	0.552	0.807	1.697	0.925	0.603	0.686	1.161	1.955	0.859	0.803	0.679	1.422	0.647
14	0.615	0.84	0.596	1.929	0.45	0.421	0.963	0.709	1.467	0.555	0.834	1.202	1.558	0.749	0.557	1.648	0.541	0.615	0.685	1.155	1.392	0.33	0.924	0.427
15	0.913	1.934	0.749	1.013	0.915	0.734	1.583	0.853	1.163	1.342	0.821	0.682	1.007	1.805	0.729	0.842	0.759	1.197	1.491	1.013	1.009	0.605	1.563	0.682
16	0.333	1.472	0.013	0.054	0.859	0.037	1.013	0.091	0.543	1.306	0.01	0.157	0.081	1.768	0.01	0.066	0.397	0.499	1.013	0	0.607	0.333	1.009	0.104

Table 8.10: Chord likelihoods for ‘Natalie Grey’, impure instruments mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	1.532	1.905	1.302	1.713	1.936	1.457	1.273	1.488	2.092	1.37	1.449	1.451	1.678	1.882	1.436	1.496	1.522	1.969	1.259	1.533	2.082	1.434	1.406	1.271
2	1.968	1.764	1.813	2.781	1.951	1.936	1.866	2.075	2.374	1.673	2.194	2.13	2.423	1.71	1.934	2.455	2.001	1.973	1.606	2.399	2.407	1.881	1.988	1.748
3	1.698	1.748	1.687	2.072	1.634	2.032	1.868	1.649	1.764	1.65	2.349	1.504	1.713	1.558	2.004	1.98	1.582	1.94	1.523	2.124	1.648	1.842	2.185	1.557
4	2.356	1.724	1.604	2.377	1.814	2.244	1.591	1.901	2.552	1.663	1.851	1.919	2.678	1.687	1.782	2.033	1.94	2.268	1.522	2.016	2.136	2.206	1.77	1.558
5	1.439	1.889	1.364	2.028	1.839	1.506	1.467	1.434	2.246	1.325	1.548	1.657	1.835	1.779	1.44	1.862	1.427	1.961	1.359	1.638	2.235	1.396	1.543	1.268
6	1.419	1.537	1.317	2.57	1.586	1.461	1.618	1.44	2.31	1.313	1.754	1.645	2.065	1.527	1.388	2.229	1.341	1.675	1.253	2.024	2.232	1.451	1.689	1.099

7	1.271	1.277	1.488	1.808	1.214	1.54	1.613	1.453	1.441	1.234	2.116	1.075	1.349	1.167	1.668	1.683	1.122	1.473	1.164	2.061	1.293	1.43	1.793	1.328
8	2.13	2.04	1.671	2.32	1.613	2.449	1.814	1.825	2.48	1.677	2.237	1.839	2.535	1.672	2.069	2.086	1.668	2.444	1.647	2.073	2.018	2.08	2.213	1.591
9	0.867	1.384	0.659	1.287	1.351	0.922	0.618	0.862	1.858	0.632	0.854	0.803	1.25	1.284	0.813	0.964	0.743	1.575	0.577	1.023	1.734	0.823	0.772	0.539
10	1.89	1.211	1.475	2.906	1.295	1.498	1.594	2.063	2.087	1.103	1.923	1.867	2.576	1.108	1.505	2.301	1.785	1.503	1.176	2.498	1.982	1.395	1.624	1.458
11	1.399	1.335	1.434	1.844	1.303	1.802	1.565	1.596	1.548	1.122	2.334	1.305	1.519	1.042	1.809	1.756	1.273	1.722	1.04	2.047	1.423	1.509	1.941	1.508
12	2.256	1.658	1.677	2.39	1.548	2.241	1.681	1.979	2.386	1.511	2.027	1.899	2.655	1.457	1.867	2.074	1.817	2.188	1.521	2.153	1.947	2.041	1.871	1.662
13	0.502	1.244	0.477	0.859	1.12	0.627	0.41	0.395	1.322	0.4	0.528	0.427	0.665	1.088	0.576	0.66	0.463	1.315	0.457	0.628	1.283	0.471	0.509	0.197
14	0.799	0.305	0.276	2.023	0.149	0.753	0.798	0.516	1.608	0.147	1.088	1.047	1.792	0.157	0.43	1.744	0.333	0.763	0.141	1.213	1.143	0.605	0.953	0.237
15	0.227	0.162	0.227	1.009	0.082	0.485	1	0.213	0.222	0.103	1.286	0.084	0.229	0	0.389	1.002	0.089	0.383	0.103	1.131	0.084	0.323	1.162	0.206
16	1.99	0.471	0.438	1.62	0.591	1.318	0.486	1.399	1.635	0.241	0.535	0.826	2.342	0.339	0.43	0.794	1.299	1.415	0.381	1.368	0.944	1.185	0.478	0.574

Table 8.11: Chord likelihoods for ‘Good Time’, impure instruments mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.693	1.042	0.37	0.057	0.304	1.77	0.025	0.023	0.463	0.654	1.007	0.042	0.454	0.314	1.367	0.043	0.295	1.43	0.385	0.024	0.065	1.042	1.022	0.009
2	1.148	0.111	1.026	1.037	0.476	0.289	0.039	2.404	0.169	0.106	1.032	0.512	1.169	0.038	1.099	0.052	1.476	0.221	0.106	1.944	0.497	0.216	0.112	1.419
3	1.414	0.461	0.813	0.267	1.193	1.506	0.18	0.109	0.633	1.769	0.477	0.261	0.542	1.115	1.113	0.298	1.102	0.852	0.816	0.146	0.321	2.16	0.48	0.14
4	2.486	1.04	0.493	0.585	1.068	2.021	0.195	0.892	0.981	1.132	1.298	0.222	1.516	1.051	1.359	0.174	1.603	1.94	0.255	0.844	0.098	2.032	1.06	0.48
5	1.025	0.093	1	0.111	1.109	1.176	0.022	0.172	0.169	1.94	0.115	0.256	0.172	0.94	1.093	0.109	1.112	0.176	1	0.025	0.256	2.022	0.115	0.169
6	0.23	0.101	0.712	0.195	1.05	0.147	0.14	1.712	0.053	0.122	0.633	1.142	0.212	0.074	0.677	0.146	1.209	0.099	0.184	0.717	1.031	0.119	0.105	1.663
7	0.469	1.014	0.236	0.051	0.376	1.324	0.075	0.102	0.163	0.477	1.013	0.138	0.184	0.308	1.178	0.079	0.396	1.155	0.24	0.043	0.09	0.618	1.017	0.13
8	2.087	0.313	0.343	0.66	0.865	1.34	0.125	0.711	1.265	0.805	0.52	0.422	1.613	0.758	0.549	0.342	1.213	1.294	0.154	0.632	0.391	1.785	0.331	0.394
9	0.507	1	0.265	0.021	0.074	1.683	0.001	0.028	0.418	0.331	1	0.008	0.44	0.067	1.264	0.001	0.095	1.419	0.265	0.021	0.007	0.749	1	0.008
10	1.049	0.014	0.503	1.004	0.24	0.135	0	1.65	0.042	0.093	0.435	0.233	1.042	0.011	0.517	0.004	1.24	0.052	0.083	1.42	0.233	0.131	0.014	0.649
11	1.014	0.164	1.083	0.049	0.711	1.751	0.158	0.37	0.642	1.408	0.141	0.44	0.66	0.408	1.115	0.114	0.729	0.751	1.132	0.044	0.357	1.996	0.189	0.435
12	2.286	0.871	0.515	0.482	1.199	1.941	0.206	0.749	1.123	1.229	1.174	0.298	1.376	1.096	1.243	0.238	1.452	1.808	0.275	0.69	0.289	2.166	0.934	0.505

Table 8.12: Chord likelihoods for ‘Tread The Water’, impure instruments mixes

8.2.4 'Pure Vocals' Case

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.211	0.854	0	0.102	1.264	0.213	0.692	1	0.474	0.589	0.104	1	0.211	0.853	0.002	0.102	1	0.476	0.589	0.102	1.264	0.211	0.693	1
2	0.729	0.939	1.464	0.595	0.668	0.945	1.353	0.697	0.349	0.798	1.078	1.374	0.517	0.676	1.125	1.344	0.836	0.824	1.399	0.667	0.456	0.683	1.014	1.447
3	0.889	0.744	1.008	0.489	0.763	0.436	1.44	0.127	0.237	1.085	0.555	1.237	0.363	1.078	0.437	1.363	0.889	0.429	1.322	0.253	0.237	0.77	0.869	1.001
4	1.834	1.942	1.345	1.438	2.307	1.548	1.017	1.287	1.857	1.517	1.391	1.811	1.535	2.033	1.572	1.563	1.985	2.064	1.198	1.039	2.008	1.639	1.244	1.412
5	0.676	1.064	0.052	0.016	0.682	0.059	1	0.052	0.024	1.675	0.109	0.016	0.017	1.682	0.109	0.016	0.675	0.066	1	0.052	0.023	0.676	1.057	0.052
6	1.106	0.63	0.725	0.857	1.287	0.434	0.766	0.219	0.905	1.022	0.455	1.565	0.836	1.196	0.435	1.475	1.218	0.608	0.746	0.128	1.017	1.001	0.476	0.836
7	1.198	0.84	1.083	0.545	1.388	0.307	1.095	0.551	0.623	0.901	0.251	1.179	0.529	1.363	0.308	1.099	1.294	0.769	1.153	0.471	0.719	0.83	0.32	1.105
8	1.028	0.822	1	0.142	0.914	0.743	1.078	0.167	0.002	0.965	0.743	1.026	0.142	0.967	0.743	1	1.054	0.744	1.078	0.142	0.027	0.887	0.821	1.026
9	0.351	1.084	1.557	0.4	1.088	1.188	0.739	0.046	1.247	1.328	0.168	0.841	0.421	1.201	1.098	0.911	0.261	1.061	1.67	0.116	1.158	1.304	0.281	0.557
10	1.271	0.826	1.007	0.495	1.313	0.473	1.367	0.242	0.408	1.242	0.583	1.049	0.335	1.469	0.411	1.24	1.24	0.701	1.194	0.434	0.376	1.116	0.771	0.988
11	1.025	0.678	0.181	1.305	1.566	0.186	0.366	0.582	1.46	0.954	0.389	1.413	1.201	1.365	0.235	1.235	1.307	0.596	0.212	0.404	1.742	0.873	0.42	0.512
12	0.986	0.124	0.255	1.012	1.189	0	0.264	0.082	1.124	0.986	0.014	1.331	1	1.11	0.003	1.264	1.065	0.124	0.252	0.014	1.203	0.986	0.011	0.334
13	0.455	0.14	0.69	0.858	1.356	0.365	0.867	1.16	0.544	0.458	0.633	1.745	0.519	0.357	0.348	1.209	1.331	0.263	0.582	0.624	1.42	0.582	0.526	1.511
14	0.473	0.665	0.267	0.641	1.647	0.349	0.415	1.144	1.217	0.209	0.158	1.544	0.768	0.724	0.08	0.693	1.197	0.864	0.337	0.293	1.941	0.408	0.228	1.196
15	0.258	0.325	1.027	0.21	0.322	0.218	1.17	0.527	0.056	0.194	0.15	1.314	0.258	0.178	0.174	1	0.524	0.203	1.194	0.213	0.322	0.072	0.317	1.317
16	0.956	0.975	0.584	1.019	1.688	0.725	0.407	0.49	1.351	1.05	0.722	1.786	1.01	1.299	0.813	1.404	1.347	0.974	0.498	0.107	1.742	1.048	0.635	0.875
17	0.536	1.264	0.274	0.637	1.466	0.293	1.24	1.19	0.874	1.309	0.505	1.392	0.657	1.516	0.298	0.66	1.248	0.501	1.033	0.458	1.586	0.546	1.264	1.213
18	0.534	0.147	0.304	0.324	1.514	0.126	0.36	1.029	0.45	0.488	0.048	1.527	0.376	0.558	0.034	0.57	1.44	0.196	0.346	0.071	1.355	0.537	0.09	1.275

Table 8.13: Chord likelihoods for 'Chelsea', pure vocals mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.334	1.338	0.179	0.016	0.309	0.42	1.014	0.22	0.592	1.009	0.262	0.049	0.334	1.26	0.257	0.014	0.051	0.67	1.009	0.185	0.309	0.342	1.093	0.219
2	1.598	0.383	0.701	1.563	0.695	1.574	1.053	0.765	1.31	0.757	1	0.256	1.489	0.492	0.648	1.053	0.874	1.31	0.701	1.563	0.586	1.684	1	0.256
3	0.549	1.322	0.518	0.862	1.332	1.128	0.945	0.423	1.462	0.683	0.932	0.333	0.549	1.17	0.67	0.776	0.419	1.614	0.683	0.866	1.332	0.976	1.097	0.336
4	0.2	1	0.026	0.724	1.107	0.005	0.296	0.308	1.247	0.005	0.279	0.373	0.447	1	0.007	0.543	0.306	1	0.025	0.478	1.354	0.005	0.277	0.128
5	0.352	0.618	0.379	1.052	0.796	0.769	1.056	0.272	0.928	0.426	1.043	0.328	0.404	0.576	0.417	1.056	0.272	0.919	0.43	1	0.848	0.727	1.094	0.276
6	0.279	1.317	1.102	1.239	0.301	0.642	1.421	0.754	1.379	1.357	1.185	0.813	1.088	1.301	1.12	1.233	0.01	0.586	1.356	1.174	1.11	0.626	1.439	0.748

7	0.827	1.053	0.286	0.929	1.534	0.815	0.782	0.783	1.578	0.237	0.706	0.61	0.827	1.027	0.237	0.755	0.783	1.604	0.313	0.929	1.534	0.788	0.732	0.61
8	0.039	1.025	0.249	0.014	0.036	1.215	0.05	0.011	0.054	0.218	1.011	0.057	0.035	0.043	1.205	0.057	0.018	1.04	0.244	0.011	0.032	0.232	1.007	0.055
9	0.805	1.117	0	0.062	0.147	0.744	1.001	0.091	0.86	1	0.001	0.03	0.805	1.117	0	0.001	0.091	0.86	1	0.062	0.147	0.744	1.001	0.03
10	0.714	0.618	0.3	1.047	1.002	1.086	1.156	0.743	0.992	0.361	1.195	0.716	0.714	0.442	0.438	1.038	0.725	1.168	0.399	1.066	1.002	0.91	1.293	0.735
11	1.312	0.545	0.382	1.389	1.259	1.228	1.165	1.526	1.212	0.057	1.419	1.131	1.312	0.293	0.472	1.161	1.359	1.464	0.219	1.556	1.259	0.976	1.256	1.298
12	0.223	1.405	0.005	0.039	0.436	1.167	0.004	0.025	0.585	0.031	1	0.018	0.205	0.436	1	0.018	0.056	1.572	0.005	0.025	0.419	0.198	1	0.004
13	0.065	2.202	0.012	0	0.58	0.699	1	0	0.645	1.012	0.622	0	0.065	1.58	0.634	0	0	1.267	1.012	0	0.58	0.077	1.622	0
14	0.399	0.164	0.088	2.158	0	0.179	0.758	0.472	1	0.015	0.995	1	1.399	0	0.252	1.758	0.399	0.164	0.015	1.23	1	0.015	0.922	0.073
15	0	1.323	0.005	0	0.206	0.117	1	0.005	0.206	1	0.122	0	0	1.206	0.122	0	0	0.323	1	0.005	0.206	0	1.117	0.005
16	0.38	1.195	0	0.011	0.669	0	1	0.094	0.206	1.38	0	0.104	0.011	1.575	0	0.011	0.474	0.195	1	0	0.3	0.38	1	0.094

Table 8.14: Chord likelihoods for 'Natalie Grey', pure vocals mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	1.893	1.521	1.232	1.542	1.4	1.44	1.283	1.378	1.493	0.731	1.056	1.47	1.774	1.213	1.022	1.45	1.681	1.921	1.25	1.358	1.281	1.131	1.073	1.286
2	1.399	0.27	0.391	2.225	0.957	0.231	1.02	1.444	0.486	0.631	1.606	0.764	1.225	0.665	0.606	1.468	1.696	0.266	0.02	2.148	0.783	0.627	1.236	0.687
3	0.521	1.001	0.322	0.328	0.292	1.353	0.115	0.102	0.56	0.402	1.1	0.441	0.56	0.286	1.212	0.438	0.292	1.236	0.227	0.1	0.331	0.638	1.005	0.212
4	1.211	0.473	0.273	1.254	0.211	0.775	0.486	0.496	1.761	0.662	0.266	1.029	2.006	0.663	0.263	1.032	0.457	0.776	0.482	0.498	1.006	0.965	0.475	0.273
5	1	0.64	1.216	1.768	0.599	0.411	1.549	1.313	0.475	0.351	1.07	0.853	1	0.527	0.6	1.497	1.124	0.587	1.079	1.957	0.599	0.298	0.933	1.042
6	0.358	0.936	0.459	1.35	0.988	0.484	1	0.419	0.927	0.459	1.017	0.069	0.358	0.919	0.476	1	0.419	0.944	0.459	1.35	0.988	0.467	1.017	0.069
7	0.001	0.42	1.294	0	0	0.715	0	1	0.001	0.294	1.42	0	0.001	0	1.714	0	0	0.421	0.294	1	0	0.295	0.42	1
8	0.331	0	0	1.532	0	0	0.201	0.331	1	0	0.201	1	1.331	0	0	1.201	0.331	0	0	0.532	1	0	0.201	0
9	0.495	0.651	0.647	1.489	0.439	0.576	1.038	0.895	0.37	0.347	1.622	0.101	0.496	0.395	0.916	1.023	0.566	0.624	0.333	1.818	0.44	0.32	1.308	0.43
10	1.205	1.109	0.456	1.246	0.248	1.605	0.493	1.028	0.764	0.199	1.745	0.299	1.308	0.182	1.417	0.651	0.792	1.588	0.165	1.38	0.35	0.678	1.454	0.432
11	0.617	0.889	0.325	1.186	0.838	1.085	1.046	0.409	0.679	0.568	1.616	0.392	0.459	0.605	0.874	1.118	0.618	1.122	0.303	1.135	0.68	0.801	1.595	0.341
12	0.318	0.161	0.321	1.64	0	0.059	0.451	0.622	1.011	0.112	0.697	1.006	1.318	0.112	0.364	1.339	0.307	0.059	0.118	0.955	1	0.011	0.493	0.321
13	0.395	1	0.041	0.496	1	0.027	0.033	0.376	1.154	0	0.008	0.16	0.522	1	0.008	0.16	0.368	1.027	0.033	0.376	1.128	0.027	0	0.041
14	1.666	1.59	1.71	1.909	1.855	1.742	1.696	1.83	1.18	1.554	1.445	1.576	1.435	1.407	1.598	1.606	2.11	1.594	1.85	1.859	1.624	1.558	1.584	1.527
15	0.496	1.004	0.041	0.147	0.011	1.51	0.02	0	0.632	0.04	1	0.163	0.632	0.015	1.025	0.163	0.011	1.485	0.045	0	0.147	0.521	1.004	0.016
16	1.669	1.415	1.021	1.178	1.224	1.573	1.28	1.43	1.858	1.233	0.884	1.06	1.828	1.387	0.943	0.885	1.194	1.727	1.339	1.255	1.383	1.545	1.201	1.137

Table 8.15: Chord likelihoods for ‘Good Time’, pure vocals mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	1.092	0.864	1.232	0.984	1.343	1.011	0.358	0.657	1.061	1.653	0.326	0.379	0.807	1.426	1.145	0.559	1.089	0.784	1.177	0.837	1.058	1.573	0.271	0.232
2	1.701	0.089	0.321	0.62	0.664	1	0.141	1.381	1.036	0.202	0.294	0.599	1.602	0.213	0.276	0.089	1.23	1.011	0.123	0.87	0.565	1.124	0.095	0.85
3	1.183	0.468	1.031	0.429	0.647	1.756	0.58	0.34	0.883	1.653	0.331	0.228	0.884	0.774	1.065	0.307	0.648	0.876	1.314	0.419	0.348	2.061	0.614	0.218
4	1.311	0.513	1.717	1.667	0.491	0.983	0.644	1.951	1.108	0.681	1.467	0.423	1.579	0.526	1.604	0.86	0.963	0.828	0.781	2.387	0.759	0.995	0.531	1.143
5	1.059	0.255	0.033	0.271	1.337	0.268	0.086	0.345	0.26	1	0.336	0.479	0.235	1.035	0.252	0.262	1.313	0.304	0.001	0.127	0.513	1.048	0.305	0.335
6	0.574	1.156	1.436	0.933	0.76	0.721	0.86	1.041	1.19	1.489	1.348	0.698	0.869	1.41	1.58	0.897	0.439	0.641	1.091	1.24	1.054	0.974	1.004	1.005
7	0.784	1	0.335	0	0.784	1	0.335	0	0	0.784	1	0.335	0	0.784	1	0.335	0.784	1	0.335	0	0	0.784	1	0.335
8	1.411	1.088	1.227	0.554	1.589	1.275	0.916	1.855	0.856	1.418	1.817	1.046	0.856	1.321	1.657	0.566	1.589	1.178	0.755	1.375	1.034	1.508	1.345	1.867
9	1.733	1.34	0.713	1.351	1.02	1.086	0.754	1.362	1.288	1.096	0.607	0.516	1.685	1.343	0.83	0.498	1.418	1.333	0.977	1.344	0.972	1.089	0.871	0.508
10	0.923	0.517	1.29	0.494	0.343	0.75	0.479	1.222	0.85	0.851	1.06	0.419	0.978	0.706	1.266	0.387	0.471	0.605	0.685	1.19	0.398	0.94	0.455	1.115
11	0.964	0.388	0.603	0.133	1.163	1.096	0.528	1.201	1.221	0.359	0.231	1.367	1.061	0.388	0.333	0.4	1.003	1.124	0.63	0.234	1.26	1.096	0.258	1.468
12	1.668	0.024	0.004	0.185	0.558	1	0.089	0.132	1.001	0.574	0.069	0.01	1.12	0.574	0.004	0.066	0.677	1	0.024	0.188	0.01	1.549	0.089	0.013

Table 8.16: Chord likelihoods for ‘Tread The Water’, pure vocals mixes

8.2.5 ‘Impure Vocals’ Case

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.386	1.058	1.055	0.363	1.425	1.006	0.86	1.308	0.817	1.634	0.41	1.262	0.52	1.184	1.056	0.363	1.128	0.556	1.506	0.409	1.559	1.132	0.861	1.308
2	1.185	0.539	0.974	0.276	1.265	0.459	0.669	0.837	0.45	1.261	0.851	0.935	0.452	1.262	0.847	0.676	1.267	0.459	0.665	0.578	0.533	1.182	0.542	1.237
3	1.214	1.05	1.309	0.94	1.607	0.722	1.747	1.185	0.7	1.391	1.086	2.244	0.86	1.386	0.852	1.772	1.767	0.718	1.513	0.712	1.253	1.059	1.289	2.016
4	1.734	2.136	1.188	1.083	1.877	1.018	1.43	1.561	1.382	2.064	1.379	1.435	1.356	2.471	1.433	1.046	1.85	1.425	1.484	1.171	1.499	1.354	1.675	1.524
5	0.488	1.051	0.037	0.015	0.494	0.045	1	0.037	0.025	1.486	0.08	0.015	0.017	1.494	0.08	0.015	0.486	0.053	1	0.037	0.023	0.488	1.043	0.037
6	1.15	0.455	0.681	0.733	1.2	0.335	0.588	0.338	0.773	1.047	0.443	1.241	0.768	1.161	0.438	1.161	1.195	0.449	0.583	0.258	0.819	1.041	0.346	0.766
7	1.362	1.075	1.881	0.832	1.469	0.877	1.32	0.895	0.937	1.693	0.542	1.432	0.875	1.657	1.107	1.343	1.408	0.841	1.885	0.805	0.983	1.459	0.546	1.406
8	1.195	0.972	1.242	0.372	1.53	0.838	1.164	1.009	0.286	1.084	0.991	1.727	0.443	1.128	0.998	1.162	1.688	0.882	1.172	0.444	0.779	0.994	0.921	1.799
9	0.674	1.378	2.392	0.705	1.29	1.361	1.375	0.585	1.635	1.717	0.664	1.528	0.847	1.598	1.589	1.557	0.502	1.242	2.299	0.615	1.463	1.581	0.572	1.438
10	1.363	1.288	1.24	0.707	1.332	0.555	1.76	0.527	0.785	1.732	0.82	1.248	0.683	1.965	0.678	1.405	1.23	0.788	1.618	0.684	0.652	1.232	1.197	1.225

11	1.037	0.679	0.401	1.307	1.946	0.202	0.498	1.223	1.471	0.984	0.499	1.948	1.264	1.352	0.357	1.288	1.74	0.57	0.355	0.563	2.174	0.875	0.453	1.204
12	0.941	0.544	0.346	1.011	1.192	0	0.672	0.228	1.126	1.36	0.114	1.367	1	1.486	0.103	1.253	1.066	0.126	0.661	0.114	1.25	0.941	0.429	0.471
13	0.651	0.321	1.263	0.966	1.476	0.856	0.917	1.419	0.742	1.018	0.868	1.825	0.726	0.575	0.989	1.257	1.459	0.413	1.038	0.851	1.55	1.111	0.643	1.71
14	1.04	0.817	0.525	0.867	2.19	0.388	0.579	1.38	1.451	0.894	0.391	1.769	1.006	1.404	0.322	0.924	1.745	0.898	0.51	0.536	2.156	0.974	0.376	1.438
15	0.177	0.313	1.178	0.189	1.01	0.221	1.136	1.304	0.17	0.193	0.334	2.043	0.261	0.185	0.348	1.046	1.101	0.214	1.15	0.307	1.095	0.093	0.306	2.161
16	0.946	1.301	0.727	1.013	1.493	0.653	0.805	0.657	1.433	1.407	0.824	1.625	1.103	1.666	0.894	1.319	1.163	0.912	0.876	0.351	1.649	1.017	0.972	0.963
17	0.419	0.86	0.321	0.472	1.355	0.262	0.877	1.257	0.671	0.918	0.441	1.36	0.527	1.039	0.311	0.512	1.211	0.383	0.748	0.41	1.463	0.442	0.868	1.298
18	0.782	0.482	1.142	0.516	1.661	0.838	0.574	1.386	0.715	1.289	0.503	1.693	0.649	0.89	0.976	0.746	1.595	0.439	1.047	0.439	1.529	1.246	0.408	1.616

Table 8.17: Chord likelihoods for ‘Chelsea’, impure vocals mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.564	1.526	0.512	0.362	0.542	0.507	1.177	0.565	0.812	1.155	0.384	0.296	0.616	1.467	0.467	0.261	0.345	0.819	1.26	0.53	0.594	0.448	1.131	0.464
2	2.137	1.346	1.51	2.114	1.437	1.849	1.652	1.369	2.091	1.375	1.197	0.843	2.083	1.508	1.133	1.58	1.429	1.982	1.589	2.106	1.384	2.011	1.276	0.836
3	1.112	1.547	1.009	1.596	1.487	1.211	1.241	0.984	1.906	1.048	1.048	1.05	1.397	1.502	0.955	1.382	0.978	1.665	1.148	1.316	1.772	1.167	1.187	0.771
4	1.143	1.893	0.952	1.768	1.51	1.095	1.105	1.257	1.708	0.735	1.503	1.174	1.418	1.482	1.248	1.446	1.22	1.843	0.85	1.529	1.785	0.685	1.401	0.936
5	0.861	1.527	0.828	1.516	1.163	1.137	1.877	0.712	1.441	1.298	1.184	0.758	0.996	1.497	0.732	1.421	0.718	1.336	1.425	1.375	1.298	1.107	1.781	0.618
6	0.617	2.067	1.355	1.397	1.155	0.888	1.519	1.023	2.015	1.495	1.232	0.991	1.271	2.017	1.307	1.323	0.411	1.411	1.593	1.355	1.809	0.839	1.47	0.949
7	1.256	1.332	0.904	1.845	1.594	0.979	1.198	1.22	2.16	0.661	1.131	1.354	1.689	1.37	0.73	1.642	1.124	1.688	0.797	1.509	2.028	1.017	1.024	1.017
8	0.852	1.572	0.994	1.14	0.654	1.442	0.986	0.972	0.766	0.727	1.692	0.802	0.942	0.766	1.617	1.02	0.831	1.481	0.911	1.19	0.745	0.636	1.609	0.852
9	0.614	1.149	0.094	0.152	0.162	0.528	1.024	0.172	0.648	1.045	0.026	0.057	0.633	1.148	0.071	0.043	0.147	0.63	1.069	0.158	0.181	0.527	1.001	0.064
10	1.301	1.666	1.139	1.652	1.71	1.58	1.769	1.257	1.81	1.03	1.585	1.228	1.329	1.449	1.101	1.623	1.229	1.999	1.285	1.652	1.738	1.363	1.731	1.228
11	1.657	1.057	0.926	2.216	1.389	1.383	1.613	1.744	1.93	0.536	1.638	1.682	2.079	0.859	0.843	1.938	1.538	1.706	0.817	2	1.811	1.185	1.53	1.466
12	0.662	1.573	0.27	0.603	0.489	1.314	0.355	0.507	0.725	0.452	1.153	0.301	0.749	0.675	1.209	0.305	0.512	1.537	0.411	0.511	0.575	0.416	1.294	0.209
13	0.443	2.05	0.498	0.388	0.634	0.74	1.192	0.425	0.689	1.27	0.452	0.192	0.443	1.634	0.722	0.192	0.388	1.105	1.462	0.425	0.634	0.324	1.416	0.228
14	0.973	0.965	0.852	2.158	0.599	0.722	1.237	0.977	1.602	0.633	1.212	1.434	1.815	0.817	0.803	1.938	0.812	0.907	0.829	1.481	1.441	0.575	1.188	0.757
15	0.14	1.334	0.088	0.165	0.261	0.121	1.044	0.161	0.261	1.047	0.128	0.025	0.14	1.261	0.146	0.049	0.14	0.334	1.063	0.185	0.261	0.048	1.102	0.045
16	0.351	1.185	0	0.01	0.622	0	1	0.086	0.196	1.351	0	0.097	0.01	1.536	0	0.01	0.437	0.185	1	0	0.282	0.351	1	0.086

Table 8.18: Chord likelihoods for ‘Natalie Grey’, impure vocals mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	1.952	1.945	1.451	1.791	1.951	1.622	1.424	1.67	2.078	1.258	1.396	1.616	1.946	1.832	1.397	1.594	1.819	2.197	1.425	1.648	1.945	1.509	1.37	1.472
2	1.762	1.17	1.374	2.608	1.577	1.235	1.57	1.994	1.635	1.311	1.989	1.576	1.981	1.331	1.471	2.04	1.924	1.255	1.053	2.459	1.797	1.396	1.668	1.427
3	1.327	1.681	1.268	1.516	1.252	2.017	1.197	1.095	1.517	1.292	2.037	1.271	1.427	1.186	1.947	1.525	1.162	1.911	1.108	1.349	1.352	1.522	1.876	1.104
4	1.839	1.179	1.088	2.032	1.297	1.531	1.139	1.386	2.17	1.307	1.2	1.687	2.351	1.306	1.164	1.728	1.479	1.531	1.104	1.428	1.809	1.658	1.216	1.082
5	2.093	1.967	2.021	2.445	1.95	1.829	2.046	2.18	2.056	1.748	1.972	2.013	2.231	1.902	1.88	2.196	2.124	1.983	1.954	2.363	2.088	1.764	1.905	1.931
6	0.871	1.333	1.027	1.735	1.445	1.089	1.316	0.978	1.534	1.046	1.395	0.707	0.996	1.35	1.068	1.409	0.908	1.392	0.989	1.68	1.571	1.105	1.357	0.653
7	0.637	0.996	1.717	0.747	0.845	1.409	0.607	1.41	1.122	1.089	1.645	0.72	0.841	0.797	2.022	0.699	0.564	1.117	0.983	1.389	1.049	1.21	0.911	1.362
8	0.485	0.336	0.586	1.602	0.457	0.645	0.381	0.592	1.509	0.472	0.495	1.166	1.485	0.336	0.586	1.336	0.433	0.509	0.472	0.762	1.457	0.645	0.381	0.325
9	0.963	1.408	1.055	1.58	1.448	1.032	0.995	1.27	1.63	0.917	1.32	0.719	1.163	1.361	1.172	1.116	0.982	1.476	0.847	1.666	1.649	0.985	1.112	0.806
10	2.185	2.045	1.877	2.793	1.639	2.462	1.563	2.36	2.487	1.478	2.558	1.935	2.795	1.469	2.463	2.17	1.947	2.453	1.468	2.595	2.249	1.886	2.149	1.736
11	1.215	1.352	1.152	1.774	1.393	1.591	1.486	1.252	1.394	1.189	1.992	1.147	1.246	1.157	1.521	1.646	1.244	1.559	1.015	1.751	1.423	1.396	1.855	1.124
12	1.591	1.642	1.719	2.305	1.234	1.953	1.511	1.756	2.265	1.354	1.873	1.85	2.367	1.266	1.906	2.043	1.336	1.865	1.544	1.949	2.01	1.577	1.697	1.494
13	0.3	1	0.105	0.375	1	0.1	0.025	0.285	1.108	0.08	0.01	0.108	0.388	1	0.085	0.114	0.28	1.019	0.1	0.291	1.089	0.1	0.005	0.024
14	1.782	1.483	1.624	2.63	1.824	1.787	1.654	2.061	2.185	1.424	1.395	2.166	2.396	1.345	1.463	2.145	2.036	1.708	1.722	2.04	2.439	1.649	1.493	1.576
15	0.473	0.983	0.068	1.141	0.031	1.491	1.017	0.031	0.606	0.062	1.985	0.176	0.606	0.013	1.034	1.152	0.031	1.442	0.066	1.008	0.164	0.522	1.983	0.043
16	1.96	1.18	0.842	1.523	1.239	1.522	1.044	1.622	1.904	1.003	0.688	1.083	2.173	1.176	0.747	0.884	1.508	1.694	1.103	1.423	1.452	1.518	0.949	0.983

Table 8.19: Chord likelihoods for ‘Good Time’, impure vocals mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.968	1.187	1.145	0.786	1.26	1.468	0.269	0.475	0.961	1.618	0.698	0.303	0.648	1.308	1.535	0.46	0.947	1.159	1.106	0.633	0.941	1.59	0.659	0.15
2	1.918	0.091	0.368	0.837	0.677	1.024	0.144	1.632	1.037	0.225	0.32	0.614	1.818	0.214	0.323	0.092	1.458	1.013	0.147	1.111	0.578	1.147	0.098	0.887
3	1.181	0.612	1.047	0.486	0.83	1.804	0.621	0.404	0.883	1.687	0.5	0.304	0.823	0.868	1.179	0.377	0.77	0.985	1.3	0.478	0.472	2.06	0.753	0.295
4	2.072	1.38	1.819	1.336	0.926	2.464	0.673	1.794	1.583	1.266	2.23	0.473	1.884	0.976	2.522	0.771	1.227	2.175	0.966	2.091	0.737	2.06	1.377	1.229
5	1.059	0.24	0.268	0.28	1.344	0.484	0.088	0.353	0.268	1.232	0.324	0.492	0.243	1.036	0.47	0.271	1.318	0.288	0.233	0.131	0.527	1.28	0.29	0.343
6	0.538	1.2	1.553	0.823	1.363	0.808	0.929	1.694	1.126	1.47	1.364	1.343	0.798	1.364	1.647	0.871	1.035	0.702	1.212	1.222	1.623	0.972	1.023	1.741
7	0.497	1	0.218	0.004	0.478	1.024	0.218	0.005	0.028	0.473	1	0.227	0.028	0.473	1	0.223	0.478	1.024	0.218	0	0.009	0.497	1	0.223
8	1.754	1.033	1.22	0.519	1.467	1.657	0.816	1.786	1.228	1.354	1.752	0.909	1.23	1.24	1.651	0.481	1.469	1.544	0.715	1.359	0.944	1.865	1.247	1.748
9	1.777	1.518	0.622	1.296	0.926	1.453	0.623	1.261	1.214	1.129	0.799	0.375	1.685	1.25	1.097	0.367	1.396	1.574	0.921	1.253	0.834	1.185	1.099	0.332

10	1.022	0.363	1.255	0.614	0.259	0.665	0.361	1.408	0.694	0.71	1.032	0.316	1.049	0.537	1.231	0.289	0.614	0.493	0.56	1.381	0.287	0.839	0.337	1.083
11	0.991	0.42	0.913	0.162	1.19	1.437	0.564	1.196	1.242	0.693	0.269	1.365	1.071	0.419	0.656	0.428	1.019	1.163	0.951	0.259	1.269	1.436	0.307	1.462
12	1.858	0.22	0.024	0.186	0.741	1.22	0.08	0.14	1.001	0.773	0.262	0.01	1.127	0.752	0.224	0.059	0.868	1.2	0.041	0.19	0.01	1.752	0.279	0.014

Table 8.20: Chord likelihoods for ‘Tread The Water’, impure vocals mixes

8.2.6 ‘Pure Bass’ Case

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.157	0.247	1	0.087	0.366	1.013	0.112	0	0.209	1.183	0.099	0	0	0.392	1.013	0.087	0.157	0.221	1.026	0.087	0.209	1.157	0.125	0
2	1	0	0	0	1.083	0	0	0.083	0	1	0	0.083	0	1	0	0	1.083	0	0	0	0.083	1	0	0.083
3	0.248	0.003	0.047	0.276	1	0.248	0.291	1	0.284	0.003	0.24	1.083	0.284	0.003	0	0.323	1	0.248	0.05	0.24	1.036	0.248	0.244	1.047
4	0	1.11	0	0	0.11	0	1	0	0.11	1	0	0	0	1.11	0	0	0	0.11	1	0	0.11	0	1	0
5	0	1.002	0	0	0.002	0	1	0	0.002	1	0	0	0	1.002	0	0	0	0.002	1	0	0.002	0	1	0
6	1	0	0	0	1.017	0	0	0.017	0	1	0	0.017	0	1	0	0	1.017	0	0	0	0.017	1	0	0.017
7	0.112	0.003	1	0	0.112	1	0.003	0	0	1.115	0	0	0	0.115	1	0	0.112	0	1.003	0	0	1.112	0.003	0
8	0	0	0.055	0.022	1	0	0.055	1	0.022	0	0	1.077	0.022	0	0	0.077	1	0	0.055	0	1.022	0	0	1.055
9	0	0.238	1	0	0	0	1.238	0	0	0.238	0	1	0	0.238	0	1	0	0	1.238	0	0	0	0.238	1
10	0	1.055	0	0	0.055	0	1	0	0.055	1	0	0	0	1.055	0	0	0	0.055	1	0	0.055	0	1	0
11	0	0	0.084	0.05	1	0	0.084	1	0.05	0	0	1.134	0.05	0	0	0.134	1	0	0.084	0	1.05	0	0	1.084
12	0	1.094	0	0	0.094	0	1	0	0.094	1	0	0	0	1.094	0	0	0	0.094	1	0	0.094	0	1	0
13	0.117	0.02	1	0	0.117	1	0.02	0	0	1.137	0	0	0	0.137	1	0	0.117	0	1.02	0	0	1.117	0.02	0
14	1	0	0	0	1.12	0	0	0.12	0	1	0	0.12	0	1	0	0	1.12	0	0	0	0.12	1	0	0.12
15	0	0	0.062	0.045	1	0	0.062	1	0.045	0	0	1.107	0.045	0	0	0.107	1	0	0.062	0	1.045	0	0	1.062
16	0	1.099	0	0	0.099	0	1	0	0.099	1	0	0	0	1.099	0	0	0	0.099	1	0	0.099	0	1	0
17	0	0	0.052	0.037	1	0	0.052	1	0.037	0	0	1.089	0.037	0	0	0.089	1	0	0.052	0	1.037	0	0	1.052
18	0.117	0.009	1	0.05	0.123	1	0.019	0	0.039	1.12	0.016	0.033	0.033	0.126	1	0.05	0.117	0.006	1.003	0.016	0.039	1.117	0.019	0

Table 8.21: Chord likelihoods for ‘Chelsea’, pure bass mixes

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Cm	C#m	Dm	D#m	Em	Fm	F#m	Gm	G#m	Am	A#m	Bm
1	0.2	1.087	0.069	0	0.077	0.21	1	0.069	0.277	1	0.079	0	0.2	1.077	0.079	0	0	0.287	1	0.069	0.077	0.2	1.01	0.069

2	0.036	1	0.077	0.082	1	0.053	0.027	0.033	1.052	0.05	0	0.076	0.085	1	0.05	0.076	0.033	1.003	0.077	0.033	1.049	0.053	0	0.027
3	0.051	0	0.066	1.071	0.043	0	0.064	0.073	1	0.043	0.13	1	1.007	0.043	0.066	1.064	0.051	0	0	0.137	1	0.043	0.064	0.066
4	0.118	1	0.04	0	0.061	1.063	0.034	0	0.057	0.068	1	0.034	0.057	0.061	1.006	0.034	0.061	1.057	0.04	0	0	0.124	1	0.034
5	0.181	1.089	0.056	0	0.078	0.192	1	0.056	0.26	1	0.066	0	0.181	1.078	0.066	0	0	0.27	1	0.056	0.078	0.181	1.011	0.056
6	0.045	1	0.059	0.092	1	0.036	0.026	0.042	1.053	0.033	0	0.075	0.095	1	0.033	0.075	0.042	1.003	0.059	0.042	1.05	0.036	0	0.026
7	0.045	0	0.056	1.072	0.038	0	0.065	0.064	1	0.038	0.121	1	1.007	0.038	0.056	1.065	0.045	0	0	0.128	1	0.038	0.065	0.056
8	0.107	1	0.042	0	0.049	1.065	0.035	0	0.058	0.056	1	0.035	0.058	0.049	1.007	0.035	0.049	1.058	0.042	0	0	0.114	1	0.035
9	0.196	1.089	0.058	0	0.079	0.206	1	0.058	0.274	1	0.068	0	0.196	1.079	0.068	0	0	0.285	1	0.058	0.079	0.196	1.011	0.058
10	0.04	1	0.064	0.086	1	0.039	0.027	0.037	1.051	0.037	0	0.076	0.088	1	0.037	0.076	0.037	1.003	0.064	0.037	1.049	0.039	0	0.027
11	0.018	0	0.045	1.063	0.012	0	0.057	0.051	1	0.012	0.101	1	1.006	0.012	0.045	1.057	0.018	0	0	0.107	1	0.012	0.057	0.045
12	0.165	1	0.069	0	0.101	1.072	0.06	0	0.063	0.11	1	0.06	0.063	0.101	1.009	0.06	0.101	1.063	0.069	0	0	0.173	1	0.06
13	0.441	1.101	0.264	0.014	0.17	0.449	1	0.341	0.548	1	0.272	0.09	0.455	1.093	0.272	0.014	0.077	0.542	1	0.264	0.183	0.441	1.008	0.341
14	0.189	0.053	0.29	1.043	0.187	0.003	0.091	0.293	1	0.238	0.334	1	1.002	0.238	0.294	1.041	0.189	0.003	0.05	0.333	1	0.187	0.094	0.29
15	0.168	1.052	0.094	0	0.049	0.172	1	0.094	0.217	1	0.097	0	0.168	1.049	0.097	0	0	0.221	1	0.094	0.049	0.168	1.003	0.094
16	0.022	1.042	0.025	0	0.04	0.025	1	0.025	0.062	1	0.027	0	0.022	1.04	0.027	0	0	0.065	1	0.025	0.04	0.022	1.002	0.025

Table 8.22: Chord likelihoods for 'Natalie Grey', pure bass mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.294	1.071	0.201	0.388	1.004	0.161	0.111	0.291	1.097	0.094	0.071	0.208	0.388	1.004	0.161	0.208	0.294	1.071	0.201	0.291	1.097	0.094	0.071	0.111
2	0.223	0.074	0.244	1.18	0.223	0.025	0.23	0.244	1	0.273	0.448	1	1	0.273	0.268	1.18	0.223	0.025	0.05	0.423	1	0.223	0.254	0.244
3	0.084	0.122	0.142	1	0.159	0.256	1	0.249	0.085	0.051	1.212	0.158	0.084	0.001	0.263	1	0.158	0.206	0.051	1.091	0.159	0.135	1.121	0.249
4	2.094	0.634	0.233	1.092	0.853	1.51	0.293	0.951	1.507	0.601	0.587	0.922	2.051	0.638	0.562	0.66	1.397	1.547	0.268	0.689	0.81	1.514	0.623	0.52
5	0.014	0.299	0.968	1	0.347	0.299	1.968	0.333	0	0.014	1.299	1.301	0	0.014	0.299	1.968	0.347	0.299	0.968	1	0.333	0.014	1.299	1.301
6	1	0.104	0	1	0.104	0	0	1	0.104	0	0	0	1	0.104	0	0	1	0.104	0	1	0.104	0	0	0
7	0	0.448	1.205	1	0	1.359	1.295	0	0	0.911	1.448	0.295	0	0	1.359	1.295	0	0.448	1.205	1	0	0.911	1.448	0.295
8	0	1.106	0	0	0.106	1	0	0	0.106	0	1	0	0	0.106	1	0	0	1.106	0	0	0.106	0	1	0
9	0.242	1.038	0.219	0.284	1	0.174	0.082	0.242	1.042	0.136	0.038	0.125	0.284	1	0.174	0.125	0.242	1.038	0.219	0.242	1.042	0.136	0.038	0.082
10	0.284	0.141	0.27	1.182	0.278	0.088	0.247	0.258	1	0.331	0.515	1.019	1.006	0.331	0.339	1.194	0.284	0.088	0.072	0.433	1	0.278	0.316	0.27
11	0.173	0.152	0.193	1.022	0.232	0.374	1.022	0.353	0.151	0.116	1.228	0.232	0.173	0.022	0.322	1	0.254	0.281	0.116	1.121	0.232	0.245	1.152	0.331

12	1.818	1.182	0.622	0.795	0.777	1.84	0.75	0.98	1.337	0.67	1.08	0.885	1.687	0.74	1.032	0.685	1.126	1.91	0.702	0.78	0.645	1.398	1.16	0.87
13	0.291	1.113	0.37	0.724	1.005	0.358	0.497	0.286	1.066	0.25	0.485	0.191	0.352	1.005	0.358	0.563	0.291	1.113	0.37	0.658	1.066	0.25	0.485	0.125
14	0.16	0.05	0.214	1.024	0.108	0.116	0.071	0.283	1.116	0.091	0.236	1.067	1.118	0.091	0.214	1.022	0.11	0.116	0.05	0.238	1.067	0.158	0.071	0.281
15	0.001	0.002	0.088	1	0.151	0.088	1	0.149	0.002	0.088	1	0.149	0.001	0.002	0.088	1	0.149	0.002	0.088	1	0.151	0.088	1	0.149
16	1.62	0.506	0.763	1.634	0.817	0.97	0.667	1.597	1.29	0.402	0.574	0.957	1.897	0.476	0.551	0.89	1.424	1.044	0.644	1.529	1.095	0.94	0.455	0.853

Table 8.23: Chord likelihoods for ‘Good Time’, pure bass mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.075	1	0	0	0	1.075	0	0	0.075	0	1	0	0.075	0	1	0	0	1.075	0	0	0	0.075	1	0
2	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
3	0.036	0	1	0	0.036	1	0	0	0	1.036	0	0	0	0.036	1	0	0.036	0	1	0	0	1.036	0	0
4	1.571	1.316	0.843	0.296	0.928	1.903	0.427	1.061	1.056	0.958	1.715	0.564	1.124	0.937	1.734	0.311	0.996	1.882	0.446	0.809	0.48	1.524	1.318	1.077
5	0.142	0	1	0	0.142	1	0	0	0	1.142	0	0	0	0.142	1	0	0.142	0	1	0	0	1.142	0	0
6	0	0	0.122	0.007	1	0	0.122	1	0.007	0	0	1.13	0.007	0	0	0.13	1	0	0.122	0	1.007	0	0	1.122
7	0.123	1	0	0	0.095	1.096	0	0.068	0.096	0.027	1	0.068	0.096	0.027	1	0	0.095	1.096	0	0	0.068	0.123	1	0.068
8	1.052	0	0	0.052	0	1	0	0.052	1	0	0	0	1.052	0	0	0	0.052	1	0	0.052	0	1	0	0
9	0.055	1	0	0	0	1.055	0	0	0.055	0	1	0	0.055	0	1	0	0	1.055	0	0	0	0.055	1	0
10	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
11	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0
12	1.778	1.094	0.827	0.298	1.092	1.636	0.302	0.846	0.921	1.17	1.607	0.448	0.997	1.165	1.615	0.359	1.168	1.631	0.31	0.757	0.311	1.707	1.09	0.907

Table 8.24: Chord likelihoods for ‘Tread The Water’, pure bass mixes

8.2.7 ‘Impure Bass’ Case

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.164	0.22	1	0.095	0.388	1.008	0.109	0.025	0.199	1.177	0.104	0.025	0	0.376	1.008	0.095	0.189	0.207	1.014	0.095	0.224	1.164	0.117	0.025
2	1	0	0	0.008	1.077	0	0	0.077	0.008	1	0	0.085	0.008	1	0	0.008	1.077	0	0	0	0.085	1	0	0.077
3	0.143	0	0.065	0.275	1	0.143	0.261	1.012	0.21	0	0.22	1.12	0.21	0	0.011	0.328	1	0.143	0.053	0.22	1.067	0.143	0.208	1.065
4	0.034	1.112	0.008	0	0.112	0.034	1	0.008	0.146	1	0.008	0	0.034	1.112	0.008	0	0	0.146	1	0.008	0.112	0.034	1	0.008
5	0	1.013	0	0	0.013	0	1	0	0.013	1	0	0	0	1.013	0	0	0	0.013	1	0	0.013	0	1	0

6	1	0	0	0	1.018	0	0	0.018	0	1	0	0.018	0	1	0	0	1.018	0	0	0	0.018	1	0	0.018
7	0.124	0.004	1	0	0.124	1	0.004	0	0	1.128	0	0	0	0.128	1	0	0.124	0	1.004	0	0	1.124	0.004	0
8	0.001	0	0.052	0.022	1.001	0	0.052	1	0.022	0.001	0	1.074	0.022	0.001	0	0.074	1.001	0	0.052	0	1.022	0.001	0	1.052
9	0	0.217	1.003	0	0	0	1.217	0.003	0	0.217	0.003	1	0	0.217	0.003	1	0	0	1.217	0.003	0	0	0.217	1.003
10	0	1.061	0	0	0.061	0	1	0	0.061	1	0	0	0	1.061	0	0	0	0.061	1	0	0.061	0	1	0
11	0	0	0.079	0.042	1	0	0.079	1	0.042	0	0	1.122	0.042	0	0	0.122	1	0	0.079	0	1.042	0	0	1.08
12	0	1.085	0	0.045	0.108	0	1	0.023	0.131	1	0	0.068	0.045	1.086	0	0.045	0.023	0.085	1	0	0.153	0	1	0.023
13	0.093	0.023	1	0	0.093	1	0.023	0	0	1.115	0	0	0	0.115	1	0	0.093	0	1.023	0	0	1.093	0.023	0
14	1	0	0	0	1.109	0	0	0.109	0	1	0	0.109	0	1	0	0	1.109	0	0	0	0.109	1	0	0.109
15	0	0	0.097	0.044	1	0	0.097	1	0.044	0	0	1.141	0.044	0	0	0.141	1	0	0.097	0	1.044	0	0	1.097
16	0	1.107	0	0	0.107	0	1	0	0.107	1	0	0	0	1.107	0	0	0	0.107	1	0	0.107	0	1	0
17	0	0	0.053	0.035	1	0	0.053	1	0.035	0	0	1.088	0.035	0	0	0.088	1	0	0.053	0	1.035	0	0	1.053
18	0.149	0.011	1.008	0.105	0.149	1.001	0.026	0.008	0.089	1.159	0.025	0.089	0.089	0.159	1.009	0.105	0.149	0.001	1.01	0.024	0.089	1.149	0.026	0.008

Table 8.25: Chord likelihoods for 'Chelsea', impure bass mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.26	1.148	0.157	0.047	0.139	0.238	1	0.188	0.352	1.016	0.15	0	0.26	1.139	0.166	0	0.047	0.361	1.016	0.188	0.139	0.229	1.009	0.141
2	0.176	1	0.218	0.279	1	0.122	0.132	0.175	1.069	0.122	0.036	0.164	0.244	1	0.122	0.2	0.175	1	0.218	0.211	1.068	0.122	0.036	0.096
3	0.075	0.049	0.043	1.086	0.069	0	0.092	0.068	1.019	0.081	0.105	1	1.024	0.099	0.043	1.062	0.075	0.019	0.03	0.129	1.019	0.051	0.092	0.043
4	0.22	1	0.056	0.082	0.124	1.069	0.048	0.035	0.108	0.131	1	0.095	0.143	0.124	1.008	0.095	0.158	1.061	0.056	0.035	0.047	0.193	1	0.048
5	0.196	1.089	0.039	0	0.073	0.215	1	0.034	0.269	1.005	0.049	0	0.196	1.073	0.054	0	0	0.284	1.005	0.034	0.073	0.2	1.015	0.034
6	0.007	1	0.011	0.043	1	0.017	0	0	1.049	0.011	0	0.043	0.049	1	0.011	0.043	0	1.007	0.011	0	1.043	0.017	0	0
7	0.091	0	0.133	1.074	0.075	0	0.058	0.149	1	0.075	0.191	1	1.016	0.075	0.133	1.058	0.091	0	0	0.207	1	0.075	0.058	0.133
8	0.099	1	0.028	0	0.031	1.074	0.021	0	0.067	0.038	1	0.021	0.067	0.031	1.006	0.021	0.031	1.067	0.028	0	0	0.105	1	0.021
9	0.187	1.083	0.058	0	0.07	0.2	1	0.058	0.257	1	0.071	0	0.187	1.07	0.071	0	0	0.27	1	0.058	0.07	0.187	1.013	0.058
10	0.091	1	0.195	0.162	1	0.121	0.098	0.085	1.063	0.116	0.019	0.137	0.148	1	0.116	0.156	0.085	1.005	0.195	0.104	1.057	0.121	0.019	0.079
11	0.02	0	0.049	1.098	0	0	0.079	0.069	1	0	0.128	1	1.02	0	0.049	1.079	0.02	0	0	0.147	1	0	0.079	0.049
12	0.355	1.18	0.057	0.232	0.151	1.086	0.233	0.136	0.166	0.332	1.02	0.128	0.293	0.326	1.015	0.143	0.279	1.08	0.228	0.152	0.089	0.233	1.191	0.047
13	0.67	1.497	0.689	0.411	0.521	0.752	1.262	0.663	0.783	1.277	0.407	0.359	0.703	1.385	0.611	0.296	0.441	0.861	1.466	0.6	0.554	0.641	1.184	0.548

14	0.337	0.367	0.335	1.212	0.304	0.164	0.355	0.325	1.184	0.467	0.362	1.034	1.159	0.516	0.33	1.143	0.28	0.213	0.323	0.435	1.127	0.313	0.35	0.257
15	0.149	1.046	0.072	0.029	0.044	0.122	1	0.101	0.164	1	0.074	0	0.149	1.044	0.074	0	0.029	0.166	1	0.101	0.044	0.121	1.002	0.072
16	0.159	1.13	0	0.006	0.326	0.001	1	0.038	0.136	1.158	0	0.044	0.006	1.287	0	0.006	0.197	0.13	1	0	0.174	0.159	1	0.038

Table 8.26: Chord likelihoods for ‘Natalie Grey’, impure bass mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.323	1.033	0.332	0.53	1.093	0.294	0.155	0.409	1.256	0.257	0.192	0.297	0.499	1.038	0.31	0.315	0.335	1.075	0.274	0.428	1.269	0.299	0.134	0.195
2	0.402	0.382	0.676	1.559	0.517	0.419	0.618	0.629	1.351	0.462	0.731	1.301	1.287	0.446	0.594	1.522	0.453	0.403	0.481	0.85	1.402	0.482	0.535	0.592
3	0.568	0.572	0.743	1.363	0.798	0.861	1.183	0.787	0.767	0.668	1.466	0.667	0.636	0.539	0.864	1.294	0.667	0.732	0.581	1.414	0.867	0.828	1.304	0.719
4	2.412	1.425	0.982	1.995	1.627	1.973	1.048	1.727	2.213	1.362	1.386	1.537	2.555	1.442	1.333	1.42	1.969	2.053	0.994	1.611	1.77	1.99	1.398	1.152
5	2.301	2.362	2.171	2.506	2.434	2.228	2.113	2.292	2.532	2.176	2.197	2.219	2.415	2.364	2.215	2.293	2.317	2.416	2.131	2.366	2.548	2.23	2.156	2.079
6	1.841	2.054	1.865	2.39	2.175	1.96	1.949	1.89	2.254	1.883	2.072	1.802	1.981	2.045	1.94	2.15	1.902	2.123	1.817	2.237	2.315	1.951	2.024	1.649
7	2.3	2.428	2.572	2.575	2.411	2.571	2.316	2.565	2.481	2.388	2.8	2.173	2.363	2.331	2.77	2.392	2.293	2.515	2.286	2.784	2.473	2.474	2.514	2.382
8	2.481	2.499	2.329	2.806	2.422	2.675	2.288	2.474	2.854	2.347	2.532	2.433	2.803	2.351	2.534	2.546	2.371	2.68	2.29	2.587	2.745	2.528	2.492	2.214
9	0.325	1.02	0.431	0.549	1.07	0.373	0.179	0.415	1.223	0.333	0.202	0.27	0.488	1.02	0.394	0.322	0.335	1.06	0.37	0.467	1.233	0.373	0.141	0.188
10	0.508	0.436	0.72	1.541	0.54	0.512	0.578	0.694	1.381	0.526	0.801	1.271	1.347	0.496	0.705	1.467	0.506	0.482	0.482	0.89	1.379	0.572	0.562	0.621
11	0.608	0.773	0.964	1.416	0.863	1.008	1.337	1.12	0.823	0.737	1.708	0.829	0.738	0.597	1.123	1.337	0.779	0.868	0.752	1.628	0.993	0.833	1.497	1.04
12	2.296	1.973	1.746	1.801	1.467	2.511	1.572	1.791	2.062	1.491	1.934	1.506	2.309	1.501	2.035	1.58	1.713	2.521	1.674	1.865	1.48	2.039	1.862	1.571
13	0.274	1.103	0.378	0.684	1.002	0.363	0.468	0.273	1.06	0.263	0.454	0.177	0.332	1.002	0.363	0.528	0.275	1.103	0.378	0.624	1.06	0.263	0.454	0.118
14	0.183	0.074	0.223	1.046	0.116	0.143	0.115	0.308	1.143	0.109	0.264	1.081	1.147	0.109	0.223	1.042	0.12	0.143	0.074	0.268	1.081	0.178	0.115	0.304
15	0.01	0.02	0.096	1	0.168	0.124	1	0.165	0.012	0.096	1.018	0.165	0.01	0.002	0.114	1	0.165	0.03	0.096	1	0.168	0.106	1.018	0.165
16	1.681	0.509	0.748	1.635	0.828	1.027	0.675	1.591	1.342	0.407	0.565	0.963	1.95	0.48	0.537	0.892	1.436	1.101	0.646	1.519	1.098	0.999	0.464	0.848

Table 8.27: Chord likelihoods for ‘Good Time’, impure bass mixes

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**	**Cm**	**C#m**	**Dm**	**D#m**	**Em**	**Fm**	**F#m**	**Gm**	**G#m**	**Am**	**A#m**	**Bm**
1	0.035	1	0	0	0	1.035	0	0	0.035	0	1	0	0.035	0	1	0	0	1.035	0	0	0	0.035	1	0
2	1	0	0	1	0.006	0	0	1.006	0	0	0	0.006	1	0	0	0	1.006	0	0	1	0.006	0	0	0.006
3	0.023	0	1	0	0.023	1	0	0	0	1.023	0	0	0	0.023	1	0	0.023	0	1	0	0	1.023	0	0
4	1.562	1.316	0.844	0.304	0.894	1.927	0.421	1.062	1.036	0.961	1.715	0.523	1.122	0.913	1.751	0.286	0.98	1.88	0.457	0.825	0.454	1.524	1.328	1.044
5	0.124	0	1	0	0.124	1	0	0	0	1.124	0	0	0	0.124	1	0	0.124	0	1	0	0	1.124	0	0

6	0	0	0.127	0.001	1	0	0.126	1.001	0.001	0	0.001	1.127	0.001	0	0.001	0.127	1	0	0.126	0.001	1.001	0	0	1.127
7	0.13	1	0	0	0.107	1.096	0	0.072	0.096	0.034	1	0.072	0.096	0.034	1	0	0.107	1.096	0	0	0.072	0.13	1	0.072
8	1.061	0	0	0.046	0.016	1	0	0.046	1	0.016	0	0	1.046	0.016	0	0	0.061	1	0	0.046	0	1.016	0	0
9	0.074	1	0	0	0	1.074	0	0	0.074	0	1	0	0.074	0	1	0	0	1.074	0	0	0	0.074	1	0
10	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
11	0	0.002	1	0	0	1	0.002	0	0	1.002	0	0	0	0.002	1	0	0	0	1.002	0	0	1	0.002	0
12	1.856	1.069	0.845	0.312	1.088	1.713	0.313	0.835	0.967	1.204	1.562	0.448	1.065	1.159	1.605	0.364	1.186	1.668	0.356	0.751	0.297	1.803	1.073	0.887

Table 8.28: Chord likelihoods for 'Tread The Water', impure bass mixes