

Computational Tonality Estimation: Signal Processing and Hidden Markov Models

Thesis submitted in partial fulfilment
of the requirements of the University of London
for the Degree of Doctor of Philosophy

Katy C. Noland
Centre for Digital Music
School of Electronic Engineering and Computer Science
Queen Mary, University of London

March 2009

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. I acknowledge the helpful guidance and support of my supervisor, Professor Mark Sandler.

Abstract

This thesis investigates computational musical tonality estimation from an audio signal. We present a hidden Markov model (HMM) in which relationships between chords and keys are expressed as probabilities of emitting observable chords from a hidden key sequence. The model is tested first using symbolic chord annotations as observations, and gives excellent global key recognition rates on a set of Beatles songs.

The initial model is extended for audio input by using an existing chord recognition algorithm, which allows it to be tested on a much larger database. We show that a simple model of the upper partials in the signal improves percentage scores. We also present a variant of the HMM which has a continuous observation probability density, but show that the discrete version gives better performance.

Then follows a detailed analysis of the effects on key estimation and computation time of changing the low level signal processing parameters. We find that much of the high frequency information can be omitted without loss of accuracy, and significant computational savings can be made by applying a threshold to the transform kernels. Results show that there is no single ideal set of parameters for all music, but that tuning the parameters can make a difference to accuracy.

We discuss methods of evaluating more complex tonal changes than a single global key, and compare a metric that measures similarity to a ground truth to metrics that are rooted in music retrieval. We show that the two measures give different results, and so recommend that the choice of evaluation metric is determined by the intended application.

Finally we draw together our conclusions and use them to suggest areas for continuation of this research, in the areas of tonality model development, feature extraction, evaluation methodology, and applications of computational tonality estimation.

*Dedicated to the memory of
Jo Causier*

Acknowledgements

This research would not have been possible without the help and support I received from the people around me, and I would like to thank them all for their useful insights, searching questions and welcome distractions.

I would particularly like to thank my supervisor, Mark Sandler, for his guidance and encouragement, and my secondary supervisors, Juan Bello, Simon Dixon and Mark Plumbley, who have given me useful and objective advice that has steered my research.

Thankyou to the past and present members of the Centre for Digital Music at Queen Mary who have together created a vibrant and friendly atmosphere for research. I would especially like to thank Matthias Mauch, for his support, and for his extensive and helpful comments on various drafts of this thesis and on my work in general. I would also like to thank Chris Harte for the numerous discussions over coffee, Samer Abdallah, Christophe Rhodes and Mark Levy for their expertise that has led to joint publications, and Dawn Black for sharing the journey.

My mother and father, Jean and Jim Noland, have always supported my love of learning and of music, and I would like to thank them for this and for everything else they have done for me over the years.

Thankyou also to Amélie Anglade, Tom Atkinson, Janet Baddeley, Paul Baddeley, Poppy Baddeley, Kate Bergel, Caroline Black, Paul Brossier, Chris Cannam, Sarah Carrington, Nico Chétry, Adam Clifford, Matthew Davies, Chris Duxbury, György Fazekas, Peyman Heydarian, Claire Hopkins, Ho Huen, Kurt Jacobson, Maria Jafari, Panos Kudumakis, Chris Landone, Rob Macrae, Usman Naeem, Andrew Nesbit, Eileen Noland, John Oag, Kat Osborn, Elias Pampalk, Ben Parker, Enrique Perez Gonzalez, Yves Raimond, Josh Reiss, Andrew Robertson, Adam Stark, Becky Stewart, Dan Stowell, Chris Sutton, Dan Tidhar, Graham Tudball, Emmanuel Vincent, Beiming Wang, Jennie Want, Steve Welburn, Vindya Wijeratne, Wen Xue and Ruohua Zhou.

I would also like to acknowledge the funding I received from the Engineering and Physical Sciences Research Council (EPSRC).

Contents

Contents	6
List of Figures	9
List of Tables	13
1 Introduction	15
1.1 Definition of Computational Tonality Estimation	15
1.2 Motivation: Applications of Computational Tonality Estimation	16
1.3 Contributions of the Thesis	17
1.4 Related Publications by the Author	18
1.5 Thesis Outline	18
2 Computational Tonality Estimation	20
2.1 General Structure of Computational Tonality Estimation Algorithms	20
2.2 Type of Key Model	21
2.2.1 Perceptual Models	22
2.2.2 Tone Profiles	22
2.2.3 Geometric Models	29
2.2.4 Preference Rules	31
2.2.5 Machine Learning	32
2.3 Extracting the Musical Features	36
2.3.1 Time Domain Signal Processing	37
2.3.2 Discrete Fourier Transform	39
2.3.3 Constant-Q Transform	40
2.3.4 Chroma Vectors and Chromagrams	42
2.3.5 Alternatives to the Constant-Q Transform	43
2.3.6 Disadvantages of the Chromagram	44
2.3.7 Adaptations of Chroma Features for Symbolic Tone Profiles	45
2.4 Classification Methods	47

2.5	The Final Tonality Estimate	48
2.5.1	Location in Time of the Main Tonality of a Piece	49
2.5.2	Hierarchical aspects of Harmony	50
2.5.3	Harmonic Progression Through Time	51
2.6	Methods of Evaluation	53
2.7	Summary	54
3	HMM for Tonality Estimation from Chord Symbols	57
3.1	Hidden Markov Models	58
3.2	Model of Tonal Space	58
3.2.1	Initialisation	59
3.2.2	Adaptation of Model Probabilities	65
3.2.3	Decoding	67
3.3	Evaluation Technique	67
3.3.1	Parameters Tested	69
3.3.2	Model Structure Variations	74
3.3.3	Confusion Matrix for the Best Case	80
3.4	The Model as a Segmentation Algorithm	82
3.5	Summary	84
4	Extension of the Model for Audio Input	87
4.1	Audio Test Collections for Global Key Recognition	87
4.2	Addition of a Chord Recognition Step	88
4.2.1	Modelling Upper Partials in the Chord Templates	89
4.3	Performance of the Model with Chord Symbols Extracted from Audio	91
4.4	Model with Continuous Observation Probability Density	97
4.5	Summary	105
5	DSP Parameters and Preprocessing	107
5.1	Investigation into the Effects of Varying the Low-Level DSP Parameters	108
5.1.1	Downsampling	109
5.1.2	Minimum Frequency	112
5.1.3	Hop Size	116
5.1.4	Sparse Kernel Threshold	118
5.2	Computation Time Comparison	120
5.3	Investigation into the Effects of Using Two Preprocessing Algorithms	123
5.3.1	Beat Tracking	124

5.3.2 Transient Removal	125
5.4 Comparison with a Tone Profile Correlation Method of Main Key Estimation	128
5.5 Summary	130
6 Evaluation Methods for Detailed Tonal Analyses	133
6.1 Requirements of an Evaluation Metric for Detailed Tonal Analyses	133
6.2 Comparison to Human Perception of Tonal Changes	135
6.3 Subjective testing	135
6.4 Application-based testing	136
6.5 Experiment Design	137
6.5.1 Tonal Analysis Methods Used in the Evaluation Experiments	137
6.5.2 Ground Truth Comparison	142
6.5.3 Retrieval Experiments	144
6.6 Results of the Comparison of Evaluation Measures Experiment	147
6.7 Summary	150
7 Conclusions and Future Directions	151
7.1 Conclusions	151
7.2 Future Directions	153
A Important Musical Concepts	155
A.1 Equal Temperament	155
A.2 Major and Minor Scales	155
A.3 Key	156
A.4 Chords	156
List of Abbreviations	158
References	159

List of Figures

2.1	Structure of a generic tonality estimation algorithm.	21
2.2	Shepard's helical model of pitch.	23
2.3	Example tone profiles.	25
2.4	Example tone profiles including an exponential decay model of the first 3 upper partials.	28
2.5	Chew's Spiral Array.	30
2.6	Illustration of downsampling factor, window size and hop size.	38
2.7	Rectangular and Hamming windows in the time and frequency domains.	39
2.8	Illustration of the division of the time-frequency plane for two different Fourier transforms and the constant-Q transform.	41
3.1	Simplified diagram of the tonality model.	59
3.2	Krumhansl's probe tone ratings for C major and C minor key contexts.	63
3.3	Krumhansl's correlations between key profiles.	63
3.4	Krumhansl's harmonic hierarchy ratings for major, minor and diminished chords.	63
3.5	Visualisation of the initial transition matrix.	64
3.6	Visualisation of the initial emission matrix.	65
3.7	Main key estimation scores for different self-transition probabilities.	70
3.8	Main key estimation scores for different initialisation parameters.	71
3.9	Main key estimation scores when different sets of parameters are adapted.	72
3.10	Main key estimation scores for different chord sampling intervals.	73
3.11	Main key estimation scores for different self-transition probabilities for the chord transitions model, with chords sampled once per annotation.	76
3.12	Main key estimation scores for different initialisation parameters for the chord transitions model, with chords sampled once per annotation.	76
3.13	Main key estimation scores when different sets of parameters are adapted for the chord transitions model, with chords sampled once per annotation.	77

3.14 Main key estimation scores for different chord sampling intervals for the chord transition model.	77
3.15 Main key estimation scores for different self-transition probabilities for the chord transitions model with 100 ms sampling interval.	78
3.16 Main key estimation scores for different initialisation parameters for the chord transitions model with 100 ms sampling interval.	78
3.17 Main key estimation scores when different sets of parameters are adapted for the chord transitions model with 100 ms sampling interval.	79
3.18 Main key estimation scores with and without augmented and diminished triads. . .	80
3.19 Confusion matrix for the best-performing model.	81
3.20 Model outputs for the Beatles' <i>I'll Cry Instead</i>	82
3.21 Model outputs for the Beatles' <i>I'm Happy Just to Dance With You</i>	83
4.1 Templates for chords on C, modelling 7 upper partials, with a decay factor, s , of 0.6. .	92
4.2 Main key estimation scores when using a decay factor, s , of between 0 and 0.8 in the chord templates, with 3 upper partials modelled.	94
4.3 Main key estimation scores when using a decay factor, s , of between 0 and 0.8 in the chord templates, with 7 upper partials modelled.	95
4.4 Main key estimation scores when modelling 0, 3 and 7 upper partials in the chord templates with a decay factor, s , of 0.6.	96
4.5 Distribution of errors with respect to the correct key for each test collection, when the simple chord templates that do not model any upper partials are used.	98
4.6 Distribution of errors with respect to the correct key for each test collection, when 3 upper partials are modelled in the chord templates with a decay factor, s , of 0.6. .	99
4.7 Percentage and MIREX scores for our best performing chord templates, with $s = 0.6$ and 3 upper partials modelled.	100
4.8 Main key estimation scores for our best performing model on the whole track and on only the first 30 seconds.	100
4.9 Visualisation of the mean values of the 12-dimensional Gaussian for each chord in the continuous model.	101
4.10 Visualisations of the covariance matrices for chords on C in the continuous model. .	101
4.11 Main key estimation scores for the best performing discrete HMM and the continuous HMM.	102
4.12 Distribution of errors with respect to the correct key for each test collection, when the HMM with a continuous observation probability density function is used. . . .	103

4.13	Distribution of errors with respect to the correct key for each test collection, when the HMM with a continuous observation probability density function and adapted mixture weights to reduce dominant errors is used.	104
4.14	Main key estimation scores for the best performing discrete HMM and the continuous HMM with adapted mixture weights to reduce dominant key errors.	105
5.1	Magnitude response of the anti-aliasing filter for a downsampling factor of 2.	110
5.2	Main key estimation scores when using a downsampling factor of between 1 (no downsampling) and 64.	111
5.3	Distribution of errors with respect to the correct key for each test collection, when the audio is not downsampled.	113
5.4	Distribution of errors with respect to the correct key for each test collection, when the audio is downsampled by a factor of 16.	114
5.5	Main key estimation scores when using a minimum constant-Q frequency of between 41.2 Hz and 164.8 Hz.	115
5.6	Main key estimation scores when using a hop size of between 1/64th of a frame (0.023 s) and a whole frame (1.5 s).	119
5.7	Accurate spectral kernel and a kernel with a threshold of 0.5 applied, for an analysis frequency of 1760 Hz.	120
5.8	Main key estimation scores when using a constant-Q kernel threshold of between 0 (no thresholding) and 0.5.	121
5.9	Main key estimation scores with equal-length frames and with beat-length frames. .	126
5.10	Main key estimation scores when using the full (downsampled) audio signal and the audio after transient removal.	127
5.11	Main key estimation scores for the best-performing HMM method and two tone profile correlation methods, using profiles recommended by Gómez and profiles derived from recordings of Bach pieces.	129
6.1	Automatically-generated detailed tonal analysis of Beethoven's Symphony number 5, movement 1. The plot was generated from the output of a tone-profile correlation key estimation method, using Gómez' profiles.	134
6.2	Example automatic tonal analyses of the Bach C minor Prelude using the profile correlation method with different frame sizes.	139
6.3	Example automatic tonal analyses of the Bach C minor Prelude using the HMM method, with tone profile and flat initialisations of the transition and observation probabilities.	140

6.4	Expert ratings for the Bach C minor Prelude.	141
6.5	Example dissimilarity matrices for similar and dissimilar pieces, with alignment costs.	143
6.6	First two bars of the original Bach prelude and the two variations.	145
6.7	Evaluation scores for the profile-based tonal analysis methods.	148
6.8	Evaluation scores for the HMM-based tonal analysis methods.	149
A.1	The C major scale, and harmonic and melodic variations of the C minor scale. . .	156

List of Tables

2.1	Relative scores for different keys used in the MIREX 2005 key finding contests.	53
3.1	Krumhansl's probe tone ratings for C major and C minor key contexts.	61
3.2	Krumhansl's correlations between key profiles.	61
3.3	Krumhansl's harmonic hierarchy ratings for major, minor and diminished chords. .	62
3.4	Krumhansl's chord transition ratings for a major key context.	74
3.5	Best parameters for the symbolic model.	85
4.1	Parameters used for chromagram calculation.	89
4.2	Simple chord templates used by the chord recognition algorithm.	89
4.3	Contributions of the upper partials to the note template, for note C.	90
4.4	Calculation of the template for note C, with 7 upper partials modelled.	90
4.5	Template for chord C major, with 7 upper partials modelled.	91
4.6	Parameters that produce the best main key estimation performance.	106
5.1	Initial parameters for the key estimation model, and alternative values tested. . . .	109
5.2	Details of tracks for which key estimation is affected by excluding the lowest octave, between 41.2 and 82.4 Hz.	117
5.3	Computation times for all 6 test collections together, for extraction of chroma features from audio and main key estimation from chroma features. Five different sets of parameter values, <i>a–e</i> , were tested.	122
5.4	Breakdown of chroma feature extraction times for parameter sets <i>b</i> and <i>d</i> from table 5.3. Values are approximate, calculated from the time taken to process one piece only.	123
5.5	Parameters that produce the best main key estimation performance.	131
6.1	Tonal analysis methods used in the evaluation experiments.	137
6.2	Parameters used for chromagram calculation in the evaluation experiments. . . .	138
6.3	Tonally similar recordings used in the retrieval experiment.	146

6.4 Ranks given to the different tonal analysis techniques by the different evaluation measures	147
A.1 Names given to the degrees of the scale.	156

Chapter 1

Introduction

This thesis addresses computational tonality estimation within the context of music information retrieval. In this chapter we define and discuss our aims, and go on to describe some of the potential uses of automatic tonal analysis. We summarise the contributions to the field made by this thesis and list our related publications, then give an outline of the thesis structure.

1.1 Definition of Computational Tonality Estimation

The overriding goal of this work is to develop computer software that can take a musical recording in digital audio format as its input, analyse it, and produce an estimation of the musical tonality. Tonality, however, is a difficult concept to define. The terms *key* and *tonality* are often used synonymously, but in this thesis we do make a distinction, based on the definitions given in *The New Grove Dictionary of Music and Musicians* [Sadie, 1980].

Key “The quality of a musical composition or passage that causes it to be sensed as gravitating towards a particular note, called the key note or the tonic. One therefore speaks of a piece as being in the key of C major or minor, etc. The key of a movement commonly changes during its course through the process of modulation, returning to the home key before the end.”¹ [Sadie, 1980, vol. 10]

This rigid definition implies a single tonal centre and mode at a given time. We use *key* to refer to a single, discrete tonal centre and associated scale, with the acknowledgement that there is likely to be more than one key exerting an influence in a given piece of music.

However, this is a very simplified view of all but the most rudimentary music. A more complex, and more correct analysis will account for the possibility that a composer might not only firmly establish a single key, but also imply aspects of several keys at once without necessarily definitively confirming them. The Grove entry for *tonality* [Sadie, 1980, vol. 19] captures this broader meaning, spanning several pages with seven different definitions given. We use *tonality* to

¹References to other dictionary entries have been omitted.

refer to the abstract concept of the music's relationship to all possible keys, which is continuously shifting. This is closest to the sixth definition given in Grove:

Tonality ‘[...] While the word ‘key’ is linked with the idea of a diatonic scale in which the notes, intervals and chords are contained, a tonality reaches further than the note content of a major or minor scale, through chromaticism, passing reference to other key areas, or wholesale modulation: the decisive factor in the tonal effect is the functional association with the tonic chord (emphasized by functional theory), not the link with a scale (which is regarded as the basic determinant of key in theory of fundamental progressions). A tonality is thus an expanded key.’ [Sadie, 1980, vol. 19]

These definitions of *key* and *tonality* are only relevant to Western tonal music, so we restrict our experiments and discussion to music that fits within that category.

Although the algorithms we develop are capable of producing estimates of the changing tonality through a musical excerpt, the abstract and subjective nature of tonality means that these estimates are difficult to evaluate, so the majority of the experiments in this thesis focus on estimation of the single most important key in a musical excerpt. Chapter 6 is devoted to finding a more suitable evaluation metric.

1.2 Motivation: Applications of Computational Tonality Estimation

The primary purpose of the work described in this thesis is to create a means of automatically generating tonal analyses that are useful as metadata for the purposes of music retrieval. The key of a piece is one of the parameters that often appears in the titles of classical music, together with some information about the structure or style, such as “Sonata in F” or “Scherzo in G”, which suggests that it is considered by composers as an important means of classification.

One can imagine a retrieval system which enables searches for queries such as “find all pieces by Mozart in the key of G minor”, or “find all songs from the 1990s that include a key shift of one tone”. One might also wish to allow more complex queries on tonality, such as “find a piece that has unstable tonality”, for which more than a simple key label must be encoded in the metadata. Tonality could also be used as one parameter for more general queries such as “find pieces that sound similar to this one”, alongside other features such as instrumentation and rhythmic patterns. A description of tonal changes through a piece can be used to aid automatic music structure analysis, which itself can be used as metadata for a retrieval system.

Although the main purpose of our work is to enable music retrieval using audio features, there are other possible applications. An accurate description of the tonal content could be used as

an aid to musicologists, to give a quick overview of a piece. The use of computers also enables very fast comparisons of the tonal changes in large music collections, which could enable new connections to be made between diverse styles.

The development of tonal models could also assist with understanding human cognition of tonality. If a model is created that can accurately predict human judgements, one can hypothesise that a similar process is happening in the brain whilst listening. Some of the earliest literature on tonality estimation is a product of the psychological research community, where the purpose is to understand how the brain perceives tonality.

These subsidiary applications however do not necessarily all require the same kind of tonal analysis. We return to this topic in chapter 6.

1.3 Contributions of the Thesis

This thesis describes the development of a new method for estimating the main key of a piece, the key changes through a piece, and the relative strengths of all keys as they change through a piece, which we interpret as descriptors of the tonality changes. The method uses a hidden Markov model to represent the relationships between chords and keys, as measured in listener rating tests. We introduce the model in the symbolic domain in chapter 3, and perform a thorough investigation into its strengths and weaknesses through measuring the performance of many model variants. We extend the model for audio input data in chapter 4 and demonstrate that a simple treatment of the upper partials in the audio signal improves results. In chapter 5 our model is shown to give better main key estimation performance than the more popular tone profile correlation approach on datasets containing mixed timbres, showing that it is better able to generalise.

In chapter 5 we also highlight the importance of low level parameter choices, which are often ignored, and perform extensive testing of their effects. We show that a suitable choice of parameters can produce significant computational savings with minimal loss of accuracy. In particular the model is robust to thresholding of the frequency transform kernels, and performance is found to improve when the sampling rate is reduced from 44.1 kHz to 2.8 kHz.

We devote chapter 6 to the question of evaluation for tonality estimation algorithms, a subject which has not been previously addressed in detail. We perform an experiment to compare an evaluation metric based on comparison with a hand-annotated ground truth to a metric that measures how well a tonal analysis performs in a music retrieval task, and show that the two metrics give very different results. This means that striving for a tonal analysis that best matches human annotations will not necessarily lead to the best analysis for music retrieval.

We also contribute an extensive critical discussion of the state of the art in computational tonality estimation algorithms in chapter 2, and suggest suitable future directions for the field

in chapter 7, in the areas of model development, feature extraction, evaluation methods and applications of computational tonality estimation.

1.4 Related Publications by the Author

Much of the work described in this thesis is based on our earlier publications, which have been re-ordered and extended. We list them here.

Conference Proceedings

Katy Noland and Mark Sandler. *Key estimation using a hidden Markov model*. In Proceedings of the 7th International Conference on Music Information Retrieval, Victoria, 2006.

Katy Noland and Mark Sandler. *Signal processing parameters for tonality estimation*. In Proceedings of AES 122nd Convention, Vienna, 2007.

Samer Abdallah, Katy Noland, Mark Sandler, Michael Casey and Christophe Rhodes. *Theory and evaluation of a Bayesian music structure extractor*. In Proceedings of the 6th International Conference on Music Information Retrieval, London, 2005.

Mark Levy, Katy Noland and Mark Sandler. *A comparison of timbral and harmonic music segmentation algorithms*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, Hawaii, 2007.

Journal Article

Katy Noland and Mark Sandler. *Influences of signal processing, tone profiles, and chord progressions on a model for estimating the musical key from audio*. Computer Music Journal, 33(1), 2009.

1.5 Thesis Outline

Chapter 1 This chapter introduces our aims and gives an overview of the thesis and its contributions.

Chapter 2 We describe previous approaches to tonality estimation in chapter 2, in the context of a generic structure that is common to all methods. A discussion is given of different types of tonality model, methods of transforming audio data into a form that fits the model, different approaches to finding the best fitting key within the context of a given model, and the form that the final tonality estimation should take. We discuss methods of tracking tonal centre changes over time, and describe a hierarchical view of tonality that has been taken by some

researchers. A description of previous methods of evaluation for key estimation algorithms is also given.

Chapter 3 In chapter 3 we introduce our model of the relationship between chords and keys.

We perform various optimisation experiments using hand-annotated chord symbols as our test data. The results of the experiments tell us the best version of the model assuming accurate chord transcriptions: a stage that could not be reached if using only audio data.

Chapter 4 The key model is extended in chapter 4 to allow it to operate on audio data, by first performing an automatic chord recognition step. We describe experiments in which we optimise the chord recognition for the upper partial content in the audio signal. We also describe a new variant of the key model which does not require an initial chord classification, but find that the separate chord classification gives better key estimation performance.

Chapter 5 In chapter 5 we address the importance of using appropriate parameters at the very first stages of feature extraction, including the frame size, frame spacing (hop size), and the analysis frequency range. We give measurements of the computational savings possible by adjusting the low level parameters. We also test the effects of using frames that correspond to musical beats, and of removing transients in the audio before the key estimation is performed, but find that neither approach is beneficial. We compare our model to an implementation of a tone profile correlation approach to key estimation, and find that relative performance is dependent on the data set, but our model performs better on the data sets that contain the greatest variety of timbres and styles.

Chapter 6 The experiments in chapters 3–5 use estimation of a single main key for each piece as a means of evaluation. In chapter 6 we discuss the possibilities for an evaluation metric that captures finer detail in the estimates. We propose a metric that is inspired by our primary aim, music retrieval, and show that this metric performs differently from a ground-truth matching measure.

Chapter 7 In our final chapter we summarise our findings, and suggest areas for future research.

Chapter 2

Computational Tonality Estimation

In this chapter we summarise the research to date in the field of computational tonality estimation. We begin by describing the general structure common to all algorithms, then discuss the different methods of implementing each part of the structure. This includes descriptions of different ways of modelling tonality using knowledge from music theory and studies of music perception, techniques for processing audio data and transforming it into harmonic features, and methods of classifying the harmonic features by musical key. We go on to discuss some important issues in automatic tonality estimation: timescale and harmonic progression, the hierarchical nature of tonality, and methods of evaluating tonality estimators.

In order to fully understand the methods described it is necessary to have a basic understanding of musical harmony. An introduction to the important concepts addressed throughout this thesis is given in appendix A, and for more detailed descriptions there are many dedicated music theory texts available (see e.g. [Taylor, 1989, 1991]).

2.1 General Structure of Computational Tonality Estimation Algorithms

Computational tonality estimation algorithms can be divided into two broad categories: those taking symbolic music such as MIDI as their input, and those taking digital audio as their input. Symbolic representations of most music are not generally available, so this research focusses on methods for tonality estimation from digital audio. Some symbolic methods have been adapted for use with digital audio, so we also describe those symbolic methods that are relevant.

All algorithms can be mapped onto the basic structure shown in figure 2.1. First a model of each possible key is created in a suitable space. This is the block in which the most fundamental differences between different algorithms lie, depending on the premise used to define the key or tonality. The next stage is to transform the music under test into the same space as the model.

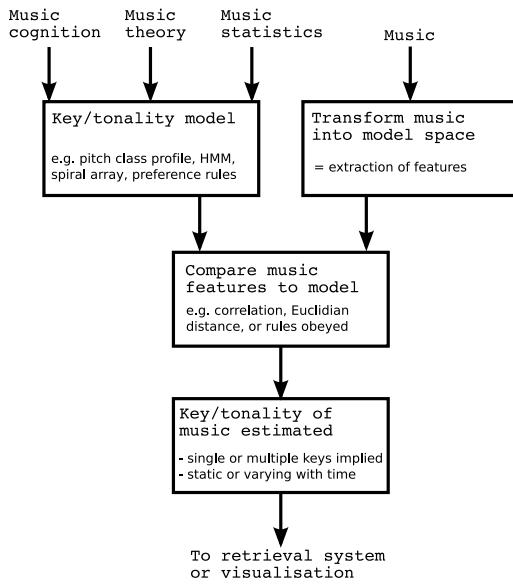


Figure 2.1: Structure of a generic tonality estimation algorithm.

The transformed music, or musical feature set, is compared to the model of each key, and the key model that is most similar to the test music is taken to be the main key of the music. There is also the possibility of using the similarity to other keys as a measure of their relative importance, leading to a description of the tonality.

There are many possible variations to this basic structure, including the type of key model used (e.g. pitch distributions [Gómez and Herrera, 2004b] or preference rules [Temperley, 2001, p. 188], see section 2.2), the method of transforming the music into an appropriate space (e.g. Fourier transform [Chuan and Chew, 2005a] or auditory nerve modelling [Martens et al., 2004], see section 2.3) the type of similarity measurement (e.g. correlation [Krumhansl, 1990, p. 37] or Euclidian distance [Fujishima, 1999], see section 2.4), the timescale of the test music excerpt (e.g. an estimate that changes through time [Chai and Vercoe, 2005] or a single estimate for the whole track [Pauws, 2004], see section 2.5.1), and the type of estimate produced (e.g. a single key classification [Zhu et al., 2005] or a position within a continuous tonal space [Chew, 2001], see section 2.5.2). Each of these aspects will be considered separately.

2.2 Type of Key Model

We divide our discussion of the different types of key model into five sections, addressing perceptual models (section 2.2.1), tone profiles (section 2.2.2), geometric models (section 2.2.3), preference rules (section 2.2.4), and machine learning (section 2.2.5).

2.2.1 Perceptual Models

Understanding how humans perceive tonality is a subject that has received attention from researchers in the psychology and cognition communities. Justus and Bharucha [2002] summarise findings in the perception and cognition of pitch and tonality that use experimental psychology (such as the probe tone method, discussed in section 2.2.2), measurements relating to musical performance (relating types of performance error to the local tonality), and cognitive neuroscience approaches (brain imaging while subjects are exposed to sound stimuli). They also discuss tonality perception in the context of child development, cultural conditioning, and evolutionary development. Auhagen and Vos [2000] give a review of the experimental methods used to investigate tonality induction by humans.

Pitch Height and Pitch Class

One of the most important and undisputed facts to arise from studies of tonality perception is described by Roger Shepard [1964]. He shows that human judgement of pitch can be separated into two attributes, which he calls *height* and *tonality*. To avoid confusing terminology we will refer to them as *pitch height* and *pitch class* respectively. The pitch height is a measure of absolute frequency, whereas the pitch class refers to the note name regardless of the octave at which it is sounded. This separation is strongly supported by music theory, in which chord types are recognised by their constituent notes regardless of the sounding octave or note ordering in frequency. Whilst the chord inversion, given by the bass note, does give some information regarding the chord stability, and there are some cases where the note ordering is important, this extra information is comparatively much less useful than the note names, and certainly could not be used alone to perform harmony analysis.

The pitch class and pitch height can be visualised by positioning notes on a helix such that notes an octave apart are positioned in a vertical line, and rotation in the horizontal plane corresponds to a change in pitch class. Figure 2.2 is a diagram of such a helix, as shown by Shepard [1999, p. 158]. There are various alternatives to this helical arrangement, such as toroids [Shepard, 1999, p. 162] [Krumhansl, 1990, p. 43], and planar arrangements [Lerdahl, 2001, p. 44, fig. 2.2(b)], [Schönberg, 1969, p. 20], all of which separate pitch class from pitch height and attempt to arrange the pitch classes such that harmonically similar notes are geometrically close together.

2.2.2 Tone Profiles

The division of musical pitch into pitch class and pitch height has led to a further important product of tonality induction research: the concept of the tone profile. Tone profiles are the basis of many automatic key and tonality estimation algorithms, including those used in this thesis.

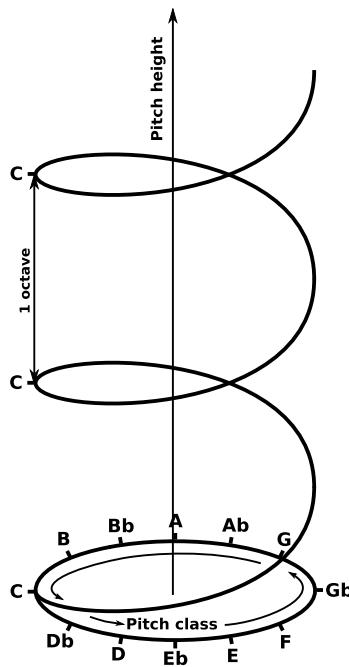


Figure 2.2: Shepard's helical model of pitch [Shepard, 1999, p. 158].

A tone profile is a twelve-element vector where each element represents one of the twelve pitch classes. The value of each element is proportional in some way to the importance of the pitch class within a given key, so the profile can be considered a model of the key, based only on its expected pitch class distribution. The profiles do not contain any information regarding pitch height. It is generally assumed that the tone profiles are transposition invariant, such that the C element has the same value in C major as the G element in G major. This leads to two distinct profiles, one for major keys and one for minor keys. Twelve-element tone profiles assume equal temperament in the music. In most cases this is an approximation, which has been applied in order to simplify the model. A more complete model would account for different temperaments.

Tone profiles have been used extensively for key estimation from both symbolic [Krumhansl, 1990, ch. 4] [Temperley, 2004] and audio data [Gómez Gutiérrez, 2006, Pauws, 2004, İzmirli, 2005a]. The pitch class distribution in the music is measured (extraction of musical features), and then compared to the tone profile for each key using a correlation measure (see section 2.4). The profile with the closest match is taken as the key estimate.

There are different ways of arriving at the tone profiles, which can be divided into three categories: those based on music theory [Chai and Vercoe, 2005] [Temperley, 2004] [Gómez Gutiérrez, 2006, p. 19], those based on cognitive studies [Krumhansl, 1990, p. 30] [Aarden, 2003], and those based on collecting statistics of real music [Purwins and Blankertz, 2005] [Temperley, 2007, p. 59] [Gómez Gutiérrez, 2006, p. 26] [Noland and Sandler, 2007].

The profiles based on music theory tend to be simple, such as the flat profiles used by Gómez

[2006, p. 19], where the values of elements corresponding to diatonic notes are set to be equal (and positive), and values for non-diatonic notes are set to 0, as shown in figure 2.3(a).

The profiles derived from cognitive studies are heavily dependent on having reliable listening test subjects and suitably generic test stimuli, and have been criticised for measuring relationships that subjects have learned through music theory training rather than relationships based on unbiased judgements [Purwins, 2005]. However, they do provide more information than do the flat profiles about the relative importance of pitches in a given key. Figure 2.3(b) shows the tone profiles derived by Krumhansl using listener rating tests [Krumhansl, 1990, p. 30], normalised to sum to 100. A hierarchy of pitch classes emerges that reflects knowledge of music theory: the most important pitch is the tonic (C in the figure), followed by the other notes of the tonic triad (E/E \flat and G in the figure), followed by other diatonic pitches, and finally the least important group contains the non-diatonic pitches.

Several researchers have proposed modifications to the Krumhansl profiles to make them more suitable for the purpose of key estimation. Temperley [2001, p.176–180] hand-tunes the Krumhansl profiles to emphasise the raised 7th in the minor scale, which is particularly important in key estimation because of its role in the perfect cadence, and to reduce the values for all non-diatonic pitches. He demonstrates improved bar-by-bar key-finding results on a monophonic piece by J. S. Bach.

Gómez and Herrera [2004b] modify the Krumhansl probe tone profiles to emphasise diatonic pitches according to their occurrence in the tonic, dominant and subdominant chords. Each of the Krumhansl profile values is taken to represent the importance of the triad of which it is the root, so for example the final profile value for the 5th scale degree is the sum of the original tonic and dominant values, since the 5th degree of the scale is contained in both the tonic and dominant chords. Contributions of other chords are not considered. This approach results in a much higher value for the 5th scale degree, and the non-diatonic pitches have values of 0, which brings this closer to a music theory profile.

İzmirli takes a similar approach [İzmirli, 2005a], by point-wise multiplying Temperley's modified version of the Krumhansl profiles [Temperley, 2001, p. 180] with a flat diatonic profile, meaning that all non-diatonic profile values are set to 0. This kind of approach helps to emphasise the most important pitches in the key. İzmirli's method gave the highest score in the first MIREX audio key-finding contest [MIREX competition, 2005a].

The third approach to creating profiles is dependent on the corpus from which the statistics are derived, and so a very large and varied corpus is required for a general profile, or a style-specific profile can be produced from a more limited database. Figure 2.3(c) shows the profiles created by Noland and Sandler [2007], which were generated from recordings of Bach Preludes

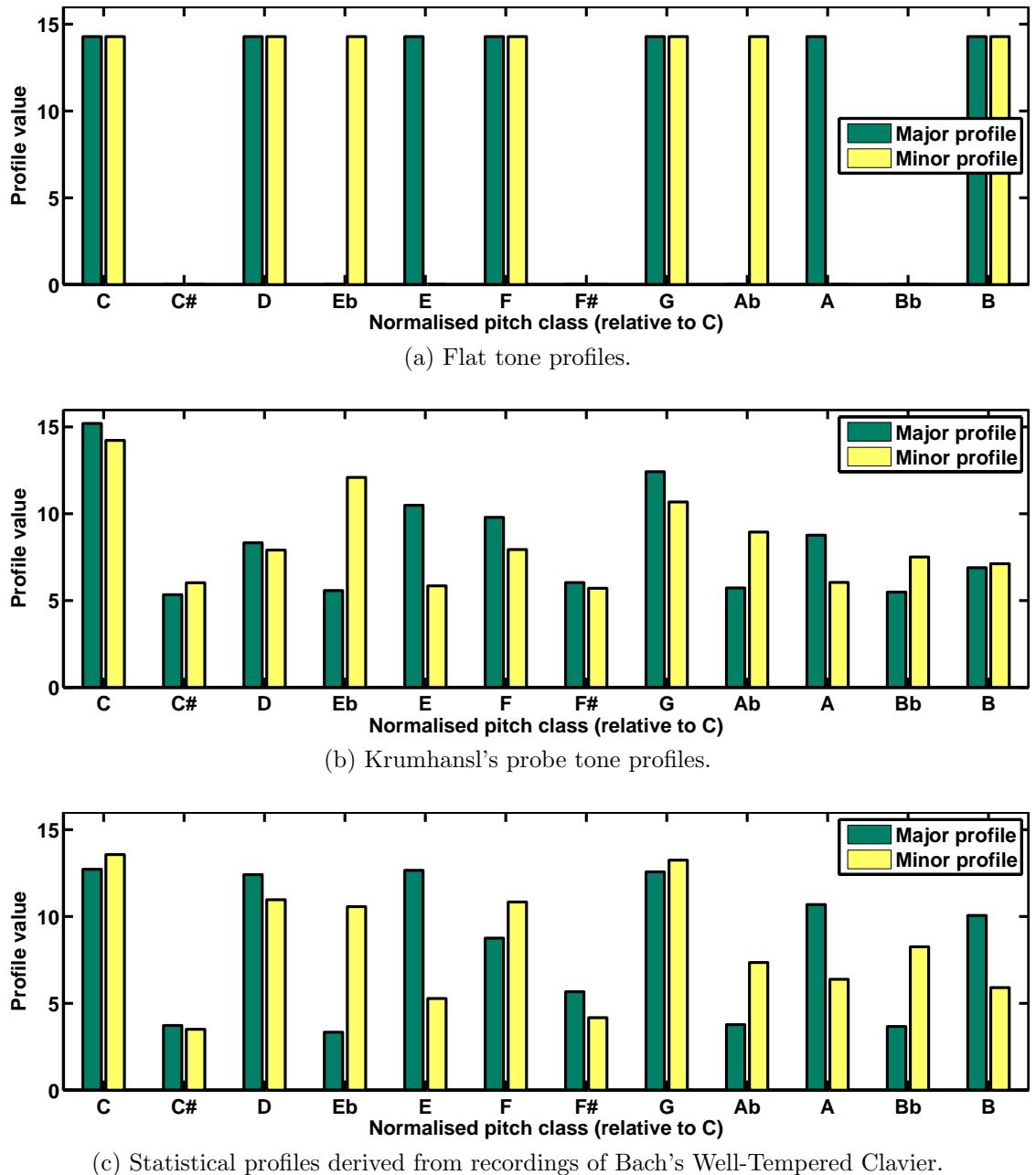


Figure 2.3: Example tone profiles, shown relative to C, all normalised to sum to 100.

and Fugues. The Preludes and Fugues take into account keyboard technique in different keys, so the averaged profiles will emphasize those scale degrees that are particularly important in all keys, and are therefore likely to be the scale degrees that are most important for defining the key. The profiles do show some obvious similarities with Krumhansl's probe tone profiles, but exhibit greater differences between the major and minor profiles for the 6th and 7th degrees of the scale, as well as a much higher relative value for the 2nd scale degree. These particular profiles were derived from audio recordings, so will account for upper partials present in the signal (see sections 2.2.2 and 2.3.7 for further discussion of upper partials).

In her PhD thesis, Gómez [2006, ch. 4] evaluates the performance of seven different tone profiles for key-finding on the first 30 seconds of 5 different data sets. No particular profile type is shown to be always superior, but she concludes that profiles based on psychological data perform better in general than statistical profiles, but that performance is genre-specific.

There is some disagreement regarding the meaning of the profiles. Those derived from finding the pitch distribution in real music clearly represent the likelihood of each pitch occurring within a given key, and so are directly applicable to key-finding algorithms that find a correlation between the actual pitch distribution and the key profile. However, those profiles derived from the results of listening tests do not have such a clear meaning. Krumhansl considers her tone profiles to be "*an indicator of a psychological orientation to a musical key*" [Krumhansl, 1990, p. 30]. She shows that there is a strong correlation between the profiles and the statistical distribution of pitches in a database of tonal music, and later uses them in a key-finding algorithm based on the pitch distribution. However, Aarden [2003] shows that the probe tone profiles are more closely related to the expectancy of pitches at the ends of phrases due to the nature of the probe tone test, which requires the phrase to stop before the listener makes a judgement. He derives an alternative profile that he argues represents the expectancy of pitches within a continuing phrase, due to his reaction time test method, and differs from the Krumhansl profile most significantly in the order of importance of the tonic, dominant and mediant degrees. Sapp observes [Sapp, 2005] that Aarden's profiles give a closer approximation to music-theory expectations than do Krumhansl's for his tonal structure visualisations, with Krumhansl's profiles favouring the dominant key at the expense of the subdominant. In order to overcome the same problem, Toivainen and Krumhansl [2003] use a method of continuous probe tone ratings taken as the musical stimulus is heard.

Such studies regarding the meaning of tone profiles are certainly valid and interesting, but perhaps the need for them indicates a limit to the efficacy of tone profiles for general key modelling. Pitch class distributions have been shown to capture much of the essence of a key, but in reality perception of tonality is a much more complex process that is dependent on many other factors, including pitch ordering and phrase structure, as well as cultural influences [Krumhansl, 2000].

Adaptations of Symbolic Tone Profiles for Audio Features

The cognitive and theoretical tone profiles are built from the importance of pitch classes within a given tonality. An additional difficulty occurs when working with audio data. Real instrument sounds consist not only of the fundamental frequency of the note being played, but also multiples of that frequency, called *upper partials*. This means that a simple frequency transformation shows more peaks than there are notes being played, so it is insufficient as a pitch detector. Upper partials are also commonly called *harmonics*, but we avoid this term so that it is not confused with the word *harmonic* used as an adjective that indicates an association with tonal harmony. It is possible for upper partials to occur at non-harmonic frequencies, but in this thesis we restrict our discussion to harmonic partials. The numbering conventions for harmonics and partials are different: the fundamental frequency is the first harmonic, but the zeroth partial, so the second harmonic is the same as the first partial, and so on.

Pauws [2004] applies the Krumhansl profiles directly to a method of key-finding from musical audio with no treatment of the upper partials. He replaces the measure of pitch class distribution in the music with a 12-bin chroma vector (see section 2.3.4) for equally-spaced frames (see section 2.3.1), giving an energy measure for each of the 12 pitch classes. The correlation between the chroma values and Krumhansl's tone profiles is then calculated in all twelve possible rotations, the highest correlation value pointing to the most likely key. The effects of upper partials falling into different pitch classes are ignored, but main key estimation performance is still at 75 % for classical piano sonatas.

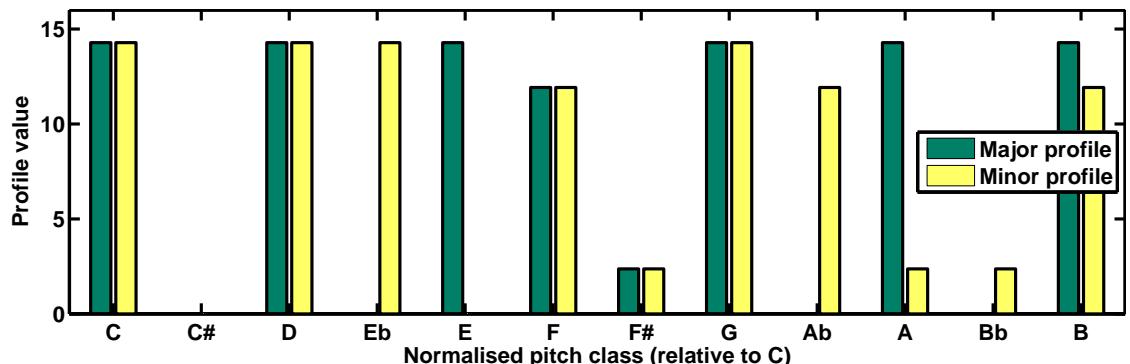
Several adaptations to the tone profiles have been proposed for the purpose of key estimation from audio, that aim to take account of upper partials present in the signal.

An early and simple approach is presented by Purwins et al. [2000], who account only for the second upper partial, which sounds at an interval of a perfect 12th above the fundamental. They simply add 1/3 of its tone profile value to the profile value for the fundamental. This seems counterintuitive, since one would expect the profile value for the upper partial pitches to be increased to match the upper partials in the audio. However, the adapted profiles are shown to closely match average chroma features for cadential figures played on a piano, although no comparison with alternative profiles is offered.

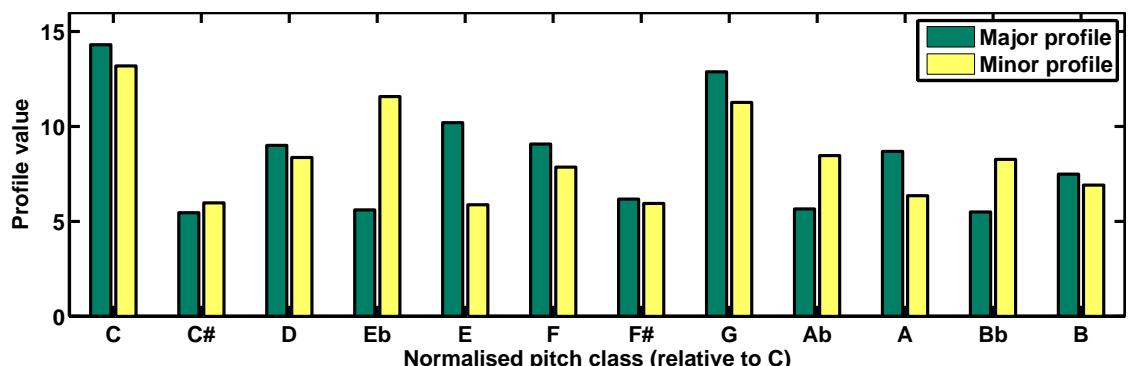
Gómez and Herrera [2004b] do increase the profile values for upper partial pitches, and also account for a greater number of partials. They model the partials as decaying linearly with frequency, and add in an appropriately weighted amount of the profile value corresponding to the fundamental to the value for the pitch at which the upper partial sounds. For example, G is the second upper partial of C, so the final profile value for G will be the original value for G plus a small proportion of the original profile value for C. This is a simple strategy, but means

that the profiles are not customised for a particular instrument so are likely to be more generally applicable than instrument-specific profiles.

In her PhD thesis Gómez [2006, p. 77–78] demonstrates improved performance using the same technique but with an exponential decay model of the partials, which we adopt for our chord templates in section 4.2.1. Figure 2.4 shows the flat profiles and Krumhansl profiles from figure 2.3 after an exponential decay model of the first 3 upper partials is applied. The difference is most obvious for the flat tone profiles in figures 2.3(a) and 2.4(a), in which the major and minor profile values for F sharp become non-zero when upper partials are modelled, since F sharp is the second upper partial of B. The values for F are relatively reduced since they do not include contributions from B flat, the pitch class of which F is the second upper partial. Similar changes are apparent in the sixth and seventh scales degrees of the minor profile.



(a) Flat tone profiles, modified to model 3 upper partials.



(b) Krumhansl's probe tone profiles, modified to model 3 upper partials.

Figure 2.4: Example tone profiles including an exponential decay model of the first 3 upper partials, shown relative to C, all normalised to sum to 100. Compare with figure 2.3 which does not include upper partial modelling.

İzmirli [2005b, 2005a] generates profiles using average spectra of real instrument samples, weighted by the profile value of the fundamental pitches. He uses piano samples, but suggests using monophonic samples from any real or synthesised instrument. Such profiles will be instrument-specific, so should outperform generic profiles if the music is played on the same instrument, but

should perform less well on music played on other instruments.

Although the Krumhansl probe tone experiments were conducted using Shepard tones [Shepard, 1964], which each contain only one pitch class, Huron and Parncutt [1993] argue, with support from pitch perception literature, that other tones could have been perceived by the subjects at upper or lower partial frequencies. They modify symbolic input to include these additional tones and to include a model of pitch memory, and show that correlation of the modified input with the Krumhansl probe tone ratings better matches listener perceptions of key relations than does correlation with only the notated pitches. An evaluation of key estimation performance is not the purpose of the article, and it is not clear how such a model could be extended for audio data which is already rich in upper partials.

Martens et al. [2004] use a different kind of key profile, based not on semitones, but on critical bands [Howard and Angus, 2006, p. 74]. The function of the ear through to the final auditory nerve pattern is modelled, and the key templates are derived from the 69-dimensional auditory nerve patterns occurring after a key-defining sequence, chords I-IV-V-I. The chords are composed of Shepard tones which contain sinusoids that are spaced apart by a whole number of octaves. The model includes some pitch memory, so the first chord will influence the final nerve pattern. A principal component analysis is performed on the 69-dimensional templates and the 4 highest variance components are used to create a 4-dimensional subspace, which is the space in which the final key templates are represented and the subsequent distance measure is calculated. Although the feature space is quite different from other profile methods, this is still a method that compares an audio signal that is rich in upper partials with a template that does not fully account for their presence.

The upper partials problem has also been addressed by adapting the musical features for use with tone profiles that are designed for symbolic music representations. These approaches are described in section 2.3.7.

2.2.3 Geometric Models

There are also methods of computational tonality estimation that make use of the geometric representations described in section 2.2.1. These methods bear similarities to the probe tone techniques: the 12-dimensional pitch class distribution vector is mapped onto a point in the geometric model, which is typically of lower dimensionality, and the point's distance from each key template is measured not in terms of correlation, but by measuring the Euclidian distance in the lower-dimensional tonal space.

In the symbolic domain one of the most well-known methods is that of Chew [2001], who makes use of a helical pitch model in her Spiral Array. She positions pitches at intervals of a major third apart (and so also pitches one octave apart, since an octave contains three major thirds) directly

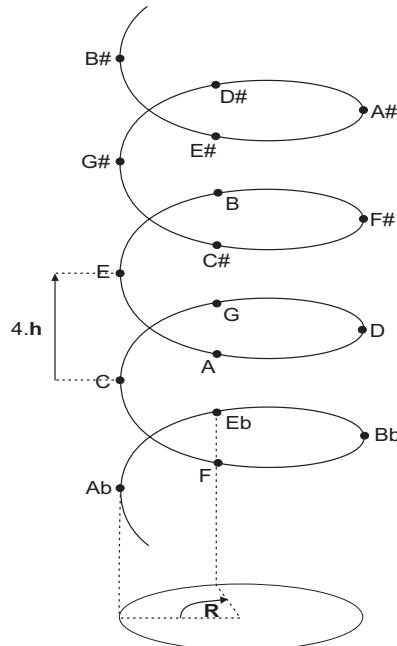


Figure 2.5: Chew's Spiral Array [Chew, 2001]. Adjacent pitches are separated by pitch height h and rotation R .

above each other, and pitches that are a perfect fifth apart are positioned adjacent on the spiral, as shown in figure 2.5. She finds a tonal centre within the helix that is calculated from the notated pitches weighted by their duration, and measures its distance to tonal centres derived from weighted averages of diatonic tones. The inside of the helix is therefore defined to represent harmonic regions, with the pitch classes remaining on the surface of the shape. Mardirossian and Chew [2005] use the Spiral Array with a pitch spelling algorithm in order to compute key estimates from MIDI, which does not differentiate between enharmonically equivalent pitches, for example, F sharp and G flat.

Lerdahl defines a whole system of tonal analysis based on stepwise distances in a chart of pitch classes [Lerdahl, 2001, p. 47], which has different levels corresponding to the hierarchy of chromatic tones, diatonic tones, triad tones, fifth tones and the tonic pitch. He illustrates the distances he calculates using various geometrical shapes, including a cone for pitch class relationships [Lerdahl, 2001, p. 50] and toroids for chord and key relationships [Lerdahl, 2001, p. 57, p. 66].

Chew's method has been adapted for use with audio data in the same way as the profile correlation method, by replacing the pitch class distribution feature with a chroma feature calculated from the frequency content of the audio signal (see section 2.3.4). Chuan and Chew [2005a] directly extend Chew's center of effect generator using chroma features, and Harte, Sandler and Gasser propose a tonal change detection function [Harte et al., 2006] that maps chroma values onto a 6-dimensional model of tonal space that is related to the 4-dimensional Spiral Array. The

additional 2 dimensions are introduced to highlight circularities formed by pitches at intervals of a minor third, as well as the perfect fifth and major third relationships modelled by the Spiral Array. We use the 6-dimensional model for our tonal similarity measurements in chapter 6.

Krumhansl [2005] provides a brief overview of historical geometrical representations of harmonic relationships, and highlights some important weaknesses. Harmonic relationships are easier to visualise in lower dimensional spaces, but all of the information cannot necessarily be represented in low dimensionality. İzmirli [2006] calculates distance measurements in various low dimensional spaces created by taking the first principal components of a 12-dimensional chroma space, and found that there was hardly any loss of key finding accuracy when using 6 components, and no serious deterioration until only 2 components were used, implying that a 6-dimensional representation is most appropriate.

Krumhansl also identifies a need to represent how strongly a key is implied, for which the colour and size of a representative shape have been used [Gómez and Bonada, 2005, Mardirossian and Chew, 2007]. Finally, she has observed that harmonic relationships are not symmetrical and vary depending on the context, which for a geometric approach would require a dynamic model that re-forms as the tonal context changes. Development of such dynamic models has not been attempted to our knowledge.

2.2.4 Preference Rules

An alternative approach for the tonality model is one based on a set of rules defined from music theory principles. Such systems have been explored less thoroughly than tone profile models, but some notable methods have been proposed.

One of the earliest methods of key estimation from symbolic music is that proposed by Longuet-Higgins and Steedman, as described by Temperley [2001, p. 169]. The algorithm starts at the beginning of the piece, and each time a new pitch is sounded any keys that do not contain the pitch are eliminated. The process continues until only one key remains. A simple tonic-dominant preference rule is added in the event of a tie break between two equally likely models: the first tone of the music is assumed to be the tonic, or if that is not possible given the possible keys the first tone is taken to be the dominant of the key.

Temperley also interprets his own profile-based symbolic key-finding method as a preference rule approach [Temperley, 2001, p. 188]. He describes a rule to prefer a key for a given musical segment that is compatible with the pitches in the segment, according to his modified tone profiles, and a rule to prefer to minimise the total number of key changes in a piece.

Zhu et al. [2005] present a rule-based approach to key finding from audio, based on a pitch profile of the test music that contains information about the most important pitches. (They suggest a unique method for extracting this profile: see section 2.3.7.) The profile is arranged

according to the circle of fifths, so that diatonic pitches are grouped together. Essentially the most sounded consecutive 7 notes are taken to be the pitches of the scale, and major/minor detection is achieved by assuming that the tonic and dominant are the most sounded 2 consecutive pitches. This model is a development of earlier work by Zhu and Kankanhalli [2003] in which the model arranges pitches around a circle in pitch height order, and the frequency distance of a pitch to the nearest scale note (up to 100 cents) is used as an error metric for the degree of fit.

Rizo and Iñesta [2005] describe a key-finding method that is essentially a preference rule approach, although pitches are visualised in a tree structure. The rules are such that a key is given preference if the notes present form diatonic triads in that key, with greater preference given when chords I and V are present, and less when only diads and single pitches within the key are present. It operates on MIDI input, and came in third place in the 2005 MIREX symbolic key finding contest [MIREX competition, 2005b].

Pardo and Birmingham [2002] approach chord recognition by defining a set of 6 chord classes with simple binary templates, and a score for a particular chord is calculated based on how many notes present in an already segmented part of the music match the template, with negative scoring for notes that do not match the template or template notes that are not present in the music. This approach is similar in many respects to a tone profile correlation method, but the authors go on to describe a set of rules for tie-breaking based on musical knowledge (the root is assumed most important, common chord types such as major and minor triads are assumed most likely, and the roots of diminished 7th chords are assumed to resolve up one semitone). Although this approach is intended for chord recognition purposes, modification of the templates could make it applicable to tonality estimation.

All of these rules are aimed at hard key classification, which is not the ultimate goal of this research. It is also difficult to evaluate which rules are most important. All of the rules make musical sense, but the literature gives little explanation of the reasons for choosing them, and they are likely to be specific to the particular sets of musical data for which they are intended. Hence we focus on more general approaches to tonality estimation.

2.2.5 Machine Learning

The field of artificial intelligence has produced numerous methods for developing automatic data classifiers, which are suited to classification when the data are multidimensional and the class boundaries are not known. The algorithms either find suitable boundaries in a set of training data where the correct classification is known, then use the same class boundaries for new data (supervised learning), or if no labelled training data is available there exist algorithms that can find the natural groupings in a set of data, in a given feature space (unsupervised learning). Detailed descriptions of the most common classification techniques are given by Duda, Hart and

Stork [2001].

Machine learning methods have become popular for automatic key estimation because they do not require an exact definition of a key in terms of its pitch class distribution. The discussions regarding tone profile meanings become less important because the algorithms are able to learn relationships from the data.

Gómez [2004b] compares a tone profile correlation approach to several different machine learning methods, including neural networks, binary trees and support vector machines (SVMs). All of the machine learning algorithms were trained on 12-bin chroma vectors (see section 2.3.4 for an explanation of chroma vectors). She found that the best machine learner was a multilayer back-propagated perceptron with a single hidden layer of 20 units, but that the improvements over other learners and the profile technique were small, for a large database of classical music from the FreeDB¹ database.

Martens et al. [2004] use an automatically-generated classification tree to separate key templates generated from an auditory nerve analysis of artificially-generated training data. The artificial data consists of 264 key-defining chord sequences played with a variety of synthesised instruments. On one piece, the tree was found to perform similarly to a distance-based measure in the auditory nerve feature space, but was more robust to changes in dimensionality (number of splits in the tree or number of auditory nerve features used).

More recently İzmirli [2007] proposed a quite different approach that applies non-negative matrix factorisation to a chromagram. The factorisation operation decomposes the chromagram into a matrix of chroma clusters and a sparse mixing matrix. The chroma clusters represent templates for the pitch class distribution of a particular key or combination of keys, and the mixing matrix represents the contribution of each cluster to each long chroma frame. The mixing matrix can be viewed as a segmentation of the music with respect to local key, the definition of local being determined by the length of the chroma windows. This is a novel approach to classifying chroma frames according to their tonal content, but is still essentially based on pitch class distributions.

For many machine learning techniques it is not clear which features the algorithm uses for classification, so the final results are difficult to explain and it is hard to suggest improvements beyond using more training data. One exception to this is hidden Markov models (HMMs), which allow the meaning of the parameters learnt to be better understood than for other machine learners, so have been used in several tonality estimation algorithms. HMMs are Bayesian models in which a series of hidden states is defined. The state can change at each time step with a certain probability, and at each time step the current state emits observable features with a certain probability. The current state is dependent only on the previous state for a first order model.

¹<http://www.freedb.org>

Standard algorithms exist for learning the different sets of probabilities and decoding the final model, which we describe in chapter 3, section 3.1, in the context of our HMM for key estimation. Rabiner [1989] and Cox [1990] also both give an introduction to HMMs.

Raphael and Stoddard [2003] define an HMM where each hidden state corresponds to a chord within a key context, with the chords limited to the seven diatonic triads within each major or harmonic minor scale. There are therefore 24×7 (number of keys \times number of chords per key) hidden states. The observation data are collections of pitches from consecutive sections of music in MIDI format, each section of a length related to the musical tempo (e.g. one bar). Also observed is a metrical position for each pitch (beat 1, 3, or either 2 or 4, restricted to 4/4 time), which allows the specification that chord tones are most likely to be sounded on strong beats. Various transposition invariances are exploited in order to reduce the number of parameters that must be trained, for example when the key remains constant the chord transitions (represented relative to the key) do not depend on the key. No evaluation is given, due to the difficulties in defining a suitable evaluation measure (see section 2.6).

Temperley [2004] develops a Bayesian method of symbolic key recognition, which is essentially an HMM, although specific HMM terminology is avoided. Regarding a pattern of notes as a surface (observation) and an underlying key progression as a structure (hidden state sequence), he applies Bayes' rule

$$P(\text{structure}|\text{surface}) = P(\text{structure})P(\text{surface}|\text{structure}) \quad (2.1)$$

where $P(\text{structure})$ can be calculated from a set of heuristically-derived modulation probabilities, and $P(\text{surface}|\text{structure})$ can be calculated using a tone profile to indicate the likelihood of a pitch class set given the key. Performance was measured according to the number of short segments (of the order of 1 bar length) that were correctly classified as the right key. The Bayesian approach was shown to be marginally better than two profile-based approaches, when all three techniques were using key profiles derived from the test data. This method assumes that all modulations are equally likely and that the likelihood of each pitch occurring in a given key is independent of every other pitch. These are acknowledged as simplifications which could be improved.

Also for symbolic music, Noll and Garbers [2004] have built a software tool, the *HarmoRubbette*, which allows the adjustment of parameters of various similar HMM-based key estimation techniques on a particular excerpt of music; and Shmulevich et al. [2001] take a similar approach to estimate key based on tone relations in a monophonic melody.

For key estimation from audio, Chai and Vercoe [2005, 2006] use chromagram data as the observations for two HMMs. In the first model, each state represents a key pair (major and its relative minor); in the second, each state represents a mode (major or minor), and decoding both models gives both the root and the mode of the key. No training was carried out on the

HMM parameters, and no preference was given to any initial key or key transition except that the probability of staying in the same key was set very high. The observation probabilities were set for both models with binary vectors determined by knowledge of the pitches present in each key and each mode.

Burgoyne and Saul [2005] define an HMM that does not use the usual Gaussian probability density functions for the observations, but Dirichlet functions, chosen for the property that they accentuate the relationships between elements in the observation vectors while giving less importance to the absolute magnitudes. This property is desirable when the observations are pitch class profiles because the relative energy in each pitch class is more important for chord recognition than the volume at which the pitches were sounded. In their model each hidden state corresponds to a chord within a key context, as with the Raphael and Stoddard model [Raphael and Stoddard, 2003]. Initial transition probabilities are not explicitly specified, but they are based on some music theory principles, with probabilities of changing chord within a key set uniformly and probabilities of changing key based on constraints given in Lerdahl's *Tonal Pitch Space* [Lerdahl, 2001]. The probabilities were then trained on five Mozart Symphonies, and the model's performance on one additional movement is discussed. The model's chord recognition performance is good on the short excerpt shown, with local key recognition less good but the estimated keys are nonetheless harmonically related to the correct keys.

Peeters [2006b] also uses HMMs for key estimation but creates a separate model for each key, learnt from chroma vectors of a set of classical pieces in known keys. With this approach the exact meaning of the hidden states is not known, so calculating the most likely state sequence is not useful, it is only meaningful to determine which complete model is most likely to have generated the observed chroma vectors. This means that each track can only be assigned one overall key. He compares the key estimation results using various numbers of HMM states and various numbers of Gaussian mixtures for the observations with a profile-based approach, but finds that the profile approach still performs better according to the MIREX score (based on percentage of tracks correctly classified but credit is also given for closely related keys, see table 2.1 on page 53).

Lee and Slaney [2008, 2007] also develop a separate HMM for each key, but in their model each state represents a chord, so the final decoding produces both a chord transcription and a main key classification. For their observation features they convert chroma features into the six-dimensional tonal space proposed by Harte et al. [2006], and showed that this produced higher chord recognition accuracy than using the chroma features directly. A noteworthy aspect of their approach is that the model parameters are learned from synthesised music from MIDI files, using a symbolic key recognition system to provide chord and key labels for supervised training. The use

of synthesised training data allows for extensive supervised training without the need for hand-annotated ground truth, but the output of another automatic harmony analysis algorithm cannot be considered equivalent to human annotations. However, the authors report accuracy figures that are equivalent to or exceed those of other state of the art methods, so the imperfections of the simulated ground truth are shown to be insignificant in comparison to the benefits of a vast training set.

Pickens et al. [2003] make use of Markov modelling to perform polyphonic score retrieval using harmonic features. A Markov model is a model of a sequence of states in which the current state is dependent on the previous state, and the states are directly observed, unlike a hidden Markov model which includes a separate observation layer. Pickens et al. use a music transcription algorithm to transform an audio query into a symbolic format, then the harmonic features for both the query and source collection can be extracted using symbolic modelling techniques. Pitch classes with simultaneous onset times are grouped together, and after some smoothing in the time domain their similarity to each of the 24 major and minor triads is calculated according to the number of shared pitch classes. The similarity to every possible chord, interpreted as a probability distribution across the modelled chord set, is used as the harmonic feature. They construct a Markov model from the sequence of harmonic features. They do not aim to perform tonality estimation, but their models share some aspects of the models we present in chapter 3 in their use of chord distributions as higher level harmonic features.

One further method that should be mentioned is that of Bello and Pickens [2005], whose purpose is extraction of a mid-level harmonic feature, evaluated in terms of chord recognition performance. The key estimation model we present in chapter 3 is strongly related to their work. They define an HMM in which each state represents a major or minor triad, and each observation is a chroma feature. They use the distance around the circle of fifths as a measure of chord relatedness, and use chord relatedness measures directly as chord transition probabilities. The observations are chroma features, with observation probabilities that model only the three pitch classes belonging to each triad as present. They train only the state transition probabilities, on a per-song basis, as we do for our key estimation model. Papadopoulos and Peeters [2007] present a comparison of other models for chord recognition that use HMMs with chroma features as observations.

2.3 Extracting the Musical Features

Having defined a tonality model, it is then necessary to transform the raw audio into a form that is appropriate for comparison with the model. Given that the fundamental pitches of notes are the feature of most interest for tonality estimation, a spectral representation of the audio signal

is the most useful. The whole feature extraction process typically consists of some time domain signal processing, followed by a transformation into the frequency domain, sometimes followed by some frequency domain processing.

2.3.1 Time Domain Signal Processing

The digital audio signal on a CD is sampled at a rate of 44.1 kHz with 16-bit precision, and has two channels, left and right. This results in a bitstream of $44100 \times 16 \times 2 = 1411200$ bits per second (ignoring overheads). We describe three methods that can be used to reduce the processing load, either by reducing the number of bits to process or by dividing the audio into manageable chunks.

Stereo to Mono Conversion

A simple way to halve the amount of input data is to convert from a 2-channel stereophonic (stereo) signal to a 1-channel monophonic (mono) signal. The information lost in carrying out this operation is the position of each sound source within the stereo field. The location of a sound source does not affect the harmony of the music being played, so all of the experiments using digital audio described in this thesis are carried out on mono signals. We use a simple conversion method which retains information from both the left and right channels, so sound sources from any location are equally represented in the mono signal.

$$M = \frac{L + R}{2} \quad (2.2)$$

where M is the mid signal (which we use as the mono input for our experiments), L is the left signal and R is the right signal.

Downsampling

Frequencies above half the sampling rate cannot be represented in a digital system [Nyquist, 1928], so a CD signal contains all of the information up to $44.1/2 = 22.05$ kHz, just above the highest audible frequency (approximately 20 kHz). It is possible to reduce the sampling rate in order to reduce the amount of data per second, but prior to the downsampling operation frequencies above half of the new, lower sampling rate must be removed. This means that some high frequency audible sounds will be lost, which may contain useful information for tonality analysis. In addition, with real filters it is not possible to completely suppress the high frequencies, only to reduce their level, so after downsampling there will always be small components derived from them that appear at a lower, usually inharmonic pitch. This is known as aliasing. If the aliased components have too high a level they will interfere with the wanted signal and reduce the performance of a tonality estimation algorithm. The amount of attenuation applied to the aliased components is entirely dependent on the filter used, but might typically be in the region

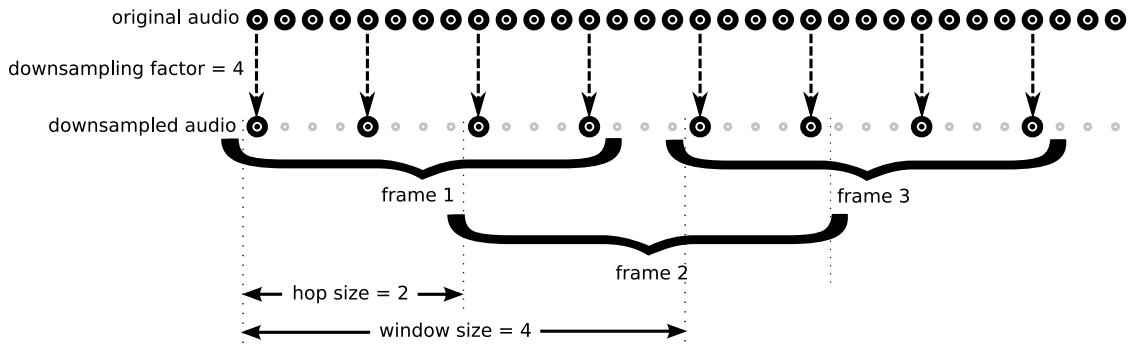


Figure 2.6: Illustration of downsampling factor, window size and hop size.

of 60–100 dB. Downsampling also results in fewer samples per second, which means that the time resolution of the signal is reduced by the same factor as the sampling rate.

Framing

Framing is an operation that divides a long signal into small chunks that are processed serially. This reduces the computer memory requirements considerably, and means that analysis algorithms can be made scalable such that the processing time is approximately proportional to the signal length. Dividing the audio into frames also allows analysis of how a particular feature, such as the spectrum, changes through time. Figure 2.6 illustrates the downsampling and framing operations.

The length of each frame determines how accurately changes in a spectral feature can be located in time. However, there must always be a compromise between time and frequency resolution: the shorter the time frame, the coarser the spectral resolution. In addition, the frames must be at least long enough to contain one complete period of the lowest signal frequency of interest.

One method of improving the time resolution for a given frequency resolution is to overlap the frames, so that the time interval between the starts of consecutive frames, known as the hop size, is less than a full frame length. The hop size is usually measured as a fraction of the full frame length so in figure 2.6 the hop size is half a frame. Although a smaller hop size allows greater time resolution for signal analysis, it also generates more features per second than a longer hop size, which means that there is more data to process and consequently an increase in processing time.

Window Function

The framing operation is equivalent to multiplying the signal by a rectangular window, such as the one shown in figure 2.7(a). In the frequency domain, this means that the true spectrum of the signal is smeared as a result of convolution with the frequency domain representation of the

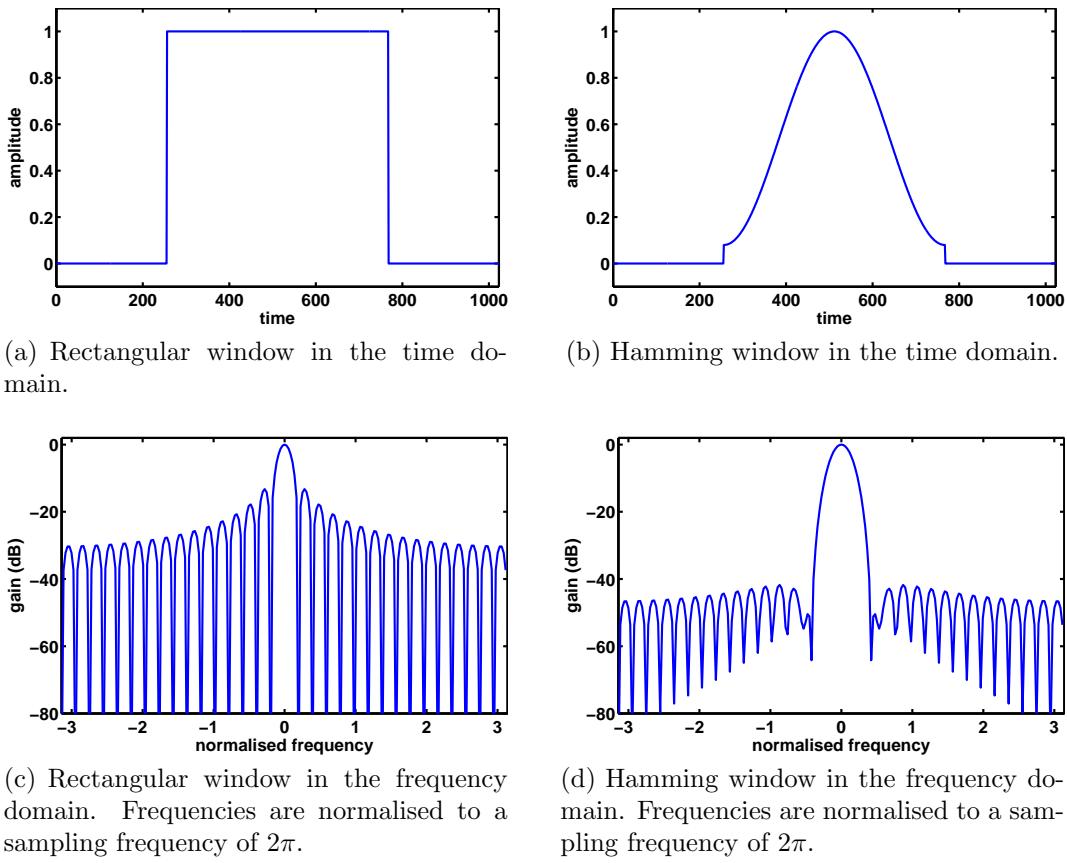


Figure 2.7: Rectangular and Hamming windows in the time and frequency domains.

rectangular window, which is a sinc function (shown in decibels in figure 2.7(c)). The spectral smearing can be controlled by using a window that is a different shape to the rectangular window, such as the Hamming window shown in figure 2.7(b), which tapers the amplitude of the audio frame and so reduces the effects of the harsh cutoff at the edges of the rectangular window. In the frequency domain, application of a Hamming window results in more local smearing of the signal, because the Hamming window has a wider main lobe in the frequency domain than the corresponding rectangular window, but less smearing across the full spectrum, because the side lobes of the Hamming window in the frequency domain are at a lower level than those of the rectangular window. Alternative windows have been defined, such as Hann, Kaiser and Blackman windows, which all control the balance between the main lobe width and side lobe height in different ways [Mitra, 2001, p. 454].

2.3.2 Discrete Fourier Transform

Having reduced the audio bit rate and divided the signal into frames, a frequency transform can be applied to each frame of the data. The discrete Fourier transform (DFT) is the most widely

used spectral transform, and is given by

$$X(k) = \sum_{n=0}^{N_F-1} x(n)e^{-2\pi j nk/N_F} \quad k = 0, 1, \dots, N_F - 1 \quad (2.3)$$

where $x(n)$ denotes the signal in the time domain and $X(k)$ the signal in the frequency domain, and N_F is the number of samples in either time or frequency. Further details about the DFT and its properties can be found in almost any digital signal processing (DSP) textbook [Mitra, 2001, for example]. When the DFT is applied to short, sequential frames of data, as in this case, it is often referred to as the short time Fourier transform (STFT).

The DFT is well understood and can be made very computationally efficient by use of a fast Fourier transform (FFT) algorithm [Cooley and Tukey, 1965]. However, it suffers from the drawback for the purposes of music analysis that the frequency samples are spaced linearly in the frequency domain, such that

$$f_{i+1} = f_i + F \quad (2.4)$$

where $F = F_s/N$, F_s is the sampling frequency, and f_i is the centre frequency of the i th bin. In the equal-tempered chromatic scale, however, the frequencies are spaced logarithmically such that

$$f_{i+1} = f_i \times 2^{1/12} \quad (2.5)$$

This means that in order to obtain enough frequency samples to resolve low pitches in the Fourier spectrum, there have to be many more samples than necessary at high frequencies. Also the frequency bins are not centered at note centre frequencies. The most common method of overcoming this problem in music information retrieval applications is to use an alternative frequency representation of the signal that better matches the note frequencies.

2.3.3 Constant-Q Transform

The constant-Q transform [Brown, 1991] is an alternative to the Fourier transform where the quality factor, or Q , of the frequency bins is kept constant. The Q value is given by

$$Q = \frac{\text{centre frequency}}{\text{bandwidth}} \quad (2.6)$$

This means that, with a suitable tuning of Q , the frequency bin centres can be made to correspond directly to the pitches of the equal-tempered scale. Although equal temperament is not used for most music, a frequency analysis designed for equal temperament is a far better approximation of the true note frequencies than one in which the frequency bins are linearly spaced. The relationship between consecutive frequencies for semitone resolution becomes

$$f_{i+1} = f_i \times 2^{1/12} \quad (2.7)$$

For semitone resolution, or 12 frequency bins per octave, the Q value is approximately 17. If greater frequency resolution is required to take account of discrepancies in the recorded pitch, more frequency bins per octave can be used, giving a lower bandwidth and higher Q .

Brown [1991] derives the following equation to calculate the constant-Q transform:

$$X_{cq}(k_{cq}) = \frac{1}{N(k_{cq})} \sum_{n=0}^{N(k_{cq})-1} W(n, k_{cq}) x(n) e^{-2\pi j Q n / N(k_{cq})}, \quad k_{cq} = 1, 2, \dots, \text{noct} \times \text{bpo} \quad (2.8)$$

where $N(k_{cq})$ is the number of time or frequency domain samples, now dependent on the frequency, k_{cq} , noct is the total number of octaves, bpo is the number of constant-Q bins per octave and W is a Hamming window.

The constant-Q transform is similar to the short time Fourier transform in its use of a window to taper the frame edges, but differs in the window lengths, which vary with frequency, such that the number of cycles in each window is equal to Q for all frequencies. The resulting signal representation is equivalent to the result of taking several short time Fourier transforms, each of a different window length corresponding to a constant-Q frequency, and retaining only the frequency values for frames that correspond to the time centre of the lowest frequency bins. This means that there is some loss of information in the transformation process, so there is no inverse transform. Figure 2.8 illustrates the division of the time-frequency plane by the Fourier transform and the constant-Q transform.

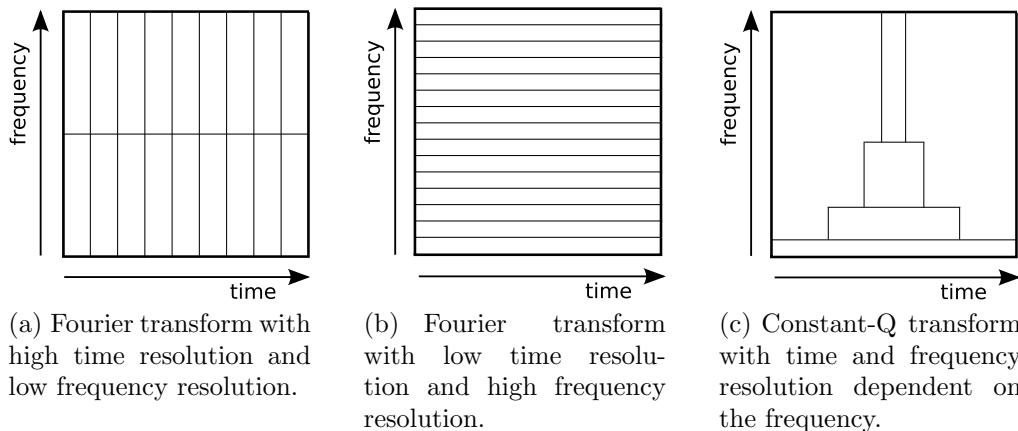


Figure 2.8: Illustration of the division of the time-frequency plane for two different Fourier transforms and the constant-Q transform.

Brown and Puckette [1992] develop an efficient algorithm that makes direct use of similarities with the Fourier transform in order to utilise fast Fourier transform algorithms.

For any two discrete signals $u(n)$ and $v(n)$,

$$\sum_{n=0}^{N-1} u(n)v^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} U(k)V^*(k) \quad (2.9)$$

where the discrete Fourier transforms of $u(n)$ and $v(n)$ are $U(k)$ and $V(k)$ respectively, and $V^*(k)$ is the complex conjugate of $V(k)$.

The constant-Q transform can therefore be calculated by taking a Fourier transform of the temporal kernels, $v^*(n, k_{cq})$, to give spectral kernels $V^*(k, k_{cq})$, and multiplying by the Fourier spectral coefficients, $U(k)$, of the input signal, $u(n)$.

By making use of fast Fourier transform efficiency the algorithm can easily run in real time. Additional computational savings can be made with minimal loss of accuracy using the fact that the spectral kernels are localised in frequency with values close to zero elsewhere. If values below a given threshold are set to zero, many multiplications need not be carried out because their result is zero. We investigate the effects on our key estimation model of varying the sparse kernel threshold, in chapter 5, section 5.1.4.

2.3.4 Chroma Vectors and Chromagrams

The constant-Q transform gives us a frequency representation of the audio signal in which the frequency bins can be tuned to the pitches of an equal-tempered scale. In order to obtain a substitute for a pitch class distribution it is usual to remove the pitch height information, by summing the constant-Q frequency bins that are spaced by a whole number of octaves, to obtain a *chroma vector*, or *pitch class profile* (PCP).

For constant-Q data with 12 bins per octave covering 5 octaves, there will be $12 \times 5 = 60$ frequency bins. Assuming the first bin is centred at 55 Hz (pitch A1), bins 1, 13, 25, 37 and 49, corresponding to frequencies of 55, 110, 220, 440 and 880 Hz, will be summed to give the final pitch class value for A. The same procedure for all 12 pitch classes gives a 12-element vector for each frame of data that represents the relative energy of each pitch class over all octaves, as given by equation 2.10.

$$\text{PCP}_i = \sum_{\text{oct}=1}^{\text{noct}} X_{\text{cq}}((\text{oct} - 1) * 12 + i), \quad i = 1, 2, \dots, \text{bpo} \quad (2.10)$$

where PCP_i is the i th pitch class profile value.

Equation 2.10 is valid if a whole number of octaves is analysed. Otherwise chroma values must be normalised by dividing each PCP value by the number of elements in the sum.

Each PCP represents the energy in each pitch class for one frame of the original audio. If PCPs are calculated for all frames and stored or displayed consecutively, the result is known as a *chromagram*.

The chromagram was first used by Fujishima [1999] for chord recognition purposes, and has since been adopted for the vast majority of music chord and key analysis applications, including the machine learning approaches.

2.3.5 Alternatives to the Constant-Q Transform

Some alternative methods for obtaining a logarithmic frequency representation have been proposed. Chuan and Chew [2005a] begin with a high frequency resolution FFT. Frequency boundaries for each note are defined, and the frequency band for each note will span many FFT bins; more for high notes than for low notes. The value of the highest peak in each frequency band is taken as the pitch strength value for that note. Octaves are then summed to give a chromagram.

They later modify the technique [Chuan and Chew, 2005b] using the presence of the first upper partial in the detected peaks to refine lower pitch estimates where the frequency resolution is poor. They also apply extra weighting to pitches that are sounded in high energy regions of the spectrum, and perform some thresholding of low and high values, before performing the final chromagram mapping.

Peeters [2006b] also begins with a high resolution Fourier transform, then applies a filterbank to merge the appropriate FFT bins into a logarithmic spectrum. He uses a final resolution of 3 bins per semitone, then discards all bins that are not centered on a pitch value. A chromagram is then calculated by summing the octaves.

Both of these approaches disregard parts of the signal which are thought to be unimportant for determining pitch: only the peaks are kept in Chuan and Chew's method [Chuan and Chew, 2005a], and only the centre third of a semitone band is kept in Peeters' method [Peeters, 2006b]. Peeters states that discarding this information reduces noise, and noise reduction is likely to be the incentive in Chuan and Chew's approach although it is not explicitly stated, and the effects have not been examined.

Pauws [2005] takes a simpler approach, using a cubic interpolation of a linear frequency spectrum to obtain values for logarithmically spaced bins. He then formulates the chromagram mapping procedure as a maximum likelihood estimation. The likelihood of each pitch class having occurred in a given frame is given by the ratio of the energy in that pitch class over all octaves, to the energy not in that pitch class over all octaves. This will enhance the peaks compared to a simple summation operation, so may help to reduce the effects of noise.

It is possible that different spectral weightings will result from these different approaches. Chuan and Chew pick a single value from a band of linear frequency bins, but for higher notes the signal energy is spread across more linear bins, so the final logarithmic representation will be weighted by an envelope that decreases with frequency. Pauws' approach will be subject to the same effect. Brown normalises the constant-Q frequency samples by the window length so should achieve a less weighted spectrum, but the Hamming windows used are symmetrical on a linear frequency axis so the logarithmic frequency bins are biased towards the lower end of the band. Peeters' approach is perhaps the most elegant, since it analyses the whole time signal

for all frequencies, and the filters used are symmetrical on a logarithmic frequency axis. These differences and their effects for music analysis applications have not been formally assessed.

Pickens et al. [2003] use a phase-vocoder technique [Serra, 1997] to estimate instantaneous frequencies, followed by two different methods of grouping time-frequency bins into notes, which both aim to produce a full polyphonic transcription. The phase vocoder can give very good frequency resolution, but a comparison with constant-Q or FFT features is not given. However, they do show that the performance of their Markov models in a music retrieval context is similar for the two different types of transcription, which indicates that it has a degree of robustness to the features used.

Martens et al. do not require a substitute for a pitch class distribution, their key model is based on auditory nerve patterns [Martens et al., 2002, 2004], so their feature vectors must also represent nerve patterns. The musical signal is decomposed using an ear model, which starts with a filterbank of 15 critical bands. Periodicities in the 15 channels are found using frame-wise autocorrelation to give pitch patterns, and the final feature is created by modelling pitch memory, adding a small amount of previous pitch patterns to the current one.

2.3.6 Disadvantages of the Chromagram

Although the chromagram is widely used for harmony analysis, it does have some disadvantages. The pitch height information that is disregarded could potentially contain useful information. It is easy to understand that for the purposes of chord recognition pitch height tells us which inversion the chord is in, and whether a pitch class one tone above the root is sounded as a 9th or between the root and third to give a discordant cluster. Perhaps there is equivalent information in the spread of notes that is useful for tonality estimation. Certainly it seems likely that the bass line is of the most importance. Such questions have not been extensively addressed in the context of audio analysis, although Mauch and Dixon do emphasise the importance of the bass by calculating a separate bass chromagram, and from it they produce a bass transcription [Mauch and Dixon, 2008].

Another common problem is that in many recordings the pitch varies from 440 Hz for A. Tuning reference frequencies vary throughout the world, and through history, and for early recordings tape speeds were not consistent, often deliberately. This would mean that the constant-Q frequency bins would no longer be at the centre of a pitch class, so some additional spectral leakage would occur. To account for the possibility of different tunings, various methods of tuning the chromagram have been proposed.

Tuning the Chromagram

Harte and Sandler [2005] begin their method of addressing the tuning problem with a higher-resolution constant-Q transform with 36 bins per octave, which is then mapped to a 36-bin chromagram. Quadratic interpolation is applied to the chromagram, and peaks in the interpolation are extracted to the nearest cent. To estimate the overall deviation $e \in \{-50, \dots, 49\}$ from standard tuning in cents, a histogram approach is taken, i.e. e is the centre of the histogram bin into which the most single peak tunings fall, where the histogram bin centres are spaced apart by 1 cent. Having found the pitch class centres, the pitch class boundaries can be easily calculated at 50 cents either side of the centre, and each energy peak can be assigned to the appropriate pitch class to give a tuned, 12-bin chromagram.

For our work on tonality estimation we use the same method of extracting peaks from a quadratic interpolation of the chromagram, but use an alternative method for determining the overall tuning deviation that is similar to that presented by Dressler and Streich [2007]. The deviation e' of each individual energy peak in cents is represented as a unit vector with angle $2\pi e'/100$. The unit vectors are summed, and the angle of the sum is mapped back onto the range of one semitone to give the overall deviation e .

Zhu et al. [2005, 2006] tune the spectrum using the same principle, but they start with a higher resolution constant-Q transform, with 10 bins per semitone, or 120 bins per octave, then do not interpolate. The highest energy peaks (subject to a threshold) for each time frame in the first 30 seconds of the track are found, then the frequency bins are summed modulo 10 to give the distribution of peaks across one semitone. The highest bin is recorded as the frequency for the current frame, and the values for each frame are accumulated into a histogram from which the most common tuning frequency for the first 30 seconds can be read.

Peeters [2006b] performs tuning frequency estimation in a high resolution linear frequency spectrum, where for a set of possible reference tuning values between 427 Hz and 452 Hz for pitch A, the amount of energy explained by each tuning value relative to the total signal energy is calculated. This is given by the ratio between the energy in the frequency bins that are centered on pitch classes for the current tuning, and the energy in all bins. The tuning value that explains most of the signal energy is assumed to be the correct tuning for the whole track, and the audio is resampled so that the following steps can operate as though A were set to 440 Hz in the recording.

2.3.7 Adaptations of Chroma Features for Symbolic Tone Profiles

In section 2.2.2 methods of adapting tone profiles for use with digital audio were discussed. There have also been some methods proposed for adapting the chroma features to better match symbolic tone profiles.

The most important difference between a chromagram derived from audio and a symbolic tone profile is the presence of upper partials of notes in the chromagram. These appear as peaks in the frequency spectrum at approximately integer multiples of the fundamental, and most of them are of a different pitch class to the fundamental, straying gradually more from pitches of the equal-tempered scale as the partial number increases. Although humans naturally group partials together to form the sound of a single instrument, in a computer-generated representation of the spectrum it is extremely difficult to identify whether a peak is a fundamental pitch or a partial of a lower fundamental. This difficulty is augmented when pitch height information is lost in a chromagram mapping procedure.

In order to reduce the influence of upper partials in the final chromagram, Peeters proposed a method to reduce their peaks [Peeters, 2006a]. A measure of how likely a frequency is to be a fundamental pitch is calculated first, based on how much of the signal energy is explained by the current frequency being a fundamental, minus a penalty for the current frequency being an upper partial. This fundamental likelihood score for each frequency is point-wise multiplied by the log amplitude spectrum, so spectrum values that are unlikely to be fundamental pitches, and hence likely to be upper partials, are reduced. The effects are not evaluated in detail but diagrams show that it is effective.

In another approach, for each frequency bin, Pauws [2005] adds a small amount of the energy at each upper partial frequency to the current frequency bin so that frequencies with strong components at partial frequencies are enhanced. The amount of each partial added decreases exponentially with the partial number.

Zhu et al. [2005, 2006] take a unique approach to modifying the frequency data to make it more suitable for harmony analysis. They first perform a constant-Q transform at 120 bins per octave, then discard all frequency bins that are not precisely in tune with the reference frequency (previously computed, see section 2.3.6). Since partials that do not form an interval of a whole number of octaves with the fundamental are not exactly in tune with equal temperament, this removes them from the processing, and the inclusion of octave-spaced partials only strengthens a pitch class that is already present. From this data a binary vector is created over the frequency spectrum that represents whether each frequency is to be part of the later processing. The highest energy frequency component is included, then the next highest is included only if it forms part of a diatonic scale with the first. Frequency components are added until the next highest peak does not form part of a diatonic scale with all previous components. This process is called consonance filtering, and serves to remove non-diatonic embellishments to the music, and wideband noise. The binary vector is then summed across all octaves to create a chromagram.

Another source of errors in chromagram representations is transient sounds, which are wide-

band signals that appear in the spectrum and can mask tonal peaks. Harmonic peak enhancement methods already described in this section will have some effect in reducing the contributions of wideband signals to the spectrum. In chapter 5, section 5.3.2 we investigate the effects of applying a specific transient removal algorithm [Duxbury et al., 2001] before the spectral transformation takes place.

2.4 Classification Methods

Having defined a key model and extracted appropriate features from the audio to match the model, the next step is to measure how the extracted features fit into the model and obtain a key estimate.

For the tone profile approaches this is a question of defining the most appropriate distance measure between the profile for each key and the chroma vector. The distance measure suggested by Krumhansl [1990, p. 37] is the correlation, where a high correlation corresponds to a small distance. The correlation coefficient r_{xy} between key profile \mathbf{x} and input vector \mathbf{y} is given by

$$r_{xy} = \frac{\sum_{n=0}^{11} (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_{n=0}^{11} (x_n - \bar{x})^2 \sum_{n=0}^{11} (y_n - \bar{y})^2}} \quad (2.11)$$

where \bar{x} represents the mean of \mathbf{x} . This has become the most popular measure to use, implemented by the top 5 entries of the MIREX 2005 audio key finding contest [MIREX competition, 2005a].

Other distance measures used in published key and chord finding algorithms include the square of the Euclidian distance, D_{Euc}^2 [Fujishima, 1999, p. 54], the inner product, D_{inner} [Temperley, 2001, p. 176], the Kullback-Leibler divergence, $D_{\text{KL}}(\mathbf{x} \parallel \mathbf{y})$ (used in a different context by Pickens et al. [2003]), and a fuzzy distance, D_{fuzz} [Purwins et al., 2000] which is similar to the Euclidian distance but the distance is reduced as the data becomes less certain (as the variance increases).

These measures are given by

$$D_{\text{Euc}}^2 = \sum_{n=0}^{11} (x_n - y_n)^2 \quad (2.12)$$

$$D_{\text{inner}} = \sum_{n=0}^{11} x_n y_n \quad (2.13)$$

$$D_{\text{KL}}(\mathbf{x} \parallel \mathbf{y}) = \sum_{n=0}^{11} x_n \log \frac{x_n}{y_n} \quad (2.14)$$

$$D_{\text{fuzz}} = |\mathbf{x} - \mathbf{y}| \times \left(1 - \frac{\sigma}{|\mathbf{x} - \mathbf{y}| + \sigma} e^{-\frac{|\mathbf{x} - \mathbf{y}|^2}{2\sigma^2}} \right) \quad (2.15)$$

where σ^2 is the variance of \mathbf{y} .

It is not clear which of these measures is best suited to the task, and there are also several alternative distance measures that have not been used in published algorithms, including the cityblock distance, mutual information, etc. We have conducted informal studies that suggest

that the distance measure does not significantly affect the results, so any one would be appropriate, but more rigorous testing would be required to confirm this.

Chew's spiral array method [Chew, 2001] and the toroidal model of Harte et al. [Harte et al., 2006] can be viewed as a kind of template approach, where the distance measure is the Euclidian distance between the template and the input vector within the space of the spiral or toroid. The dimensions of the space are derived from music theory principles, so these distance measures are perhaps intuitively the most appropriate, but formal experiments have not been conducted to prove this. The most appropriate distance measure should be derived from the most appropriate tonal space, but there is no space that is clearly the most appropriate.

The preference rule approaches do not require any kind of distance measure, since the key classification is performed simply by applying the rules, and no further processing is required. The quality of the classification is of course dependent on how well the rules capture tonal information.

Machine learning methods tend to have a standard classification method that is part of the learning algorithm. Those based on Bayesian statistics have a single method for finding the most likely model parameters given the observed features, for example the Viterbi algorithm uses dynamic programming to find the most likely complete path through the states of a hidden Markov model (see section 3.2.3 in chapter 3). For support vector machines, neural networks and classification trees the classification method is ingrained in the training procedure, defined by the classification structure chosen at the start. Although the classification step is clearly defined for these methods, the training procedure usually involves some form of error minimisation, and the metric used for error measurement could be one of a large number of distance measures in the feature space. Machine learners are therefore still subject to the need for useful tonal features and an appropriate distance measure between them, but their ability to adapt to the data should mean that they are relatively insensitive to such choices.

2.5 The Final Tonality Estimate

Some consideration should be devoted to the form of the final tonality estimate. The majority of methods described result in a single key estimate for the whole piece, but frame-based analysis allows for other possibilities. If a single key estimate is the goal, there might be sections of a piece that are more useful than others for finding that key. It is also possible to create a single key estimate for every frame, and so track key changes. Some approaches produce not only a single key estimate for every frame, but a measure of how likely every key is in that frame, so complicated analyses of the relative importance of all keys can be produced. We discuss these issues in the following sections.

2.5.1 Location in Time of the Main Tonality of a Piece

Many harmonically simple pieces remain in the same key throughout. However, it is likely that a piece of music will not remain focussed on a single tonal centre. It is usual for a piece to begin and end firmly in a single key, but explore other tonal areas in between, either by full modulations or by shorter term tonicisations (a distinction that is itself blurred).

To utilise this observation several key recognition approaches analyse only the beginning or end of the track. Turning once again to the MIREX audio and symbolic key finding contests [MIREX competition, 2005a,b], the contests are run using only the first 30 seconds of the track in order to simplify the problem, and most of the contest entrants use additional methods to bias the analysis towards the beginning of the excerpt.

Gómez entered two versions of her algorithm [Gómez, 2005], one of which averaged the pitch distribution across the whole 30 seconds, the other analysed only the first 15 seconds and performed only very slightly better (86.05 % correct compared to 85.90 %). Purwins and Blankertz also only analyse the first 15 seconds, and weight the frames by a quarter cosine function to give more importance to the earlier frames.

Zhu's rule-based approach [2005] progresses through the excerpt accumulating evidence for the different tonalities on a frame by frame basis and stops when enough evidence has been acquired. The precise details of the rules and stopping criteria are not given.

Chuan and Chew [2005c] and Mardirossian and Chew [2005] analyse a set number of seconds from the beginning of the excerpt determined by the length that gave the best results for the training data. The lengths used were 13.32 seconds for the audio and 28 seconds for the symbolic case.

The winning algorithm for symbolic key-finding was that of Temperley [2005], who tracks the tonality of the whole excerpt then selects the key of the first frame of his complete analysis.

İzmirli won the audio key-finding contest [İzmirli, 2005b]. He generates a key estimate for every audio segment together with a confidence measure, then the confidence measures for each key are summed for the whole excerpt to give the key in which there is most confidence. This analyses the whole excerpt, but each segment is defined from the beginning of the track, so the first notes are included in every segment but the last notes are only analysed by the last segment, which consists of the whole excerpt. This has the effect of biassing the analysis towards the beginning of the excerpt.

The other two algorithms, by Pauws [2005] and Rizo and Iñesta [2005], assess the whole excerpt with no bias towards any particular part.

It is not possible to directly compare the effects of the different weightings applied since there are many other significant differences in the algorithms, but in general it seems that the beginning

of the excerpt is favoured.

Given that the contest musical examples were all cropped to include only the first 30 seconds, none of the submitted algorithms made use of the fact that pieces tend to end in the main key, since the ends of the examples are not the ends of the pieces. However, it is also more common in general tonality estimation algorithms that are not related to the MIREX contest to focus the analysis on the start of the piece rather than the end, for example in the Longuet-Higgins algorithm [Temperley, 2001, p. 169], highlighted by the use of opening segments of music as test data [Krumhansl, 1990, p. 81] [Chew, 2001] [Gómez Gutiérrez, 2006, p. 109].

Gómez [2006, p. 134–5] presents an investigation into the best part of the track for global key finding using a tone profile correlation approach. The results show that an analysis of the whole track provides the best estimation, but the result from the first 10 or last 25 seconds can be a good approximation. This experiment was conducted using a collection of classical pieces of various styles. Her findings are supported by the study by Pauws [2004], where the most accurate results are obtained by analysing the whole piece, but he suggests that the first or last 25 seconds can give a reasonable approximation.

2.5.2 Hierarchical aspects of Harmony

The focus of the discussion so far has been related to extracting a single global key for a whole musical excerpt. This is useful for music in which there is a constant key, or for short sections of music in which locally there is a constant key, but to describe a whole piece by a single tonal centre and mode is a very simplified view of music. We wish to account for the possibility that the tonality could be continuously shifting as a piece unfolds.

Perhaps a more appropriate model of tonality would be a hierarchical one where at the lowest level the chords are observed, with each chord estimate occupying a short time, for example one beat. The next level up would begin to analyse the tonality of short term chord progressions, over for example one bar. A slightly higher level would deal with short term tonicisations, a higher level still would analyse the long term modulation structure, with the highest level of all representing the main key for the whole track.

Some of the tonal models described in section 2.2 have been used to track modulations within a piece. The most commonly used method is to analyse only a short section of the music at a time, so a progression of key estimates is produced. This approach is used by many, including Gómez [2006, p. 141] and Chew [2006]. Such an approach raises the question of how long the segments should be. The length of the ideal segment for defining the current key is dependent on the tempo and musical style, and also on the intended meaning of the word “current”. Zhu and Kankanhalli [2004] use a rule-based approach to finding modulations in MIDI, but they make extremely limiting assumptions about the modulations that are allowed. The approach is

designed to follow a transposition of one or two semitones that often occurs for the final chorus of a pop song.

The approaches based on probability [Peeters, 2006b, Chai and Vercoe, 2005, Temperley, 2004] are also dependent on the frame lengths, since the likelihood of changing key from one frame to the next will depend on the time between one frame and the next, and on the particular music used. Even if the scope implied by the word “current” were defined, it is not clear how to set the most appropriate resolution for a varied music collection.

Bello and Pickens [2005] show that using frame lengths based on the tempo extracted by an automatic beat-tracking algorithm [Davies and Plumbley, 2007a] improves chord recognition results using an HMM by roughly 10 %. Shenoy et al. [2004] also use beat-length frames, but no comparison is made with a system using equal-length frames. We conduct an experiment to assess the effects of using beat-length frames on our key estimation method in chapter 5.

Shenoy et al. [2004] employ a two-stage system for main key estimation that begins to model some harmonic hierarchy, starting first with a template-matching chord recognition algorithm, then comparing the estimated chords to a heuristically-defined template of the chords belonging to each key. This structure bears some similarity to our work, although we also make use of a hidden Markov model rather than a simple template to relate the chords to the key.

Gómez [2006, p. 146] shows that it is possible to perform a multiresolution analysis with many layers of segments ranging from less than a beat in length to the length of the entire piece. This gives an estimate of tonal changes at many different timescales. Relative strengths of different keys can also be visualised by a point moving inside a geometrical tonal space [Chew and cois, 2005], or by a planar version of a tonal space with colour representing the key strength at each point [Toivainen, 2005].

The tree-based global key estimation algorithm of Martens et al. [2004] naturally incorporates a hierarchical structure, but this has not yet been exploited for a hierarchical tonality estimation.

2.5.3 Harmonic Progression Through Time

The concept of harmonic progression is of great importance for analysing tonality. Whilst the specific pitches played can tell us a lot about the likely tonality for some musical excerpts, for others this information is insufficient for an accurate tonality judgement: Gómez [2006, Appendix B] reports results of between 13 and 88 % key accuracy for various profile types and five different sets of test music using a tone profile correlation approach. Clearly there must be additional information in the music that helps a human to analyse the tonality.

Krumhansl [2005], Gómez [2004b] and Pauws [2004] all state that ignoring pitch order is a weakness of the tone profile approach. Some scale degrees are more important than others for defining tonality, based on their relationship to each other, to the melody, to the meter and to

the phrase structure. For example, a chord G on the final beat of the bar followed by a chord C on the first beat of the next bar strongly implies C major. However, if the C chord is sounded on the final beat of the bar followed by a G chord on the downbeat, the tonality is more ambiguous, and is likely to be interpreted as G major. The note distribution would be the same in both cases.

Toiviainen and Krumhansl [2003] develop a model of tonality based on first order tone transitions (pairs of tones), with tone transition strengths determined using an algorithm that was developed empirically, guided by previous research into pitch memory and perception of interval sizes. These pitch transition strengths are then correlated with tone relatedness ratings previously reported by Krumhansl [1990, p. 125] to give a key estimate.

The progression of chords has been taken into account to some degree by the approaches to chord recognition that use HMMs [Sheh and Ellis, 2003, Bello and Pickens, 2005, Lee and Slaney, 2006], for which each state is dependent on the previous state, thus incorporating some sense of progression in the model. The model probabilities can be set so that the transition from G to C has a different likelihood from the transition from C to G, and the transition likelihoods can be different for each key. Chai [2005] is one of the first to use HMMs for key recognition purposes, using one two-state model where each state represents a mode, major or minor, and a second model in which the states represent key notes. Peeters' approach [2006b] more closely follows speech recognition methods in its use of a separate HMM for each key, where the meaning of the hidden states does not correspond to any particular musical feature.

HMMs present a useful framework for beginning to improve on pure pitch distribution methods. However, they are limited in their range since the state likelihoods are dependent only on the previous state, so long term harmonic structures cannot be modelled. In addition, the probability of remaining in the same state must always decrease exponentially with time if a standard HMM is used, which is likely to not be representative of the durations of tonal regions. It is possible to define Bayesian models [Murphy, 2002] in which the state duration probabilities can be controlled, which is a possible direction for future research. Pickens et al. [2003] showed that for their harmonic model, first and second order Markov chains produced better retrieval performance than zeroth order Markov chains.

Papadopoulos and Peeters [2008] have shown that improved chord recognition performance can be achieved by modelling chord transitions as a function of metrical structure, using the assumption that chords are most likely to change on the first beat of the bar. It seems likely that the next generation of tonality estimation research will concentrate on models that combine the interrelated musical properties of key, chord and meter, after the rule-based approach of Shenoy and Wang [2005].

2.6 Methods of Evaluation

We now turn our attention to the question of how to evaluate tonality estimation algorithms. Evaluation is an essential aspect of algorithm development, and a numerical measure of performance is desirable for the comparison of different algorithms.

The simplest method of evaluating the accuracy of a tonality estimation algorithm is to use music extracts which have the main key as part of the title, such as “Minuet in G”, then calculate the percentage of the test extracts for which this main key is correctly found by the algorithm. This is the approach taken by the vast majority of published methods. The keys taken from titles are assigned by the composer, so must be correct. Alternatively, it is a tractable task for a musically trained individual to manually annotate the main key of a set of musical extracts, since for most cases there is no question over what this key should be, so a test set can be created from pieces that the composer has not annotated with the key. In this work extensive use is made of the songs of the Beatles, which have been analysed by the musicologist Alan Pollack [2000].

The MIREX key finding contests [MIREX competition, 2005a] modified the simple percentage scores to give some credit for estimated keys that are not correct but are closely related to the correct key. The relative scores are shown in table 2.1. This has the effect of improving the scores for an algorithm that puts pieces in the correct tonal region but is not necessarily accurate to the precise key. It is intuitively a measure more closely related to both music theory and perception than one that gives a zero score to both the dominant key and the key furthest away on the circle of fifths. The precise values were arrived at heuristically, but the principles on which they were set are grounded in musical knowledge.

Table 2.1: Relative scores for different keys used in the MIREX 2005 key finding contests.

Key	Score
Correct (tonic)	1
Dominant/subdominant	0.5
Relative major/minor	0.3
Parallel major/minor	0.2

Disadvantages of Global Key Measures

As described in section 2.5.2, it is common for a piece of music to modulate from one key to another, which means that a single key estimate for the whole piece is inadequate to describe its harmonic path. Some efforts have been made to model finer details of tonal progression, but aiming for an analysis of this complexity instantly makes not only the problem of tonal analysis more difficult, but also the evaluation of the results. Whilst the main key of the music is often stated in the title or can be easily annotated by a musician, detailed hierarchical analyses are

much harder to find. There are also often disagreements between musicologists, or agreements that a particular section is tonally ambiguous. A good analysis of tonality should be able to detect and represent these ambiguities, and hence a good evaluation method must also account for them. Vos gives a discussion of some of these problems [Vos, 2000], and we discuss them in detail in chapter 6.

To date evaluations of this type of complex automatic analysis have focussed on visualisations, such as those of the multiresolution key finding of Gómez [2006, p. 146]. The results can be displayed as a two-dimensional colour plot to give an instant visual overview, which she calls a keyscape, of the tonality changes in the piece. This kind of plot is inspired by the work on tonality visualisation of symbolic music data by Craig Sapp [2001, 2005], and is appropriate for displaying the hierarchical nature of tonality, but a method has not yet been proposed for quantitatively evaluating its accuracy. Gómez and Bonada have also developed visualisations that change as a song is played, using colour to represent the value of a key strength function over a toroidal model of tonal relationships [Gómez and Bonada, 2005].

Krumhansl [1990, p. 96–110] makes comparisons of her algorithm’s estimates against detailed analyses of a single piece as annotated by musicologists. These are fruitful studies but are not extendable to large scale testing due to the time required to manually annotate the music. Temperley [2007, p. 90] uses annotations in a music theory textbook to evaluate various key-finding methods for symbolic music, but highlights the shortage of data as a problem which has meant that he had to test the algorithms on the same data as his profiles were derived from.

Pickens et al. [2003] evaluate their harmonic model in terms of its retrieval performance. Their aim is not to perform accurate chord or key estimation, but to create a suitable harmonic model for retrieval purposes. We argue in chapter 6 that retrieval based measures should be used to evaluate all methods of tonality estimation for which accurate retrieval is the ultimate goal.

For the majority of this thesis we give percentage and MIREX scores for estimates of a single global key for each piece in our music collections, since these scores are simple to calculate, easy to understand, and allow for comparison with previous methods. However, we return to the question of evaluation in chapter 6, addressing the need for a method of evaluating a complete tonal analysis that contains much more detail than a single key estimate.

2.7 Summary

In this chapter we have discussed previous approaches to computational tonality estimation from audio data, which can all be represented in terms of the general diagram shown in figure 2.1. The methods are grounded in knowledge derived from studies of perception and music theory, the most important concepts being the separability of pitches into pitch class and pitch height,

and the idea that some tonalities are judged to be closer together than others. For all of the experiments described in the following chapters we use only pitch class information for tonality estimation.

The most important differences between approaches lie in the structure of the key model. An important class of methods is the tone profile correlation approach, in which a key is defined by its pitch class distribution. Geometric models that treat a key as a point within a space that represents the perceived relationships between pitch classes have also been used. These are essentially tone profile methods that use an alternative distance measure to the correlation. Methods that use a series of preference rules have been proposed, but these are very restrictive and are not able to produce detailed tonality judgements. Another class of model is the machine learners, which are able to adapt key classification boundaries to fit the music. Of particular importance are the methods using hidden Markov models (HMMs), which allow for relationships between keys, chords and pitches to be encoded, and add first order modelling of harmonic progressions through time. In chapter 3 we introduce an HMM to model relationships between chords and keys, and continue to develop it in chapters 4 and 5.

When working with audio data it is necessary to transform the data into a space in which it can be compared to the key model. We have discussed the approaches used to perform this feature extraction, the most commonly used method being a constant-Q frequency transform followed by mapping onto a chromagram. The chromagram is often used in place of a pitch class distribution, so methods of addressing the additional upper partials that are present in audio have also been described. In chapter 4 we calculate a chromagram as our first step towards estimating the musical key from audio, and use an exponential decay model of upper partials in the following step, chord estimation.

We have discussed methods of comparing the extracted features to the key model, and continued by considering the format of the final tonality estimation. Important considerations are the timescale over which estimates are made, and expression of the hierarchical nature of harmony. A multiresolution analysis is considered to be the most complete solution. Our model is capable of producing both a single main key estimation, and an estimation of how the likelihoods of all major and minor keys change through the piece.

Finally we discussed evaluation methods, which are essential for algorithm development. The most common evaluation metric is a simple percentage score for the number of pieces for which a single main key is correctly found. However, a complex tonal analysis cannot be effectively evaluated using this simple metric. To date, more complex automatic analyses of tonality have only been evaluated using either a very small amount of labelled test data, or using visualisations which do not produce a numerical output. For the experiments described in chapters 3 to 5 we

use simple percentage scores, and MIREX scores which give some credit for incorrect estimates that are closely related to the correct key. In chapter 6 we investigate more advanced evaluation metrics.

Chapter 3

Using a Hidden Markov Model for Tonality Estimation from Chord Symbols

The most commonly used methods of tonality estimation from audio data are key template matching approaches based on pitch class profiles. Such methods have been favoured due to their simplicity and efficiency of implementation. Various improvements have been suggested, including different distance measures (see section 2.4), methods of deciding which is the most important part of the music to analyse (section 2.5.1), methods of addressing upper partials present in the signal (sections 2.2.2 and 2.3.7), and estimating the tonality in a sliding window of various lengths to give a multiresolution analysis (section 2.6). However, there are some problems intrinsic to tone profile techniques that limits their application for tonality estimation. The distribution of pitch classes can give important clues about the tonality and has been proved to be useful in computer algorithms, but by no means contains all of the relevant information. In this chapter we introduce the use of a hidden Markov model (HMM) for tonality estimation, which addresses two additional factors that are important:

1. Musical context. Surrounding harmonies can alter the meaning of a set of notes, so it is important to take into account the ordering of musical events in time.
2. Variation between styles. Different styles of music will use the pitches in different ways, and even a single composer does not always write in the same style, so some degree of adaptability is required of a generic tonality estimation technique.

We introduce our particular HMM representation of the relationships between chords and keys in the symbolic domain, which allows us to assess the model without the problems associated with audio data. The model is evaluated with various parameter settings, using hand-annotated chord labels as observations and a global key recognition measure for optimisation. We summarise our findings and note the parameters that most affect the key recognition performance. In chapter 4

we extend the work to include audio input.

3.1 Hidden Markov Models

A hidden Markov model (HMM) is a formal probabilistic framework that interprets a given sequence of observed events as drawn from an underlying, hidden sequence of states with a certain probability. It is a generative model, meaning that the observation probability distribution allows the probabilities of an observation sequence given a certain state sequence to be calculated, and using Bayes' rule [Duda et al., 2001, p. 615] it is also possible to infer the most likely state sequence given a sequence of observations. The number of hidden states and the observation space must be defined in advance, together with a set of initial probabilities for the first state, the state transitions and the observations given the state. These initial probabilities can be adapted to match a dataset using a form of expectation-maximisation (EM) [Duda et al., 2001, p. 124], before the inference is performed. The mathematical principles of hidden Markov models, together with a description of standard techniques for training and decoding, have been presented extensively [Rabiner, 1989] [Cox, 1990] [Duda et al., 2001, p. 128–139]. In the following sections we introduce our particular HMM for tonality estimation and describe how we use the standard HMM techniques to find a key sequence given an observed chord sequence.

3.2 Model of Tonal Space

The notion that a chord sequence can strongly imply an underlying key, or allude to more than one key, fits well into the HMM structure. We define a discrete HMM in which the hidden states represent keys, and the observed data are chords [Noland and Sandler, 2006]. The most likely sequence of underlying states (keys) and the likelihood of each state at each time frame (relative importance of all keys over time) can therefore be inferred from the observable data (chords). Figure 3.1 shows a simplified diagram of the model. Each state represents a key, and the model is fully connected so that transitions are permitted from any key to any other key, or the key can stay the same. At each time step the key is modelled as emitting one of the possible observable chords, for example C major or A minor.

There are $N = 24$ possible major and minor keys and $M = 49$ different chords included in the model. The chords can be any major, minor, augmented or diminished triad, or *no chord*, which occurs during silence or entirely percussive sections. More complex chords are not explicitly modelled, since chords that are not based on triads are very rare in the Western music repertoire for which this analysis is intended, and it was considered that the sequence of underlying triads, excluding extensions, is sufficient to define the key. All inversions of the same chord are treated identically, which means that the choice of bass note does not affect the estimated key.

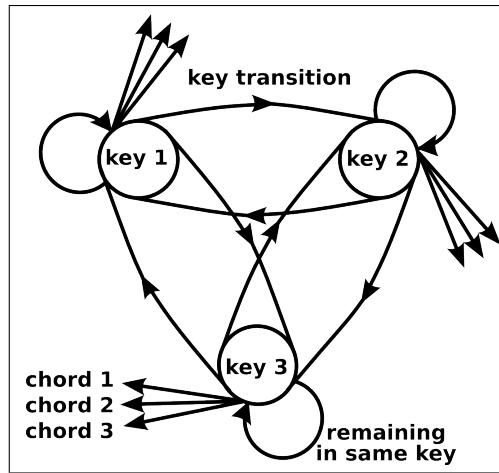


Figure 3.1: Simplified diagram of the tonality model. Only 3 keys and 3 chords are shown for clarity, but the 24 possible major and minor keys and 49 different chords are included in the actual calculations.

3.2.1 Initialisation

The model is completely defined by three sets of parameters, which all must be initialised:

1. The transition matrix $\mathbf{A} = (a_{ij})$, of size $N \times N$, in which element a_{ij} is the probability of transition from the i th to the j th key.
2. The emission matrix $\mathbf{B} = b_j(k)$, of size $N \times M$, in which element $b_j(k)$ is the probability of chord k occurring in key j .
3. The initial state probability vector, $\pi = (\pi_1, \dots, \pi_N)$, in which element π_i is the probability of starting in the i th key.

The rows of matrices \mathbf{A} and \mathbf{B} and the vector π describe discrete probability distributions, and must therefore sum to unity.

Perceptual Experiments used for Initialision

The EM algorithm used for adaptation is sensitive to the initial parameters, so we wish to define them as closely as possible to a model of perceived harmonic relationships. To this end we make use of the results of perceptual experiments carried out by Carol Krumhansl and colleagues [Krumhansl, 1990] to give numerical values to our intuitions regarding the relationships between the observations (chords) and hidden states (keys). For our first version of the model we make use of the following three sets of results.

Probe tone profiles Ten listeners were asked to judge how well probe tones corresponding to each pitch class fit within a given major or minor key context, on a scale of 1 (very good) to 7 (very bad). Average ratings were calculated to represent the importance of each pitch class within a major or minor key context, producing a probe tone profile that can be circularly

shifted to create a profile for each key. The ratings are given in table 3.1 and figure 3.2 for the C major and C minor key contexts. These are the same probe tone profiles as used in Krumhansl's symbolic key finding algorithm [Krumhansl, 1990, p. 77–110], and shown in figure 2.3(b) on page 25 of this thesis.

Correlations between key profiles The correlation between every pair of key profiles was calculated using equation 2.11 on page 47, to give a measure of how closely each pair of keys is related. The results are given in table 3.2 and figure 3.3.

Single chord ratings For the single chord rating test, listeners were asked to judge how well single chords fit within a key, using the same scale of 1 to 7. This test included all major, minor and diminished triads in both major and minor keys, and the average ratings are shown in table 3.3 and figure 3.4.

Initial state probabilities

The initial state probabilities reflect any prior information that we may have about the most likely key, before any of the music has been heard. It is plausible that in Western music certain keys occur more often than others due to composer preferences or ease of playing on particular instruments, and the initial state probabilities could be set to reflect this distribution. However, although using a non-equal key initialisation may increase the number of pieces that are correctly classified, it would decrease the likelihood of a piece that is in an unusual key being correctly classified, and we would like the model to work equally well for all keys. Hence we set all initial key probabilities equally, to 1/24.

State Transition Probabilities

The initial transition matrix should express how likely it is that when in a particular key, the music moves to another key at the next time step. Intuitively it is most likely that the music will stay in the same key, and if it does change key it is most likely to move to one that is closely related. The initial key transition matrix was created using the key profile correlations in table 3.2.

The values were circularly shifted to give the transition probabilities for keys other than C major and C minor; an operation that assumes G is to G major as C is to C major, etc. Let $\mathbf{P} = (p_{ij})$ be the 24×24 matrix containing correlations between the 24 major and minor keys. Then our initialisation for the transition matrix $\mathbf{A} = (a_{ij})$ is given by

$$a_{ij} = \frac{p_{ij} + 1}{\sum_{r=1}^{24} (p_{ir} + 1)}, \quad i = 1, 2, \dots, 24, j = 1, 2, \dots, 24. \quad (3.1)$$

Table 3.1: Krumhansl's probe tone ratings for C major and C minor key contexts [Krumhansl, 1990, p. 30].

Probe Tone	Context	
	C Major	C Minor
C	6.35	6.33
C♯/D♭	2.23	2.68
D	3.48	3.52
D♯/E♭	2.33	5.38
E	4.38	2.60
F	4.09	3.53
F♯/G♭	2.52	2.54
G	5.19	4.75
G♯/A♭	2.39	3.98
A	3.66	2.69
A♯/B♭	2.29	3.34
B	2.88	3.17

Table 3.2: Krumhansl's correlations between key profiles [Krumhansl, 1990, p. 38].

Key 1	Key 2	
	C Major	C Minor
C major	1.000	0.511
C minor	0.511	1.000
C♯/D♭ major	-0.500	-0.158
C♯/D♭ minor	-0.298	-0.394
D major	0.040	-0.402
D minor	0.237	-0.160
D♯/E♭ major	-0.105	0.651
D♯/E♭ minor	-0.654	0.055
E major	-0.185	-0.508
E minor	0.536	-0.003
F major	0.591	0.241
F minor	0.215	0.339
F♯/G♭ major	-0.683	-0.369
F♯/G♭ minor	-0.369	-0.673
G major	0.591	0.215
G minor	0.241	0.339
G♯/A♭ major	-0.185	0.536
G♯/A♭ minor	-0.508	-0.003
A major	-0.105	-0.654
A minor	0.651	0.055
A♯/B♭ major	0.040	0.237
A♯/B♭ minor	-0.402	-0.160
B major	-0.500	-0.298
B minor	-0.158	-0.394

Table 3.3: Krumhansl's harmonic hierarchy ratings for major, minor and diminished chords [Krumhansl, 1990, p. 171].

Chord	Key Context	
	C Major	C Minor
C Major	6.66	5.30
C♯/D♭ Major	4.71	4.11
D Major	4.60	3.83
D♯/E♭ Major	4.31	4.14
E Major	4.64	3.99
F Major	5.59	4.41
F♯/G♭ Major	4.36	3.92
G Major	5.33	4.38
G♯/A♭ Major	5.01	4.45
A Major	4.64	3.69
A♯/B♭ Major	4.73	4.22
B Major	4.67	3.85
C Minor	3.75	5.90
C♯/D♭ Minor	2.59	3.08
D Minor	3.12	3.25
D♯/E♭ Minor	2.18	3.50
E Minor	2.76	3.33
F Minor	3.19	4.60
F♯/G♭ Minor	2.13	2.98
G Minor	2.68	3.48
G♯/A♭ Minor	2.61	3.53
A Minor	3.62	3.78
A♯/B♭ Minor	2.56	3.13
B Minor	2.76	3.14
C Dim	3.27	3.93
C♯/D♭ Dim	2.70	2.84
D Dim	2.59	3.43
D♯/E♭ Dim	2.79	3.42
E Dim	2.64	3.51
F Dim	2.54	3.41
F♯/G♭ Dim	3.25	3.91
G Dim	2.58	3.16
G♯/A♭ Dim	2.36	3.17
A Dim	3.35	4.10
A♯/B♭ Dim	2.38	3.10
B Dim	2.64	3.18

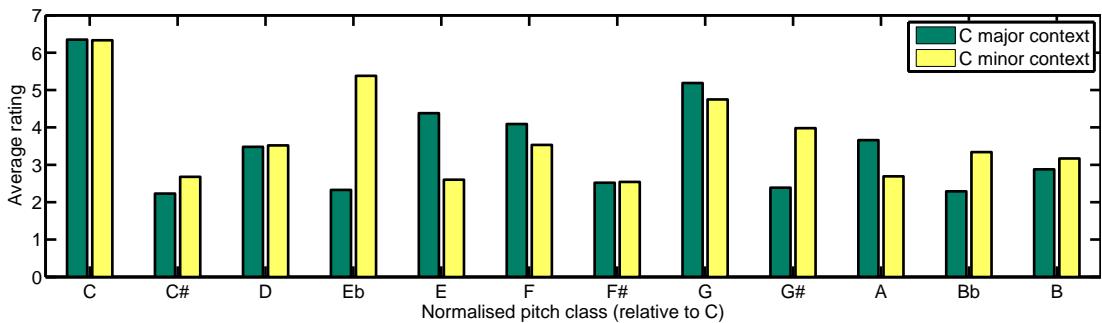


Figure 3.2: Krumhansl's probe tone ratings for C major and C minor key contexts [Krumhansl, 1990, p. 30].

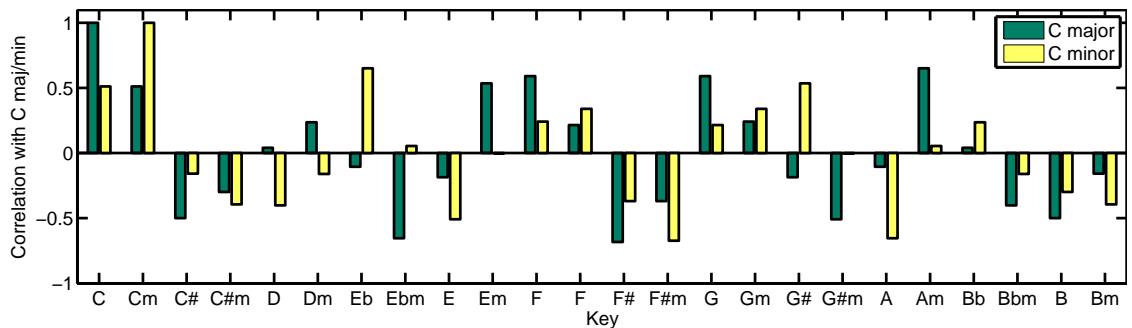


Figure 3.3: Krumhansl's correlations between key profiles [Krumhansl, 1990, p. 38].

Equation 3.1 shows that we added 1 to all the correlation values, which forced them to be positive, then they were normalised to sum to unity for each key so that they can be treated as probabilities. The final 24×24 transition matrix is shown in figure 3.5.

Observation Probabilities

The initial observation probabilities, stored in the initial emission matrix, should reflect the human expectation of the chords that are likely to be played when in a particular key. For example, the chord E major is very likely when the music is in the key of E major, since it is the tonic triad, so the probability of state E major emitting chord E major will be very high. However, E major is also the dominant chord of key A major, so the probability of state A major emitting chord E

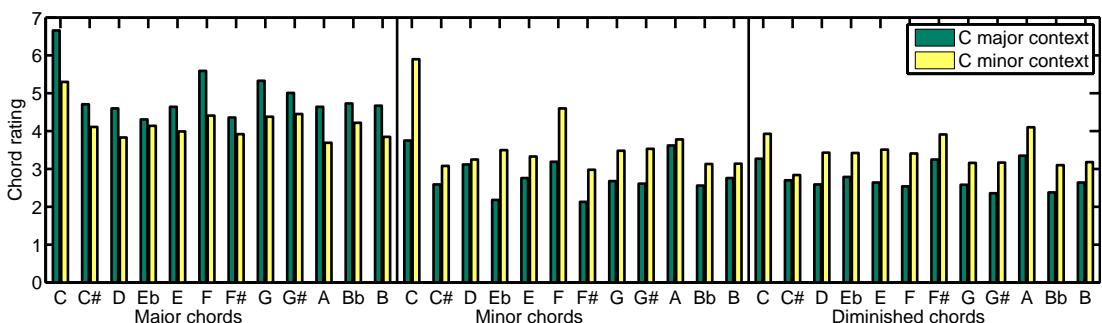


Figure 3.4: Krumhansl's harmonic hierarchy ratings for major, minor and diminished chords [Krumhansl, 1990, p. 171].

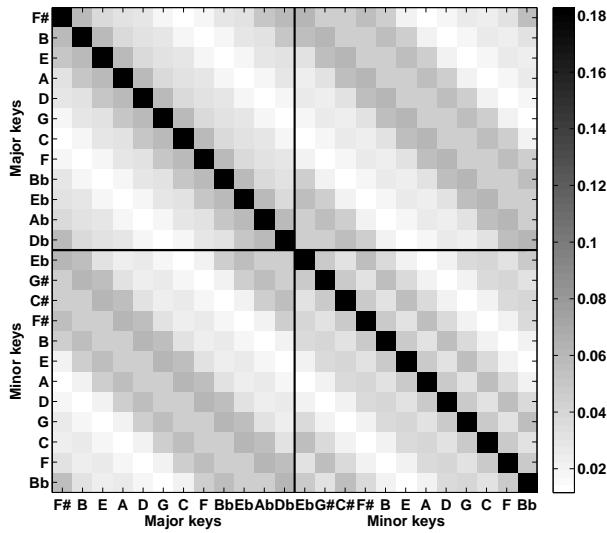


Figure 3.5: Visualisation of the initial transition matrix, with keys rearranged according to the circle of fifths.

major will be almost as high. Chord E major is not diatonic to the key of B flat major so the probability of state B flat major emitting chord E major will be very low.

The ratings for chords within a key, given in table 3.3, were used to provide numerical values for the emission matrix. These only cover the major, minor and diminished triads, but the model includes all major, minor, augmented and diminished triads as well as the possibility of there being no chord, so some additional numerical values were required. To allow us to compare this model to a version that uses chord transitions as observations (see section 3.3.2), for which ratings for diminished chords are not given, we only used the ratings for major and minor chords. The rating value for the minor triad on the seventh degree was also used for the diminished triad on the seventh degree, and values for all other augmented and diminished triads and *no chord* were set uniformly low, to 1, corresponding to “fits poorly” on Krumhansl’s seven-point rating scale. The values were circularly shifted to give the probabilities for other keys, with the exception of the *no chord* values since *no chord* has the same function in every key. Let $\mathbf{C} = (c_{jk})$ be a 24×49 matrix that contains the 49 chord ratings for each key, i.e. c_{jk} is the rating of the k th chord in the j th key. Then our initial observation probabilities are

$$b_j(k) = \frac{c_{jk}}{\sum_{r=1}^{49} c_{jr}}, \quad j = 1, 2, \dots, 24, \quad k = 1, 2, \dots, 49 \quad (3.2)$$

This normalisation operation allows us to treat the ratings as probabilities. Figure 3.6 shows the initial emission matrix.

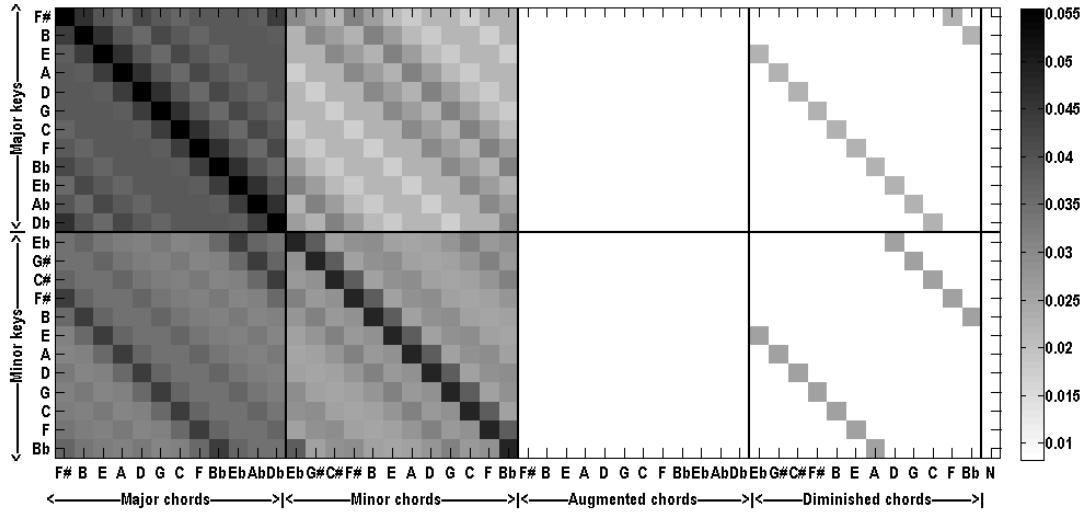


Figure 3.6: Visualisation of the initial emission matrix.

3.2.2 Adaptation of Model Probabilities

We use Baum-Welch training [Rabiner, 1989] in order to adapt the model parameters to an observed chord sequence, maximising the likelihood of the chord sequence given the model. This is a form of the EM algorithm, where first the expectations of each key transition and each key-chord pair over time is calculated given the training data (observed chords) and the current model. The expectations are then normalised and used as the next estimates for the state transition probabilities a_{ij} , and the observation probabilities, $b_j(k)$, respectively.

The following description of the Baum-Welch training procedure was adapted from Rabiner's tutorial [Rabiner, 1989], using the same notation. We first calculate the forward and backward variables, $\alpha_t(i)$ and $\beta_t(i)$, defined respectively as the probability of the partial chord sequence O_1, O_2, \dots, O_t and being in key i at time t given the model λ , and the probability of the partial chord sequence $O_{t+1}, O_{t+2}, \dots, O_T$ given key i at time t and the model. $N = 24$ is the total number of keys.

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (3.3)$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N \quad (3.4)$$

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (3.5)$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N \quad (3.6)$$

Using the forwards and backwards variables together it is possible to calculate the probability,

$\gamma_t(i)$, of being in key i at time t , given the whole observed chord sequence and the model. $\gamma_t(i)$ can also be interpreted as the probability of making a transition from key i to any key, since a transition must occur at every time step.

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (3.7)$$

In order to update the transition probabilities we define $\xi_t(i, j)$, the probability of being in key i at time t and key j at time $t + 1$, given the complete observation sequence and the model. $\gamma_t(i)$ is simply the sum of $\xi_t(i, j)$ over j .

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \quad (3.8)$$

We are now in a position to calculate updated values, \hat{a}_{ij} and $\hat{b}_j(k)$, for the transition and observation probabilities a_{ij} and $b_j(k)$. This is the maximisation step of the EM algorithm.

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from key } i \text{ to key } j}{\text{expected number of transitions from key } i \text{ to any key}}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (3.9)$$

$$\hat{b}_j(k) = \frac{\text{expected number of times in key } j \text{ and observing chord } v_k}{\text{expected number of times in key } j}$$

$$\hat{b}_j(k) = \frac{\sum_{t:O_t=v_k} \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (3.10)$$

The initial state probability vector, π , is re-estimated in a similar fashion, by calculating the likelihood of the model starting in each state given the observation sequence.

$$\hat{\pi}_i = \gamma_1(i) \quad (3.11)$$

At each iteration of the adaptation procedure, all three sets of HMM parameters can be updated using equations 3.9, 3.10 and 3.11, or any combination of the three sets can be held constant. Baum-Welch training is guaranteed to find a local maximum, but is reliant on good initialisation, particularly for the observation probability values, in order to find the global maximum. We use Baum-Welch training to adapt the HMM parameters for each individual track. This approach allows the model to adapt to the particular harmonic features of the track in question, which would not be possible if adaptation were performed on a large dataset to create a single setting for the model parameters.

3.2.3 Decoding

In this work we make use of two forms of HMM decoding. We want to find both the individual likelihood of each key at each time step, and the optimal complete key sequence, given the model and the observed chord sequence for the song. Both of these problems have analytical solutions, although there are various methods of addressing the latter that vary in their definition of the optimal state sequence.

The likelihoods of every key at each time step have been previously calculated as part of the adaptation procedure, and are simply the values of $\gamma_t(i)$, as given by equation 3.7, after the final adaptation iteration. We refer to these values of $\gamma_t(i)$ as the posterior state probabilities.

The Viterbi algorithm is used to find the most likely sequence of keys. The following description is adapted from Rabiner's tutorial [Rabiner, 1989]. The algorithm begins by defining $\delta_t(i)$, the highest probability of any single key sequence that accounts for the first t chords and ends in key i . The argument that maximised $\delta_t(i)$ is retained, in $\psi_t(i)$. $\delta_1(i)$ and $\psi_1(i)$ are initialised as

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (3.12)$$

$$\psi_1(i) = 0. \quad (3.13)$$

After initialisation the values of $\delta_t(i)$ and $\psi_t(j)$ are calculated recursively.

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i)a_{ij}]b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (3.14)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i)a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (3.15)$$

When the end of the chord sequence is reached, the final state of the most likely sequence, q_T^* , is known.

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (3.16)$$

It is then possible to track back through time using the values of $\psi_t(j)$ to find the state q_t^* at time t that maximised landing in state q_{t+1}^* at time $t+1$.

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots 1 \quad (3.17)$$

MATLAB code for HMM training and decoding was taken from Kevin Murphy's HMM toolbox [Murphy, 1998].

3.3 Evaluation Technique

The algorithm was tested on songs from the first 8 Beatles albums, using hand-annotated chord labels provided by Harte [2005, 2005] as the observations. The labels include the constituent

pitch classes for the chord (excluding non-harmony notes, as judged by Harte) with a chord type if applicable (major, minor, no chord, etc.), and a start and end time for every chord. We modified the annotations to cover only simple triads and *no chord*: triad extensions were ignored, and non-triadic chords were mapped to the closest triad type using the `degrees2quality` function in the C4DM chord toolbox¹. If a chord type is included in the label the function returns it directly. Otherwise, if only the pitch classes are given, the function translates the pitch classes into a binary pitch class profile (PCP) to form a 12-element row vector, then calculates the inner product of the profile with five chord templates, stored in matrix \mathbf{S} , for major, minor, augmented, diminished and suspended chords. The chord type giving the highest inner product is selected.

$$\text{chord type} = [\arg \max(\mathbf{PCP} \times \mathbf{S})] \quad (3.18)$$

$$\mathbf{S} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 5 & 0 & 5 & 0 \\ 6 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 4 & 0 \\ 4 & 4 & 0 & 0 & 4 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Since our model does not account for suspended chords, we separately map suspended chords into the major category, since there are on average more major chords than minor in the Beatles collection. This will inevitably lead to some spurious major chord symbols, however in practice, in the Beatles chord annotations, suspended chords are relatively rare so average performance will not be greatly affected. In general it would be more appropriate to directly model only major and minor chords, as we do in chapter 4 when we work with audio data.

It is possible to use chord labels directly from the annotations as our observations, regardless of how long each chord lasts. However, we later extend the model to work with audio that does not have labelled start and end times. To enable fair comparisons between the symbolic and audio-enabled models we sample the chord sequence at a regular rate, such that sample times that fall between a chord start and end time are given the corresponding chord label, and any others are labelled *no chord*.

In order to produce a quantitative evaluation of the key analysis algorithm, an experiment to test its ability to extract the overall key of a song was devised. A subjectively-assessed ground truth is provided by Pollack [2000], who gives a detailed analysis of all the Beatles' songs. It

¹Chord tools to be made publicly available at <http://www.elec.qmul.ac.uk/digitalmusic/downloads>

should be noted that the ground truth often mentioned more than one key, in which case the first was taken to be the most important. Also several of the songs are modal or include modal inflections, and do not directly fit the model of major and minor keys. Lydian and Mixolydian modes were treated as major, and Dorian and Aeolian as minor. We chose to include these songs in our test collection despite the likelihood of their causing the results to degrade, because in many cases the songs include aspects of both modal and tonal harmony so it was not always clear into which category the song should be placed. To reference modal music from within an otherwise tonal piece is a valid creative choice, and we did not wish to exclude such music from our analysis.

To determine the overall key, the posterior state probability matrix (see examples in figures 3.20(a) and 3.21(a) on pages 82 and 83), which contains the likelihood of each key at each time step, was summed across time, giving an overall likelihood for each key. The key with the largest likelihood value was taken to be the key of the song.

$$\text{key} = \arg \max \left(\sum_{t=1}^T \mathbf{PSP}_t \right) \quad (3.19)$$

where t is the frame number, T is the total number of frames, and \mathbf{PSP}_t is a vector containing the posterior state probabilities for frame t .

All results are given as percentages of correct key assignments for the test collection, and the MIREX measure is also presented (see section 2.6 on page 53 for a description of the MIREX measure).

3.3.1 Parameters Tested

We test several model parameters in order to discover which are the most important, and to optimise the model, as described in the following sections.

Likelihood of Staying in the Same Key

The number of frames spent in a single state will be dependent on the frame hop size and the tempo of the music. If the hop size is small and the tempo is slow the music will stay in the same key for more frames than if the hop size is large and the tempo is fast, on average. This means that the key profile correlation value of 1 used for initialising the state self-transition probabilities will not necessarily be appropriate for our model. In fact, it seems likely that the value for making a transition to the same key should be significantly increased with respect to the other transition probabilities, since we would expect only a few key changes within a single piece of music. We vary the self-transition likelihood before normalisation between 1 and 16 to find an empirical optimum.

All other parameters are kept constant, with the observation and remaining transition probabilities initialised with the Krumhansl rating data as described in section 3.2.1, only the prior state probabilities and the state transition probabilities were updated during the training process, and each observation sample was a complete chord of the length given by the annotations. Figure 3.7 shows the results.

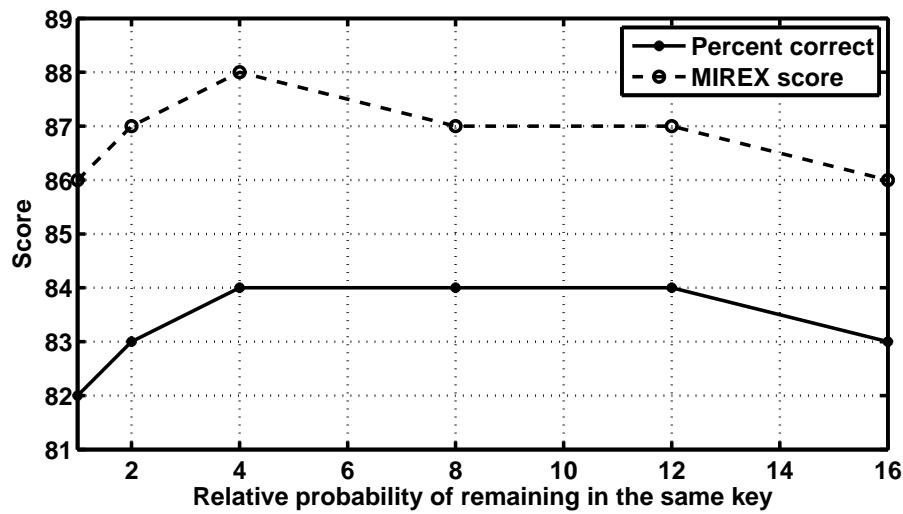


Figure 3.7: Main key estimation scores for different self-transition probabilities.

The best self-transition value was 4, with between-state transitions ranging from -0.673 to 0.651 (all before normalisation). However, the differences in performance were small, since for all values tested above 2 the self-transition likelihood is several times larger than all other probabilities, and so dominates. For a standard HMM such as the one used here, the state duration probability exhibits an exponential decay with time, which may not be well-suited to modelling the duration of a tonal centre within a piece of music. For many of the songs in the Beatles collection there is only one constant key, but in these cases the model is likely to force a change of key in its analysis if the self-transition probability is too low. However, our method of summing the posterior state probabilities to find the main key will reduce the impact of any short changes to other keys. There also exist variants of HMMs that allow alternative state duration probability densities to be specified [Murphy, 2002]. However, determination of the most appropriate probability density function is non-trivial, and so such studies will be left as a topic for future research. For all subsequent experiments using single chords as observations we use a standard HMM with relative state self-transition probability of 4.

Model Initialisation

In order to test the importance of the initialisation parameters, we test the model using three different sets: those described in section 3.2.1 that are derived from the Krumhansl probe tone experiments, a flat version, and a random initialisation. For the flat version of the emission

matrix, chords within the key are initialised with probability 0.99, and 0.01 otherwise, then normalised to sum to unity for the key. Similarly, for the flat transition matrix, the probability of staying in the same key was initialised to 0.99, and that of changing key to 0.01, then they were normalised. The elements of the random transition and emission matrices were taken from a uniform probability density function then normalised.

Figure 3.8 compares our best result from section 3.3.1 against the models initialised with flat and random profiles. In all three cases only the prior and transition probabilities were allowed to be updated, and each observation was a complete annotated chord.

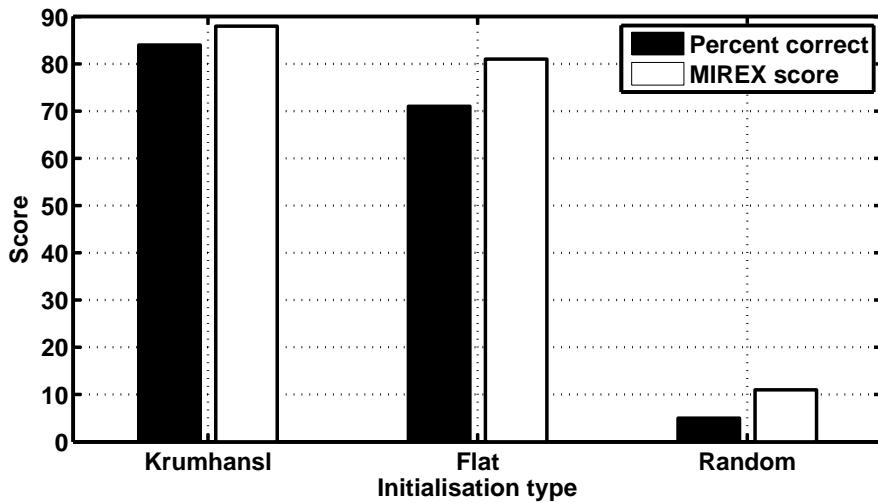


Figure 3.8: Main key estimation scores for different initialisation parameters.

Initialisation with the Krumhansl profiles proved far superior to the other two methods. Using the flat profiles resulted in a loss of 13% accuracy (7% MIREX score), and the random initialisation was unsuccessful. For a random choice of key we would expect to be correct for 1 in 24 tracks, or 4%, with a MIREX score of a little over 10, and the performance here was only just higher than these values.

The EM training algorithm is well known to be dependent on good initialisation, which is supported by these results. When the initial probabilities are random and unsupervised learning is used, as in our case, there is no longer any reason to interpret the hidden states as keys, so random initialisation is not appropriate for our model. We also see the importance of good initialisation in the difference in performance between the flat profiles, which only indicate which chords are diatonic, and the Krumhansl profiles, which give weights to the diatonic chords representing their importance. We therefore select the Krumhansl initialisation parameters for our model.

Adaptation of Model Probabilities

The observation probabilities represent the likelihood of a key generating a given chord, which can be interpreted as the extent to which a given chord implies a key using Bayes' rule [Duda

et al., 2001, p. 615]. This relationship is a perceptual one, which may vary between listeners, but in the chord rating data we already have values that are the average of 10 people's judgements. Although the chords will vary from song to song, we believe that the keys implied by a given sequence will remain the same for a given listener (or for the “average” listener), so we argue that disallowing updates of the observation probabilities during the adaptation phase will give the best results. Another way of presenting the same argument is that the only way of specifying the meaning of the hidden states is through their relationship to the observations, and in an unsupervised training operation such as ours there is no mechanism to ensure that the meaning of the states is maintained. Hence we can no longer interpret the state sequence as a key sequence if the observation probabilities are adapted. We therefore test the effects of disallowing updates of the observation probabilities. In order to verify that our choice of initialisation parameters is good, we also test the effects of disallowing updates of the initial state distribution and the state transition matrix. Figure 3.9 shows the percentage of correctly assigned overall keys for the 110 songs in the first 8 Beatles albums, with different HMM parameters adapted. MIREX scores are also given.

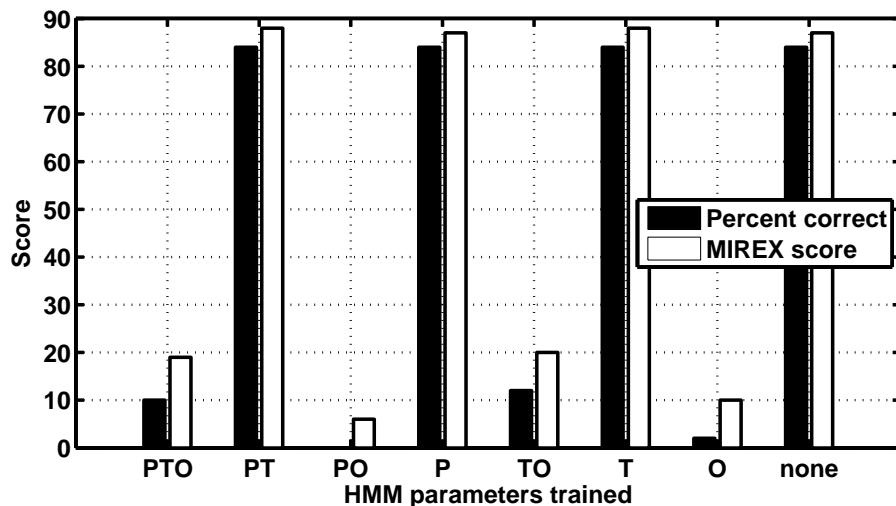


Figure 3.9: Main key estimation scores when different sets of parameters are adapted. P=prior probabilities, T=transition probabilities, O=observation probabilities.

The results in figure 3.9 verify that disallowing adaptation of the observation probabilities gives the most accurate representation of the song, since allowing adjustment alters the meaning of the hidden states. Adaptation of the prior state probabilities had little effect on the number of songs correctly classified. This is most likely due to the step where the key probabilities across the whole song were summed: the prior probabilities will only affect the first few frames and will therefore have limited effect on the overall key estimation. The suitability of the perception-based initialisation was confirmed by the case with no adaptation, which differed from the best model only in the MIREX score. Adaptation of the transition probabilities with or without adaptation

of the prior probabilities for each song gave the highest score of 84 % of songs correctly classified, with a MIREX score of 88.

Chord Sampling Interval

The chord sequence sampling interval, analogous to the frame hop size in our audio-enabled model (see chapter 4), was also varied to find a suitable value. This value affects the likelihood of the music changing key between samples: if the sampling interval is very small the music is much more likely to remain in the same key from one sample to the next. Figure 3.10 shows the results for various sampling intervals, and the case where observations were taken directly from the annotations so each sample is the length of one chord, resulting in a varying sampling interval. Results are given as percentages, and the MIREX measure is also shown (see table 2.1).

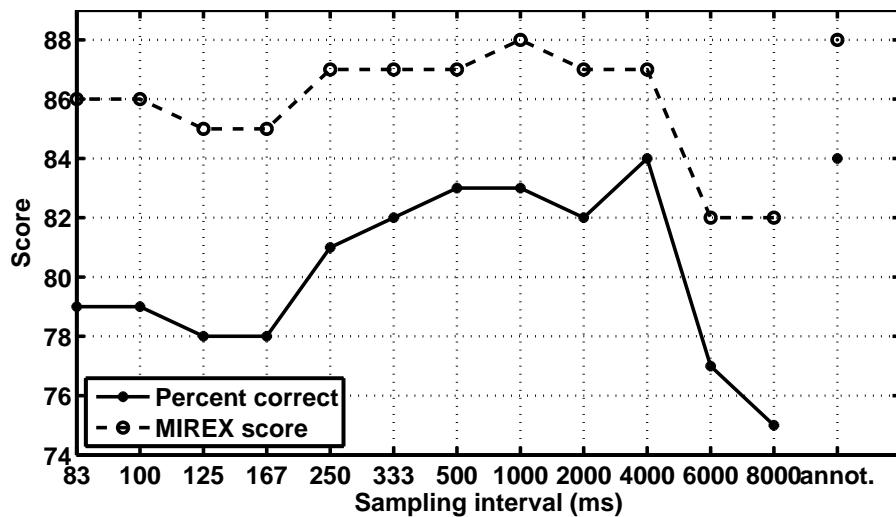


Figure 3.10: Main key estimation scores for different chord sampling intervals, and with a varying chord sampling interval determined by hand annotations of chord times (indicated by “annot.”).

The observations based directly on the annotations gave the best results. This is not surprising since in section 3.3.1 we optimised the state self-transition probabilities for the annotated observations. In this case the model probabilities are most stable with respect to the chord changes in the music rather than a fixed sampling interval, which leads us to test the effects of using beat-length frames for our audio-based model in chapter 5, section 5.3.1.

Of the examples with fixed sampling rates, a sampling interval of 4000 ms, or 4 s, gave the best results, with fairly stable results down to about 250 ms. This preference for longer sampling intervals implies that sampling more than once per chord is not necessary, as would be expected since this would give us no additional information regarding the harmony.

Table 3.4: Krumhansl's chord transition ratings for a major key context [Krumhansl, 1990, page 193].

First Chord	Second Chord							Ave
	I	ii	iii	IV	V	vi	vii	
I		5.10	4.78	5.91	5.94	5.26	4.57	5.26
ii	5.69		4.00	4.76	6.10	4.97	5.41	5.16
iii	5.38	4.47		4.63	5.03	4.60	4.47	4.76
IV	5.94	5.00	4.22		6.00	4.35	4.79	5.05
V	6.19	4.79	4.47	5.51		5.19	4.85	5.17
vi	5.04	5.44	4.72	5.07	5.56		4.50	5.06
vii	5.85	4.16	4.16	4.53	5.16	4.19		4.68
Ave	5.68	4.83	4.39	5.07	5.63	4.76	4.76	

3.3.2 Model Structure Variations

There are two additional variations that we wish to investigate, concerning the structure of the model itself.

Using Chord Transitions as Observations

The HMM structure incorporates some of the harmonic progression information, in the sense that the key at each time step is dependent on the key at the previous time step. However, the model includes no such correspondence between consecutive chords, and, as explained in section 2.5.3, the progression of chords is of great importance in harmonic analysis.

In order to address the issue of chord progressions, we redefine the model such that each observation represents a chord transition, or pair of consecutive chords. For example, for a chord sequence **Dm-Bdim-C** the first observation would be **Dm-Bdim**, and the second **Bdim-C**.

To initialise the new observation probabilities, Krumhansl's ratings for chord transitions within a key, given in table 3.4, were used to provide numerical values for the emission matrix. The ratings only include the diatonic chords of major keys, so some additional values for other major, minor, augmented and diminished triads and for minor keys were created using alternative rating data and guided by our understanding of harmony.

The pre-normalised probabilities for staying on the same chord, for diatonic chords, were taken from the ratings of individual chords within a key, given in table 3.3. The values of these ratings begin at 2.76, which is much lower than the chord transition ratings, but repeated chords on either a frame-by-frame or beat-by-beat level are very likely, so we want the self-transition ratings to be higher. We added a constant value, hand-tuned to 2, to the individual chord ratings, which lifted the ratings for staying on the same chord to between 4.76 and 8.66, compared to between 4.00 and 6.19 for a chord transition. Pre-normalised values for transitions involving either one or two non-diatonic chords were set uniformly low, to 1, or “fits poorly” within the key context on

Krumhansl's seven point rating scale. We refer to these initialisation parameters as "Krumhansl 1". We also tested a version in which values for transitions involving exactly one non-diatonic chord were set to 2, and those for transitions between two non-diatonic chords were set to 1. We refer to this version of the initialisation as "Krumhansl 2".

For minor keys the ratings for major keys corresponding to the same scale degrees were used, using the chords of the harmonic minor mode. The emission matrix was then normalised so that the observation probabilities summed to unity for each key. The final emission matrix had dimensions $24 \times (48+1)^2 = 24 \times 2401$, since there are 48 possible chords and the possibility of *no chord* to form the chord transitions, in 24 possible keys. This is a large increase in the size of the emission matrix, but if we are to hold the probabilities constant in the adaptation phase, as is optimal for the single chord observation model (see section 3.3.1), this should not be problematic.

For the "flat" version of the model the observation probabilities before normalisation were set to 0.99 where both chords are within the key and 0.01 otherwise, and the probabilities for the "random" initialisation were again drawn from a uniform probability density function.

Experiments that varied the self-transition likelihood, model initialisation type and HMM parameters adapted were carried out on the new chord transitions model. Figures 3.11 to 3.14 show the results.

Figure 3.11 shows that varying the probability of remaining in the same key had little effect on the results. All values tested, from 1 to 16, were significantly larger than values for changing to any other key, which were taken from the correlations between key profiles and so ranged between -0.683 and 0.651 before shifting and scaling to allow them to be treated as probabilities. It is likely that if the self-transition probability approached the range of other values we would see some deterioration in key estimation performance.

We see in figure 3.12 that the initialisation parameters are also very important when using chord transitions as observations. Even with some hand tuning both versions of the Krumhansl initialisation performed better than the flat initialisation. Once again random initialisation was unsuccessful, since we cannot be sure that the hidden states represent keys in this case, and if they do we have no way of knowing which state corresponds to which key. The best results of figure 3.13 were achieved with no adaptation at all, which shows that a good initialisation is much more important than training.

Figure 3.14 shows a dramatic improvement in the results for smaller sampling intervals, so we performed another iteration of the optimisation process using a sampling interval of 100 ms. This value is a compromise between the improved performance at higher sampling rates and shorter computation time at lower sampling rates. These results are given in figures 3.15 to 3.17.

We observe in figure 3.15 that the optimal pre-normalised state self-transition probability is

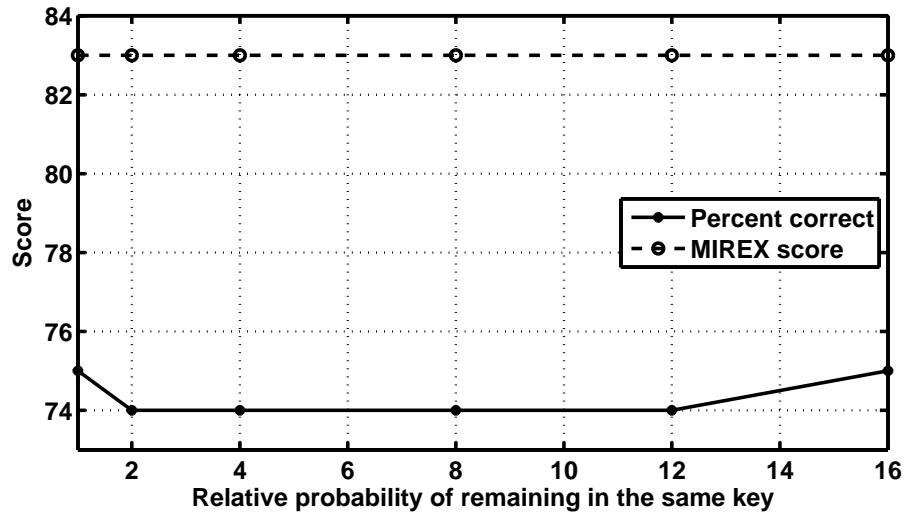


Figure 3.11: Main key estimation scores for different self-transition probabilities for the chord transitions model, with chords sampled once per annotation.

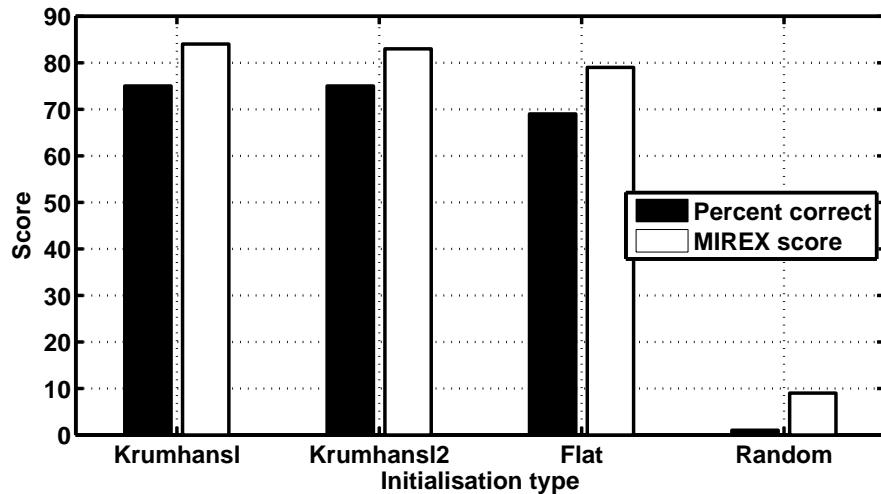


Figure 3.12: Main key estimation scores for different initialisation parameters for the chord transitions model, with chords sampled once per annotation.

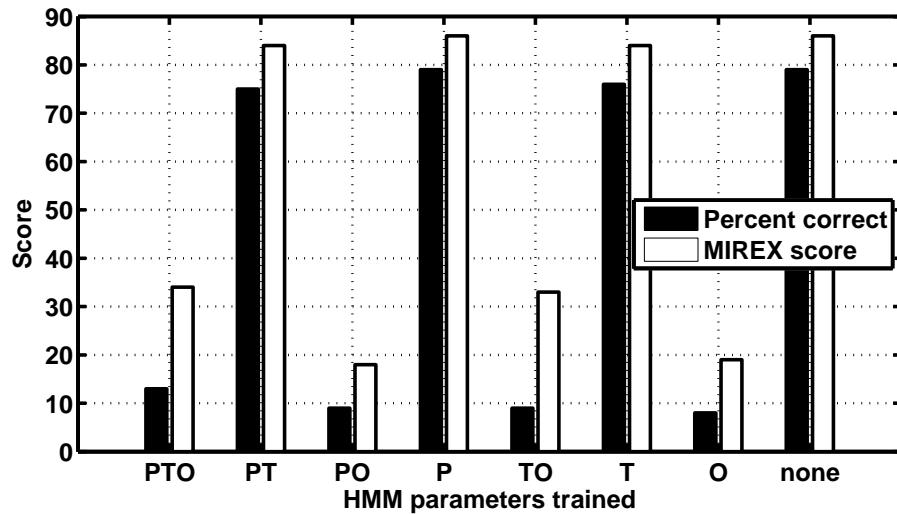


Figure 3.13: Main key estimation scores when different sets of parameters are adapted for the chord transitions model, with chords sampled once per annotation. P=prior probabilities, T=Transition probabilities, O=observation probabilities.

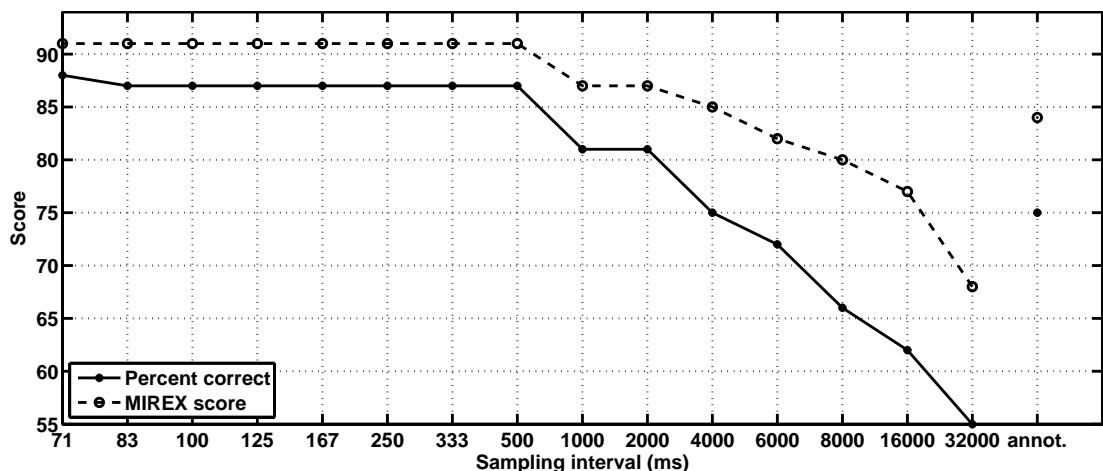


Figure 3.14: Main key estimation scores for different chord sampling intervals for the chord transition model.

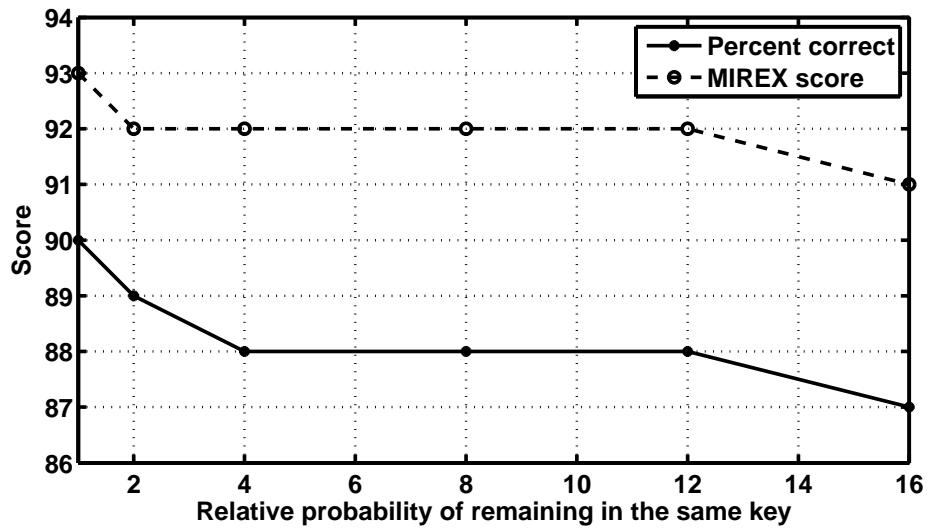


Figure 3.15: Main key estimation scores for different self-transition probabilities for the chord transitions model with 100 ms sampling interval.

1, lower than that for the single chords model. However, the differences in performance are small, and we believe that the precise value of this parameter is not of great importance so long as it is kept high enough to dominate over transitions to other keys.

We see again in figure 3.16 that knowledge-based initialisation is vital to our model, but there is no difference in performance between the two initialisation types based on Krumhansl's ratings. This shows that the relative probabilities of non-diatonic chord transitions are not important for determining the key, it is the relative values of the diatonic chords that are required for key estimation.

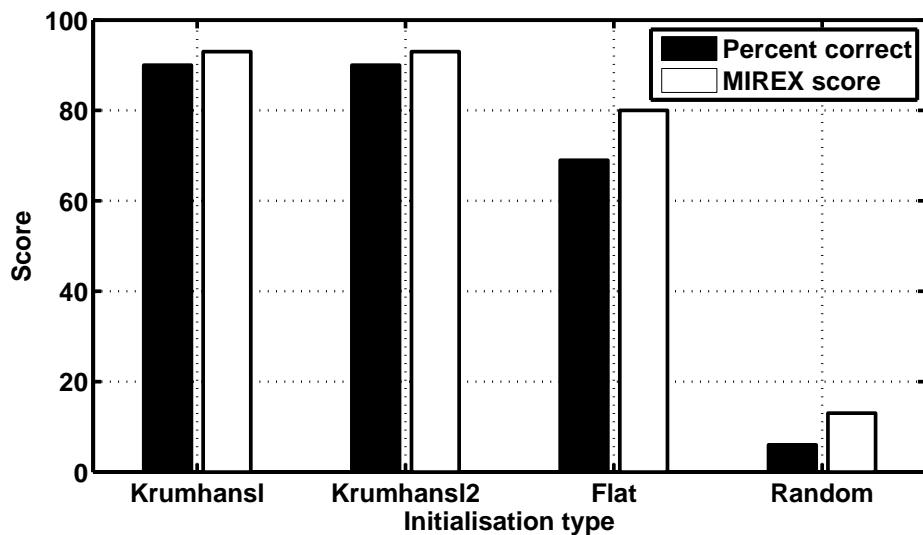


Figure 3.16: Main key estimation scores for different initialisation parameters for the chord transitions model with 100 ms sampling interval.

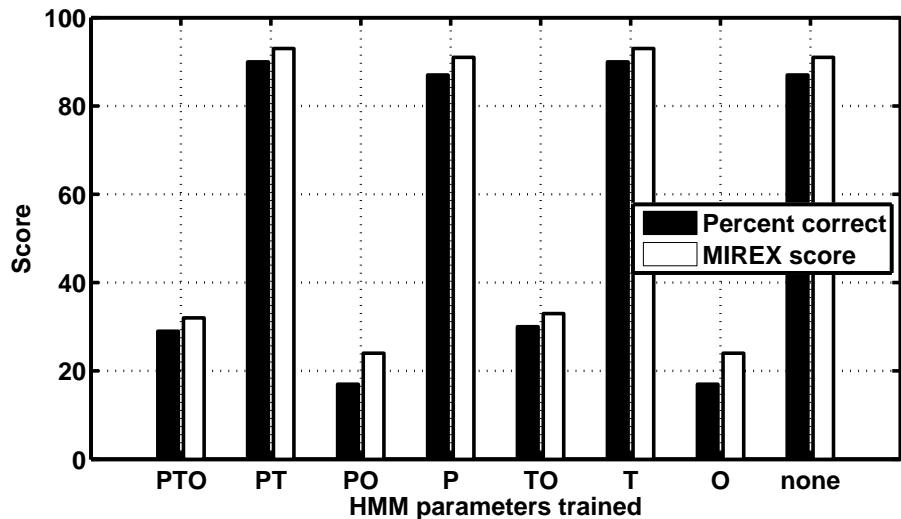


Figure 3.17: Main key estimation scores when different sets of parameters are adapted for the chord transitions model with 100 ms sampling interval. P=prior probabilities, T=transition probabilities, O=observation probabilities.

Figure 3.17 shows that adapting the transition probabilities offers improvement of a few percentage points, but adapting the observation probabilities severely reduced performance. Adapting the prior state probabilities made no difference. These training results match the findings for the model with single chords as observations (compare with figure 3.9).

In addition, using chord transitions as observations was able to give 6 % better global key recognition than the optimal single chord observation model, which suggests that using the chord transitions in this way is beneficial to tonal analysis. The optimisation process was an iterative one over a large parameter space that can find only a local optimum, so we cannot say for certain that the chord transitions model is better, but the results reported are in keeping with our expectation that modelling of chord progressions is important for tonality analysis.

The Need to Model Augmented and Diminished Triads

Now we address the question of whether it is necessary to explicitly model diminished and augmented chords as observations, or if major and minor only would suffice. Whilst diminished and augmented chords do have distinct harmonic function, they are less frequent than major and minor chords and it is plausible that they are not required to determine the key, so from an engineering perspective it may be preferable to simplify the model.

In order to test this, augmented chords were labelled as major and diminished chords as minor, and the observation probabilities corresponding to augmented and diminished chords were removed, resulting in an observation matrix of size 24×25 for the single chords model, and 24×625 for the chord transition model.

Figure 3.18 shows the results for both models, using the optimal parameter values from our

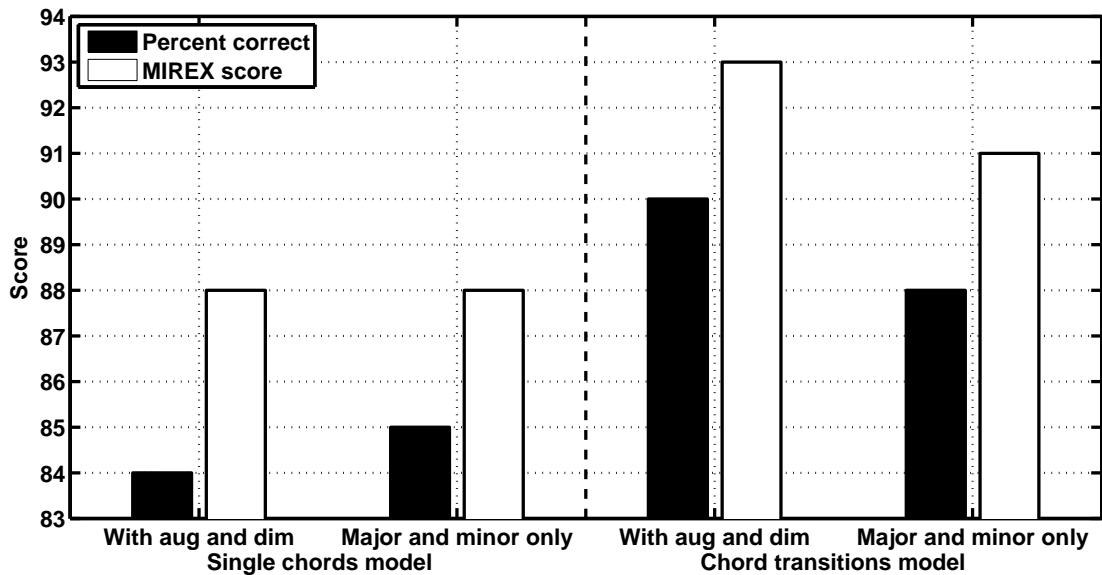


Figure 3.18: Main key estimation scores with and without augmented and diminished triads.

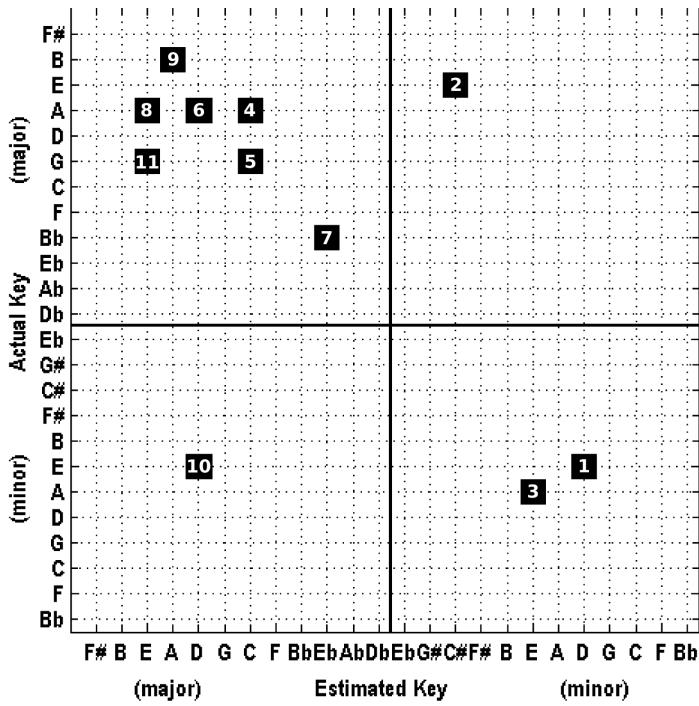
results so far. We see that inclusion of the augmented and diminished triads offers some small improvements. Therefore in all following experiments we include augmented and diminished triads as possible observations.

3.3.3 Confusion Matrix for the Best Case

We now investigate the incorrect key estimates for our best version of the model, as previously investigated in this chapter. The settings we have found to give the best performance are a relative self-transition probability of 1 before normalisation, initialisation using the results of probe tone studies (Krumhansl 1), prior and transition probabilities adapted with observation probabilities fixed, and a sampling interval of 100 ms, with the model that uses chord transitions as its observations. The parameters are described in section 3.3.1, and the chord transitions model is described in section 3.3.2.

Figure 3.19 shows the confusion matrix for the incorrect estimates. Three of the modal songs were incorrectly classified: two Mixolydian songs, *You've Got To Hide Your Love Away* and *She Said She Said*, were mistaken for the major key on their fourth degree, due to their flattened 7th, and one song with Dorian inflections, *Don't Bother Me*, was mistaken for the major key on its 7th degree, due to its flattened 3rd and 7th. These errors are comparable to errors between relative major and minor keys.

One song in A major, *Good Day Sunshine*, was mistaken for its dominant, E major. The song gives strong emphasis to chords B major and F sharp major, which are not diatonic to the key of A major and will both bias the key estimation towards the dominant. Another song in A major, *Another Girl*, was mistaken for its subdominant, explained by the particular stress on



- 1 Don't Bother Me
- 2 And I Love Her
- 3 Things We Said Today
- 4 When I Get Home
- 5 You've Got To Hide Your Love Away
- 6 Another Girl
- 7 She Said She Said
- 8 Good Day Sunshine
- 9 Doctor Robert
- 10 Being For The Benefit Of Mr. Kite
- 11 A Day In The Life

Figure 3.19: Confusion matrix for the best-performing model. Only incorrect estimates are shown. the flattened 7th degree of the scale which gave rise to G major chords, used here to give a blues feel rather than a move to the subdominant key. A song in A minor, *Things We Said Today*, was mistaken for its dominant, E minor. This is plausible, since in the main verse sections A minor and E minor chords alternate so occur equally often.

The remaining five incorrect key estimates are for songs where more than one key is mentioned in the ground truth for the home key, and it is one of these alternative keys that has been selected by the model. The songs affected are *And I Love Her*, *When I Get Home*, *Doctor Robert*, *Being For The Benefit of Mr. Kite*, and *A Day In The Life*.

We have been able to understand and explain all of the incorrect key assignments, and the nature of the errors is such that we would not expect to achieve any better results without including some higher level musical analysis such as finding the phrase structure. Lee [2008, p. 100–108] proposes a method of cadence finding from chord sequences, which could be used to aid key estimation since the cadential chords tend to be the most important for confirming a key.

3.4 The Model as a Segmentation Algorithm

The output of the Viterbi algorithm gives us the most likely sequence of keys given the chords of a particular song. This output can be used as a structural segmentation of the song, by grouping contiguous frames that are in the same key into one segment. For cases where the sections of the song, such as *verse* and *chorus*, are in different keys, the method can work well as a music structure extractor.

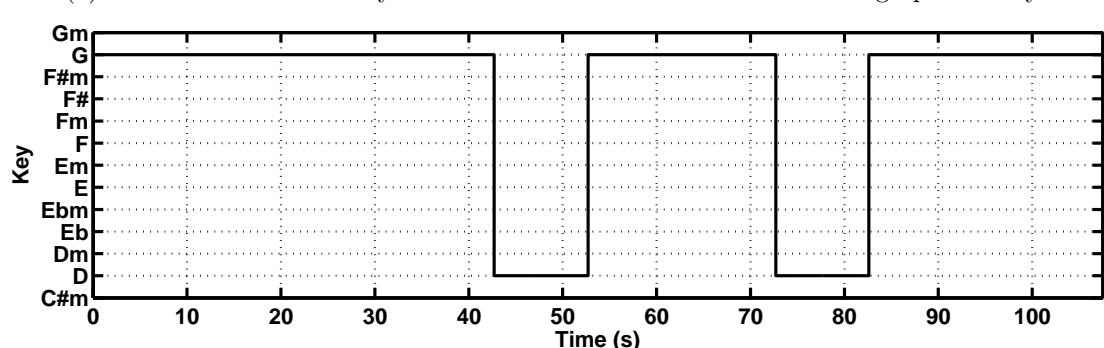
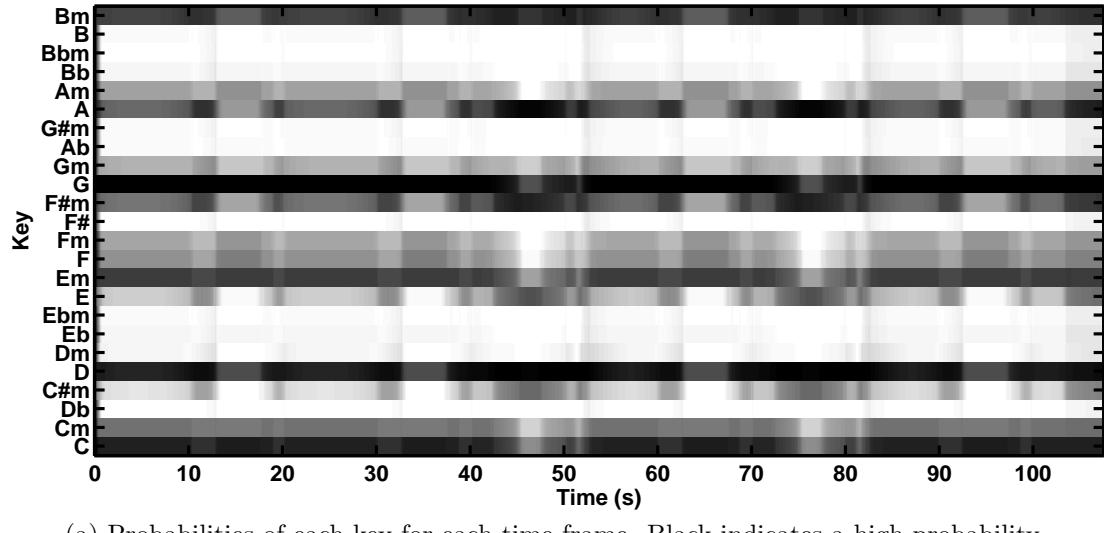


Figure 3.20: Model outputs for the Beatles' *I'll Cry Instead*.

Figures 3.20 and 3.21 show some examples of such results. The posterior state probabilities are shown in figures 3.20(a) and 3.21(a), and the most likely key sequence for the whole song is shown in figures 3.20(b) and 3.21(b). Ground truth for the key changes in the Beatles' songs is not available to our knowledge, but the figures show that the algorithm is capable of extracting meaningful structure. In *I'll Cry Instead* (see figure 3.20(b)) the two bridge passages in D major, at 42 s to 52 s and 72 s to 82 s, have been clearly separated. Similarly, in *I'm Happy Just to Dance With You* (see figure 3.21(b)) the C \sharp minor sections (choruses) and E major sections (verses) have been identified. The Beatles modified the final chorus such that the chords forming the transition back to E major are heard sooner than in previous choruses, then interrupted with C \sharp

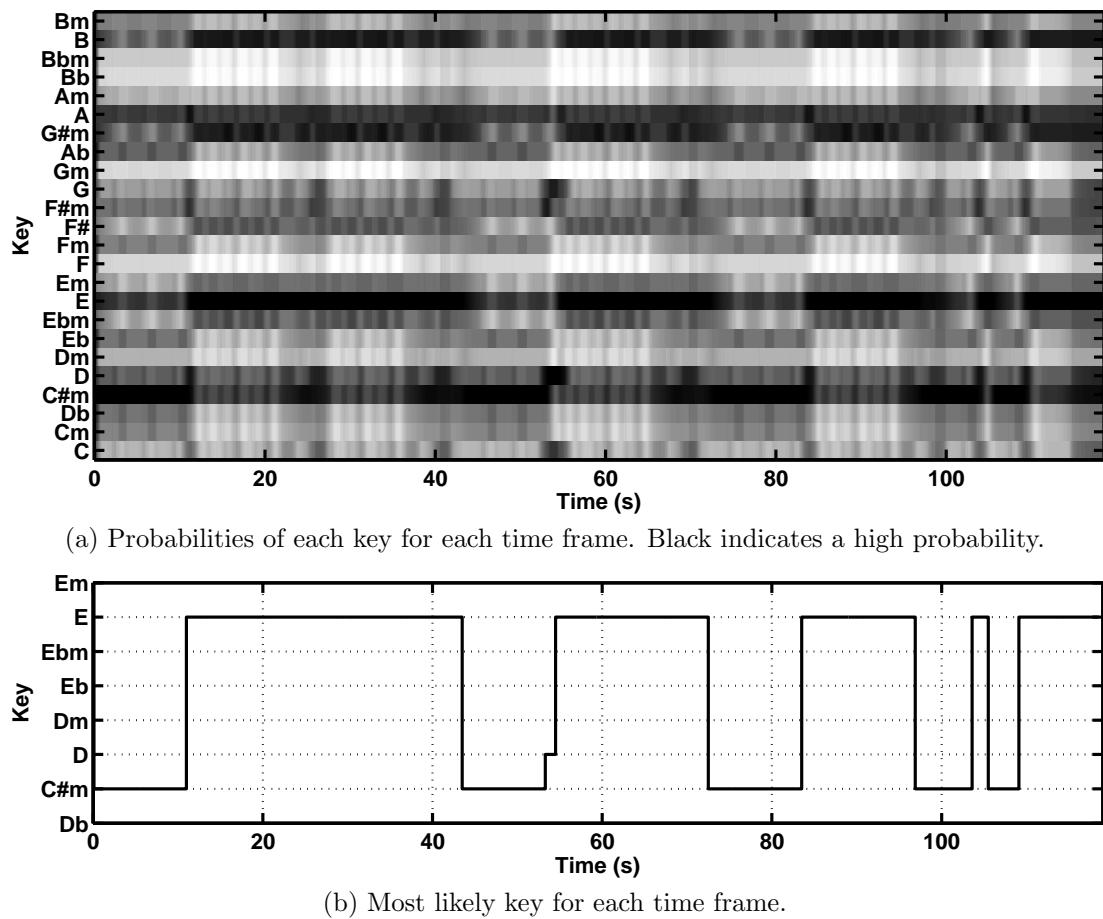


Figure 3.21: Model outputs for the Beatles' *I'm Happy Just to Dance With You*.

minor again. This transition appears in figure 3.21(b) as a short E major section at about 103 to 105 s. This demonstrates one of the weaknesses of our approach, that although the chords were most closely related to E major at this time, the key was not firmly established, but our model is not able to make such high level judgements. It is expected that this type of error would occur less frequently if the chords' position relative to the musical phrases were taken into account.

Inspection of figures 3.20(a) and 3.21(a) reveals further structure that is not apparent in the hard key classification. For example, in the first E major section of *I'm Happy Just to Dance With You*, two consecutive verses are played. The hard classification of key shown in figure 3.21(b) cannot show the repetition, but the same period in figure 3.21(a) shows a pattern in the posterior state probabilities of approximately 16 s in duration that is repeated once. Alternative structure extraction techniques such as similarity matrix analysis [Foote, 1999] may be able to identify this repetition.

Some further investigation of the algorithm as a musical structure extraction technique is given by Levy et al. [2007].

3.5 Summary

In this chapter we have introduced our use of a hidden Markov model to represent the musical relationships between chords and keys in order to perform an analysis of the tonal progressions in a piece of music, from the sequence of chords. Using standard HMM decoding techniques we can extract the relative probabilities of every major and minor key at every time step, and the most likely complete key sequence.

We tested the model's capability of finding the overall key of a piece of music using the songs in the first 8 Beatles albums, for which we have human annotated chord symbols and main key labels available. In order to optimise the model we performed experiments to test several of the parameters.

We adjusted the initial probability setting for remaining in the same key, but found that it did not make a significant difference to performance for all of the values tested, which all placed the probability of remaining in the same key higher than the probability of moving to any other key.

The initialisation parameters were found to be very important. Initialisation based on perceptual ratings of harmonic relationships was found to be the most successful, with theoretical flat (binary) initialisation performing moderately well, but random initialisation did not work at all, even with extensive training.

Allowing the observation probabilities to be updated during the adaptation phase was extremely detrimental to results. We believe this is because if the observation probabilities are allowed to change we can no longer be certain of what the hidden states represent, since we do not directly observe them. Adaptation of the prior and transition probabilities did offer some improvement in performance, but good initialisation was found to be much more important than training.

We tested a more complex model that uses chord transitions, or pairs of chords, as the observations, in order to incorporate more information about the progression of chords in the music. This model was able to improve main key estimation accuracy by 6 percentage points. It may be fruitful to investigate alternative methods if incorporating temporal information, such as integration of features to give a harmonic context, or measuring chord transitions as vectors in a geometric tonal space.

We also studied the effects of removing augmented and diminished triads from the model, which made only a small difference to performance so would be a suitable way of simplifying the model if required.

The single chord observation model gave the best results when the data was sampled once per chord, but the chord transition observation model performed better with smaller sampling

intervals, of 0.5 s and below.

As a result of this optimisation process the model we use henceforth is one that has chord transitions as its observations, with all major, minor, augmented and diminished triads included. We initialise using human rating data (Krumhansl 1, defined in section 3.3.2), with a state self-transition value of 1 (before normalisation), and allow only the prior and state transition probabilities to be updated during the adaptation phase. The chords are sampled at a rate of 10 per second. These values are summarised in table 3.5.

Table 3.5: Best parameters for the symbolic model.

Parameter	Value
Observations	chord transitions
Relative state self-transition likelihood	1
Chord sampling interval	100 ms
Initialisation	Krumhansl 1
Adaptation	Prior and transition probabilities

We acknowledge that the model was optimised using only songs by the Beatles, so may not give the best results for other types of music. However, we do not have chord annotations for other types of music available, and we believe that the most significant results, that good initialisation is important and that the observation probabilities should not be adapted, will hold for other types of music, given that the differences in results produced by varying these parameters was so large. The different sets of initialisation parameters were not genre-specific, and the argument that adapting the observation probabilities will alter the meaning of the hidden states is not related to the music on which the model is trained. The other parameters tested did not produce such large differences in performance, so it likely to be less important for these to be at their exact optimum for a more general test set.

Our best version of the model was able to give 90 % correct main key estimation on the Beatles data, and we were able to explain and understand all of the errors. The type of errors seen could only be avoided by including some much higher level musical analysis such as cadence finding, so we would not expect to achieve performance above 90 % using a hidden Markov model as described in this chapter.

We showed that in addition to finding the main key of a song, the output of the Viterbi algorithm can be used to perform a harmony-based segmentation. For cases where the musical structure includes key changes this can be an effective structure extraction technique.

The purpose of beginning our investigation in the symbolic domain was to understand the model as much as possible without the complications of using audio data. However, if our approach is to be widely applicable it must operate on an audio input. Having achieved the

best performance we can in the symbolic domain, we continue in chapter 4 by adding audio functionality.

Chapter 4

Extension of the Model for Audio Input

We have introduced a model of chord-key relationships that has been successful for main key estimation from chord symbols. However, for it to be generally applicable we wish to adapt it to take audio data as its input. This chapter extends the model by adding an audio chord recognition algorithm as a first step, which provides chord labels to our symbolic model. The chord recognition algorithm uses a template-matching approach. We compare the performance of different chord templates, which model the presence of upper partials in the signal to varying degrees. We also adapt the HMM to have a continuous observation probability density function in the chroma feature space, thus eliminating the need for a separate chord recognition step, and compare its performance to the discrete model.

4.1 Audio Test Collections for Global Key Recognition

The amount of available test music is immediately increased when the model is adapted to work with audio data, since we no longer have to rely on the results of human annotation tasks. The experiments in this chapter are therefore run on our complete test set, which consists of six separate collections.

MIREX The training data supplied to contestants of the 2005 MIREX audio key estimation competition [MIREX competition, 2005a]. The collection consists of the first 30 seconds of 96 pieces of classical music, two in each major and minor key. The music was synthesised from MIDI. Main key annotations are provided as part of the training data.

Alkan 25 Preludes, op. 31, played by Olli Mustonen (piano). There is one prelude in each key, with an extra one in C major to make 25 tracks with the main key given in the title.

Beatles All songs from the first 8 albums by the Beatles, a total of 110 tracks. Main key annotations are taken from a detailed study of the Beatles' work by Alan Pollack [2000].

The collection included 99 songs in a major key and 11 songs in a minor key.

Mixed classical A mixture of music with the key specified in the title. Composers represented are: Albinoni, C.P.E. Bach, J.S. Bach, Beethoven, Brahms, Chopin, Dvořák, Haydn, Mozart, D. Scarlatti, Schubert, Shostakovich, Vivaldi. Various instrumentations are also included: piano solo, orchestra, various solo instruments plus orchestra, various chamber ensembles, choir. The collection includes 29 pieces in a major key and 21 in a minor key.

Rachmaninov 24 Preludes, op. 3 no. 2, op. 23 no. 1-10, op. 32 no. 1-4 and op. 32 no. 5-13, played by Vladimir Ashkenazy (piano). There is one prelude in each key, with the main key given in the title.

Bach Recordings of the Preludes and Fugues from the Well-Tempered Clavier, Book 1. There is one Prelude and one Fugue in each key, giving a total of 48 tracks with the main key given in the title. The pieces were played by Glenn Gould on a piano.

4.2 Addition of a Chord Recognition Step

The chord recognition algorithm that we use to give an input to our discrete HMM is described by Harte and Sandler [2005]. It begins with a 36-bins-per-octave constant-Q transform covering 9 octaves, to give a total of 324 constant-Q frequency bins. The constant-Q data is folded down to a chromagram and tuned using a quadratic interpolation of the peak frequencies within the pitch class, resulting in a tuned, 12-bin chromagram. Refer to section 2.3.3, and the work of Harte and Sandler [2005] and Brown and Puckette [1991, 1992] for details.

The chromagram calculation requires a number of low-level parameters to be set, such as the downsampling factor and frame hop size. These parameters are described in detail in chapter 5 together with experiments to test the effect of varying them. Table 4.1 shows the parameters we use at this stage of the study. We do not downsample the data, in order to keep as much information as possible, so the sampling rate remains at 44.1 kHz. The maximum constant-Q frequency is chosen to be just below half the sampling rate, and the minimum constant-Q frequency corresponds to MIDI note E1, which is the lowest note on a standard bass guitar and a perfect fifth above the lowest note on a standard 88-key piano, so should encompass all of the fundamental frequencies (it is possible for lower notes to be played but very rare). The frame size is a function of the minimum constant-Q frequency, and is included for reference. The hop size and kernel threshold are based on the recommendations of the authors of the chord recognition algorithm [Harte and Sandler, 2005].

To give a sequence of chord estimates from the tuned chromagram, the inner product of each frame of the chromagram and each possible rotation, corresponding to each possible root note,

Table 4.1: Parameters used for chromagram calculation.

Parameter	Value
Downsampling factor	1
Max. frequency (Hz)	21096
Max. frequency (MIDI note)	E10
Min. frequency (Hz)	41.2
Min. frequency (MIDI note)	E1
Frame size (seconds)	1.5
Frame size (samples)	65536
Hop size (frames)	1/8
Hop size (seconds)	0.19
Hop size (samples)	8192
Sparse kernel threshold	0.0055

Table 4.2: Simple chord templates used by the chord recognition algorithm.

Chord type	Template
major	1 0 0 0 1 0 0 1 0 0 0 0
minor	1 0 0 1 0 0 0 1 0 0 0 0
augmented	1 0 0 0 1 0 0 0 1 0 0 0
diminished	1 0 0 1 0 0 1 0 0 0 0 0

of four binary chord templates is calculated. The four templates correspond to major, minor, augmented and diminished triads, and are shown in table 4.2. The template and rotation giving the highest inner product for each frame are used to indicate the estimated chord. Let \mathbf{w}_{gh} be a vector containing the chord template for chord type g with root h , and \mathbf{c} be the chroma vector for the current frame, then

$$\text{chord} = \arg \max_{g \in \{1,2,3,4\}} \left(\arg \max_{h \in \{1,\dots,12\}} (\mathbf{c} \cdot \mathbf{w}_{gh}) \right) \quad (4.1)$$

The resulting estimated chord sequence is then used as a direct replacement for the hand-annotated chord symbols used in chapter 3.

4.2.1 Modelling Upper Partials in the Chord Templates

The simple chord templates of table 4.2 do not take account of the upper partials present in audio signals. The relative magnitudes of upper partials vary significantly between instruments, and even for different notes and volumes on the same instrument, so creating a model that accurately represents the upper partials for any input is extremely complex. However, Gómez [2006, p. 77–78] has had success in a template-matching key estimator using a very simple model of the upper partials, and we apply the same approach to our chord templates. She assumes that the partials decay exponentially with frequency. This is a generalisation, but we will show that on

average it gives better accuracy than ignoring the upper partials altogether.

Table 4.3: Contributions of the upper partials to the note template, for note C.

Partial number	0	1	2	3	4	5	6	7
Pitch class	C	C	G	C	E	G	B \flat	C
Contribution to note template	1	s	s^2	s^3	s^4	s^5	s^6	s^7

The new template for a C major chord will therefore have nonzero values not only for C, E and G, but also for their upper partials, D, B, etc. We start by calculating a note template using the harmonic series shown in table 4.3. The decay factor, s , is a parameter that we investigate, and must satisfy $0 \leq s < 1$ for exponential decay with frequency. The fundamental and seven upper partials span just over 3 octaves, so we sum together values for corresponding pitch classes to give a 12-value template as shown in table 4.4. This note template can be regarded as a model 12-element chroma vector for the note. Templates for other notes are created simply by rotating the template for C such that the root note always has value $(1 + s + s^3 + s^7)$.

Table 4.4: Calculation of the template for note C, with 7 upper partials modelled.

Pitch	Value
C	$1 + s + s^3 + s^7$
C \sharp	0
D	0
E \flat	0
E	s^4
F	0
F \sharp	0
G	$s^2 + s^5$
G \sharp	0
A	0
B \flat	s^6
B	0

To create templates for chords the appropriate note templates are added together. Table 4.5 shows the calculation of the C major chord template, which is the sum of the C, E and G note templates. Figure 4.1 shows the major, minor, augmented and diminished chord templates on C for $s = 0.6$.

We also examine the effect of truncating the harmonic series earlier, by testing templates that model only the fundamental and first 3 partials. These templates are created in exactly the same way, but the terms in powers of s higher than 3 are omitted from the calculations. Some alternative approaches to modelling upper partials [Pauws, 2005, Peeters, 2006a] adapt the constant-Q features for symbolic profiles by enhancing frequency peaks that have strong components at upper partial frequencies. Our approach of adapting symbolic profiles to audio

Table 4.5: Template for chord C major, with 7 upper partials modelled.

Pitch	Value
C	$(1 + s + s^3 + s^7)$
C♯	0
D	$s^6 + (s^2 + s^5)$
E♭	0
E	$s^4 + (1 + s + s^3 + s^7)$
F	s^6
F♯	0
G	$(s^2 + s^5) + (1 + s + s^3 + s^7)$
G♯	s^4
A	0
B♭	s^6
B	$(s^2 + s^5) + s^4$

data requires much less online processing because the profiles can be calculated in advance, but restricts the processing to chroma features, which do not contain pitch height information. Our choice to investigate modelling 3 and 7 upper partials was determined by the restriction to chroma features: the first 3 partials span 2 octaves above the fundamental, and the first 7 partials span 3 octaves, which are then folded down to exactly span the chroma feature vector.

An interesting property of both of these adjusted templates is that the value for the fifth of the chord is always higher than the value for the root note for the major and minor templates, for all s , which conforms to the music-theoretic principle that the dominant is particularly important. This could mean that the tonic-dominant errors in the key estimation are particularly affected by the modelling of upper partials. We will revisit this idea during our analysis.

4.3 Performance of the Model with Chord Symbols Extracted from Audio

Table 3.5 at the end of chapter 3 summarises the parameters of our best symbolic model. We use this model with automatically-extracted chord symbols to estimate the main key for all six of the music sets. We present the results graphically across different values of s and with 0, 3 and 7 upper partials modelled as part of the chord templates, in figures 4.2 to 4.4.

We turn first to figure 4.2, which shows the percentage and MIREX scores (see section 2.6 on page 53 for a description of the MIREX score) for each dataset separately and the average across all of the datasets for the case where 3 upper partials are modelled. We see that on average a decay factor, s , of 0.6 gave the best performance, about 4.5% higher than with the simple templates ($s = 0$). When the value of s is very small the contributions of the harmonics will be too small to make a significant difference to the performance, but if s is too high the contributions will be higher than those found in the mainly natural instrument sounds in our dataset. The

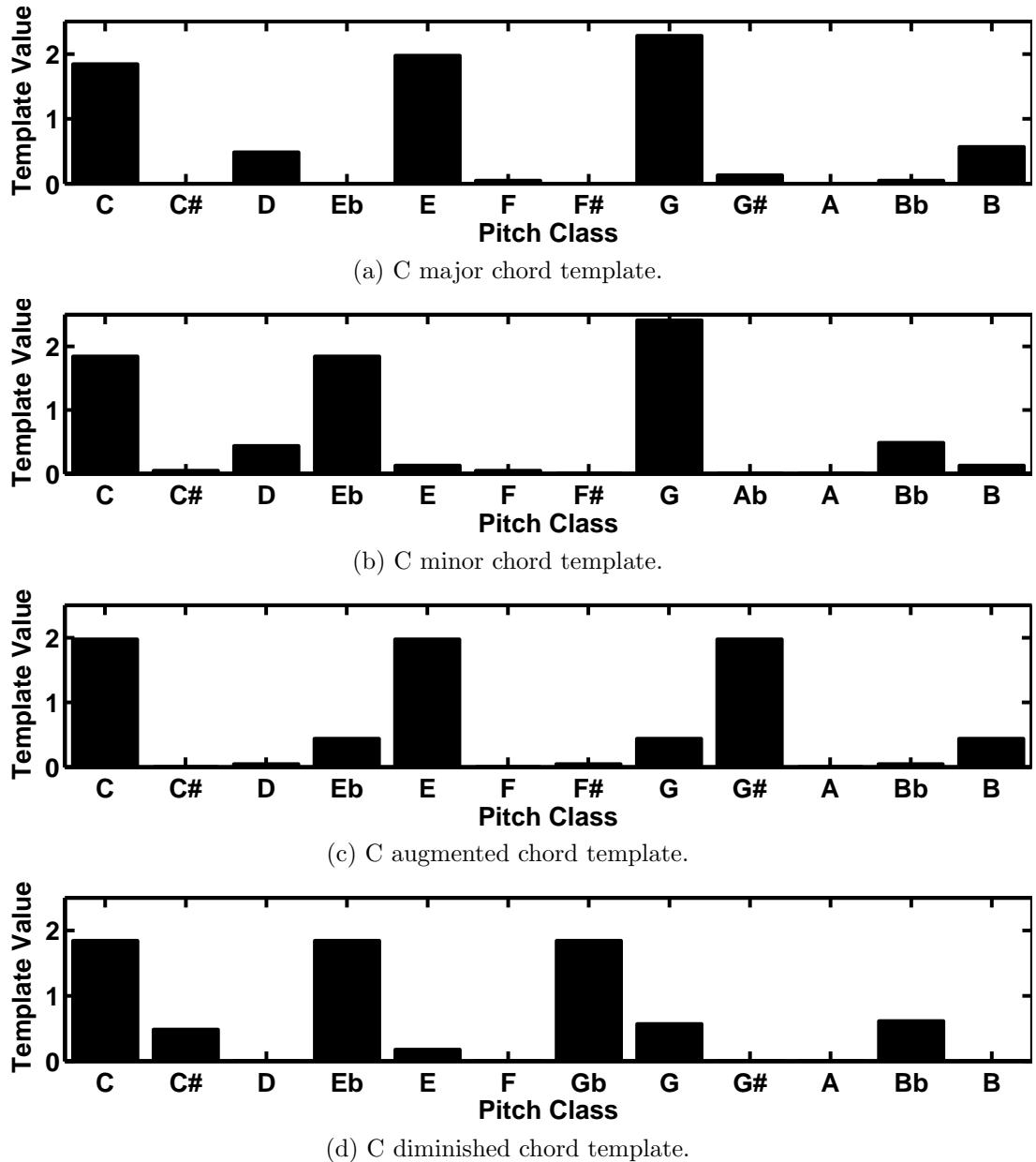


Figure 4.1: Templates for chords on C, modelling 7 upper partials, with a decay factor, s , of 0.6.

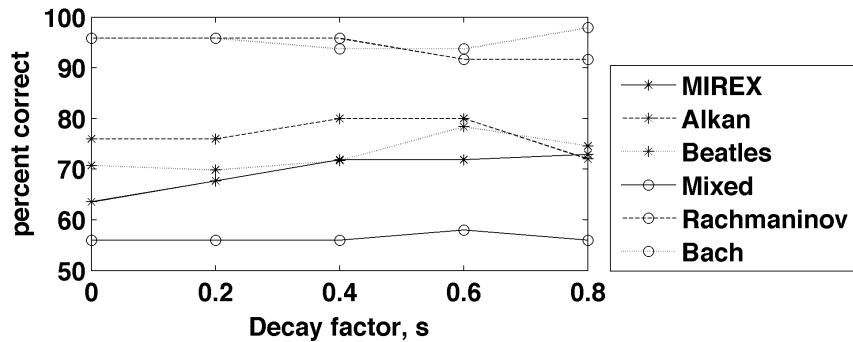
observed results show a steady increase in mean performance up to $s = 0.6$, and start to decrease again at $s = 0.8$, and so the pattern fits with our expectations. Gómez also uses $s = 0.6$ [Gómez Gutiérrez, 2006, p. 78].

The Bach and Rachmaninov datasets defy the trend, both showing a small drop in performance as s is increased from 0.2, then for the Bach set performance improved again at $s = 0.8$. The modelling of upper partials is an extensive simplification of natural acoustic sounds and so cannot be expected to be ideal for every recording. It has improved results particularly for the Beatles test set, and slightly for the mixed classical test set, which are the two with the most varied timbres, so we conclude that the exponential decay model is a useful approximation even though it has been less successful for the Bach and Rachmaninov collections, which are both played entirely on a piano. Scores for both of these collections are consistently very high, and above 90% even for the worst case, so we continue to use the exponential decay model.

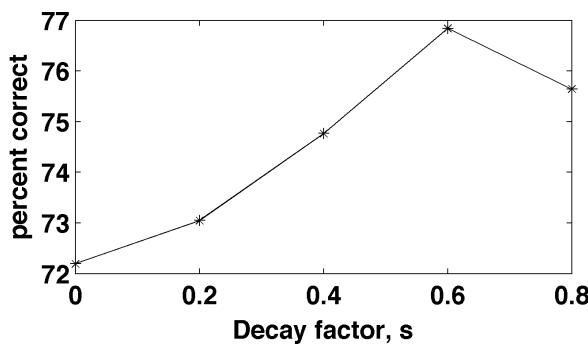
Figure 4.3, which shows the results for the case where 7 upper partials were modelled, shows a similar pattern, with the peak performance again at $s = 0.6$. The MIREX scores show a local minimum at $s = 0.4$, but the total variation in average MIREX scores is less than 1% so we disregard it.

Figure 4.4 allows us to examine the relationship between performance and the number of partials modelled, for the best value of s , 0.6. The figure shows that on average it is beneficial to model the upper partials, and the best overall performance was with 3 upper partials modelled, just under 2% higher than with 7. We attribute this to the simplicity of the exponential decay model: acknowledgement that the strongest 3 partials are present in the audio signal has improved results, but attempting to model up to the 7th partial with an exponential function appears to be a poor generalisation of the varied timbres in the music. The contribution of the seventh partial is 0.028 when $s = 0.6$, which is a very small contribution compared to that of the fundamental (which has a contribution of 1), so is insignificant when the instruments in the recordings vary. We select the best performing templates to use in subsequent experiments, which model 3 upper partials with s at 0.6.

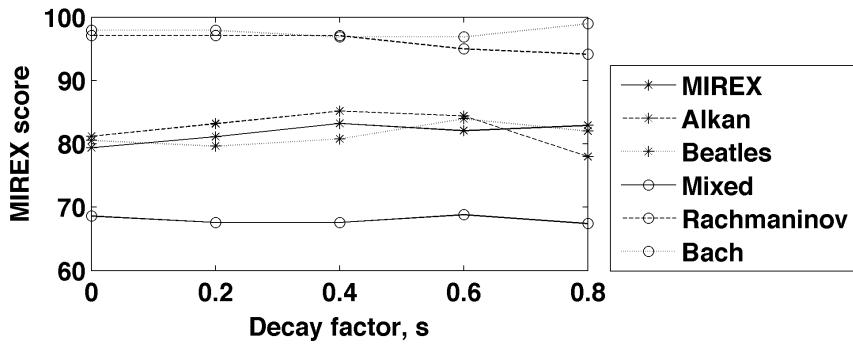
Figures 4.5 and 4.6 allow us to examine in more detail the types of error that modelling the upper partials has the greatest effect on. Figure 4.5 shows the errors for the simple chord templates, divided into 5 categories according to the relationship of the estimated key to the correct key. Figure 4.6 shows the same information for our best-performing model, with 3 upper partials modelled and $s = 0.6$. The figures show that the altered templates reduced the number of dominant key errors more than any other type of error. We attribute this to the contribution of the 2nd partial of the root note adding to the template value for the dominant note, and thus giving the dominant a higher template value than the root. This will mean that, for example,



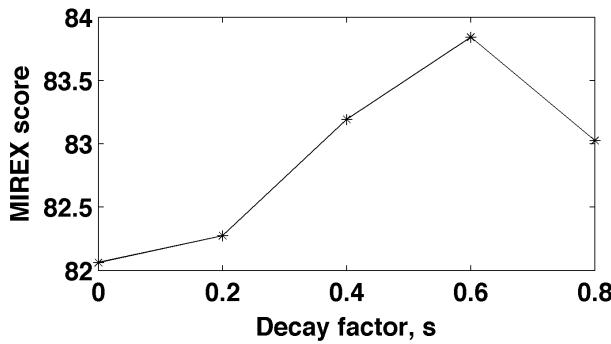
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.

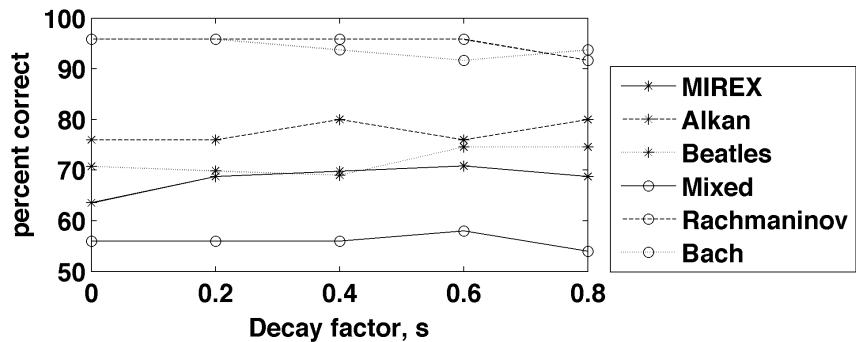


(c) MIREX scores for the separate music collections.

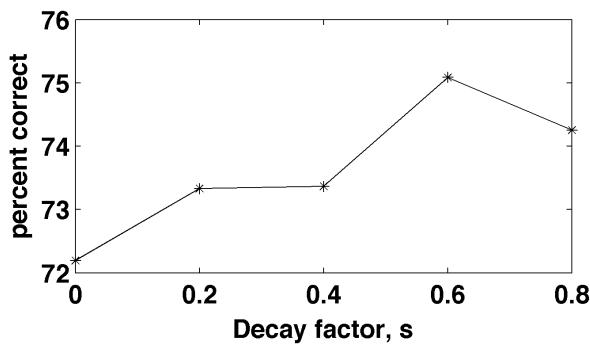


(d) MIREX scores for all of the collections together.

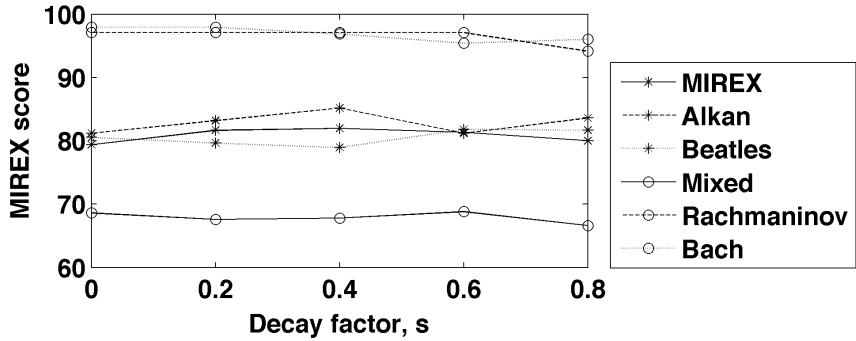
Figure 4.2: Main key estimation scores when using a decay factor, s , of between 0 and 0.8 in the chord templates, with 3 upper partials modelled.



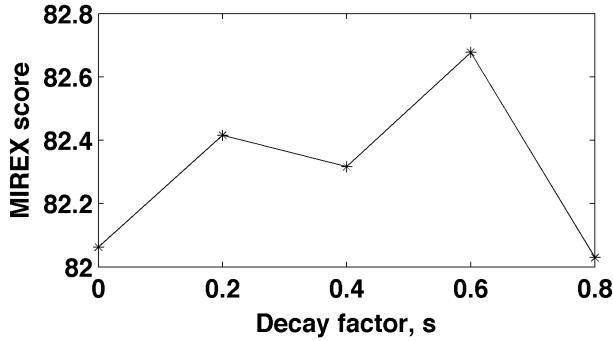
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 4.3: Main key estimation scores when using a decay factor, s , of between 0 and 0.8 in the chord templates, with 7 upper partials modelled.

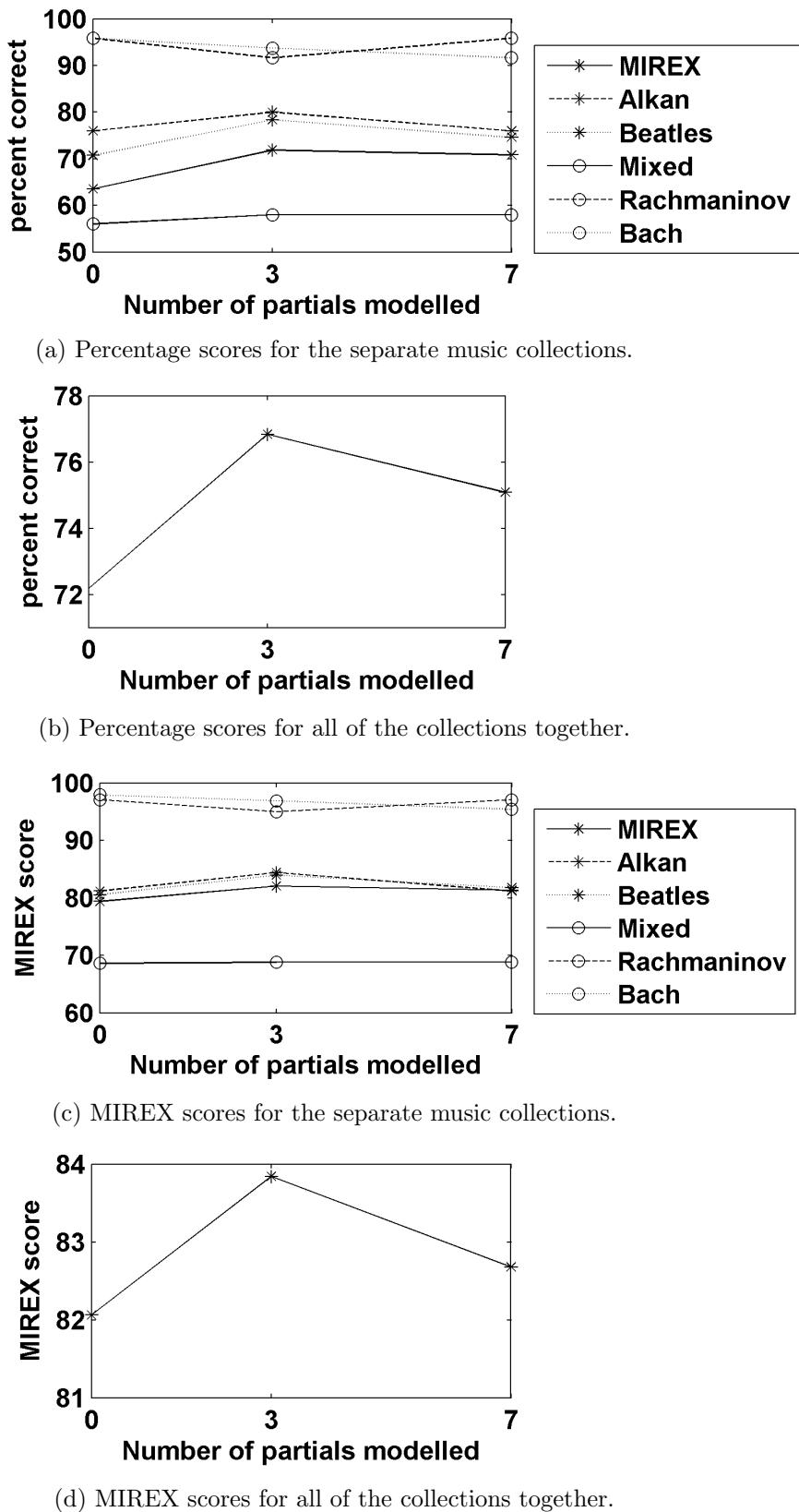


Figure 4.4: Main key estimation scores when modelling 0, 3 and 7 upper partials in the chord templates with a decay factor, s , of 0.6.

when a chroma vector has a strong G component the probability of a C chord being chosen over G is increased, so the chord estimation will tend more towards the subdominant. Differences between the models in all other types of error are very small.

Figure 4.7 shows more clearly the performance of our best model so far, with $s = 0.6$ and 3 upper partials modelled, across the different test collections. Scores are very high for the Bach and Rachmaninov test sets, which we attribute to their having generally less complex chord sequences than the other classical collections. We did not expect such high performance on the Rachmaninov collection due to the post-Romantic style of Rachmaninov’s compositions, however informal listening tests confirmed that in these Preludes the harmony does remain relatively stable. More thorough and formal studies would be required to confirm this in relation to our other datasets.

The model performed comparatively poorly on the mixed classical dataset. The pieces in this collection are on average longer than the other datasets, which allows more scope for key changes and may cause the main key to be less clear. To test this we also applied our best performing model to only the first 30 seconds of every track, and give the results in figure 4.8. We see that for only the mixed classical collection the performance was improved by using just the first 30 s, and the scores are brought up to a similar level to the scores for the other collections. This supports our hypothesis that the pieces in the mixed classical collection include more tonal development than the pieces in other collections.

4.4 Adaptation of Model for Continuous Observation Probability Density

The hidden Markov model tested so far is a discrete model, that is dependent on a template-based chord recognition algorithm. In this section we remove the need for hard chord classification by adapting our HMM to have a continuous observation probability density over the chroma feature space.

We would like to maintain a connection between chords and implied keys, since it is through this connection that humans perform tonality analysis, so rather than allowing the model to freely adapt to the chroma data we interpret the continuous observations as weighted mixtures of chords, instead of the more usual learned weighted mixtures that do not necessarily clearly correspond to particular musical features. We model only single chords as the observations, since modelling chord transitions poses additional complications in the continuous domain and we have found that for the discrete case single chords are effective as observations (see chapter 3). We use the same key transition probabilities as for the discrete version, but model a chord as a single multivariate Gaussian in the 12-dimensional chroma space, and the observation probability

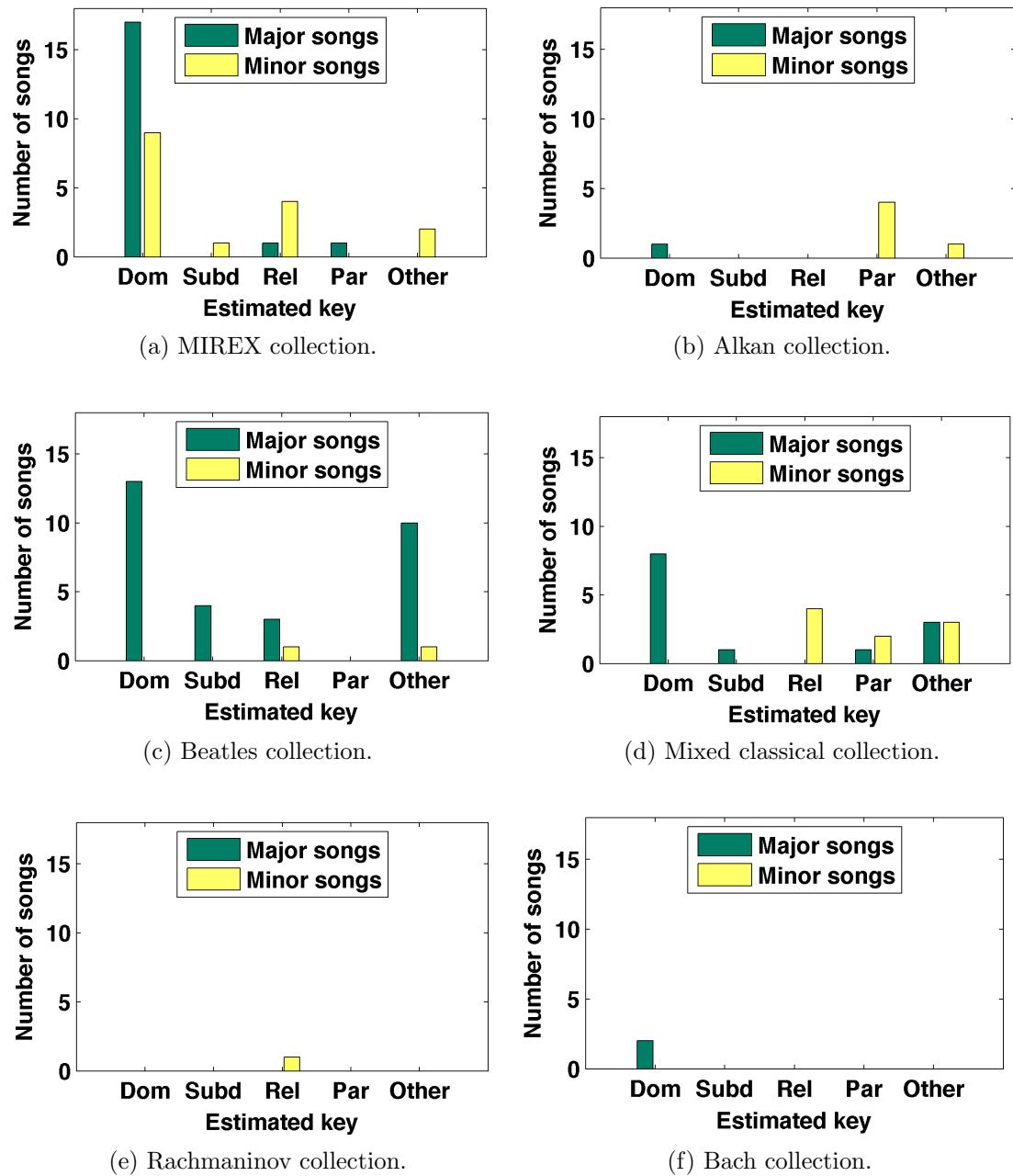


Figure 4.5: Distribution of errors with respect to the correct key for each test collection, when the simple chord templates that do not model any upper partials are used. Dom = dominant, Subd = subdominant, Rel = relative, Par = parallel.

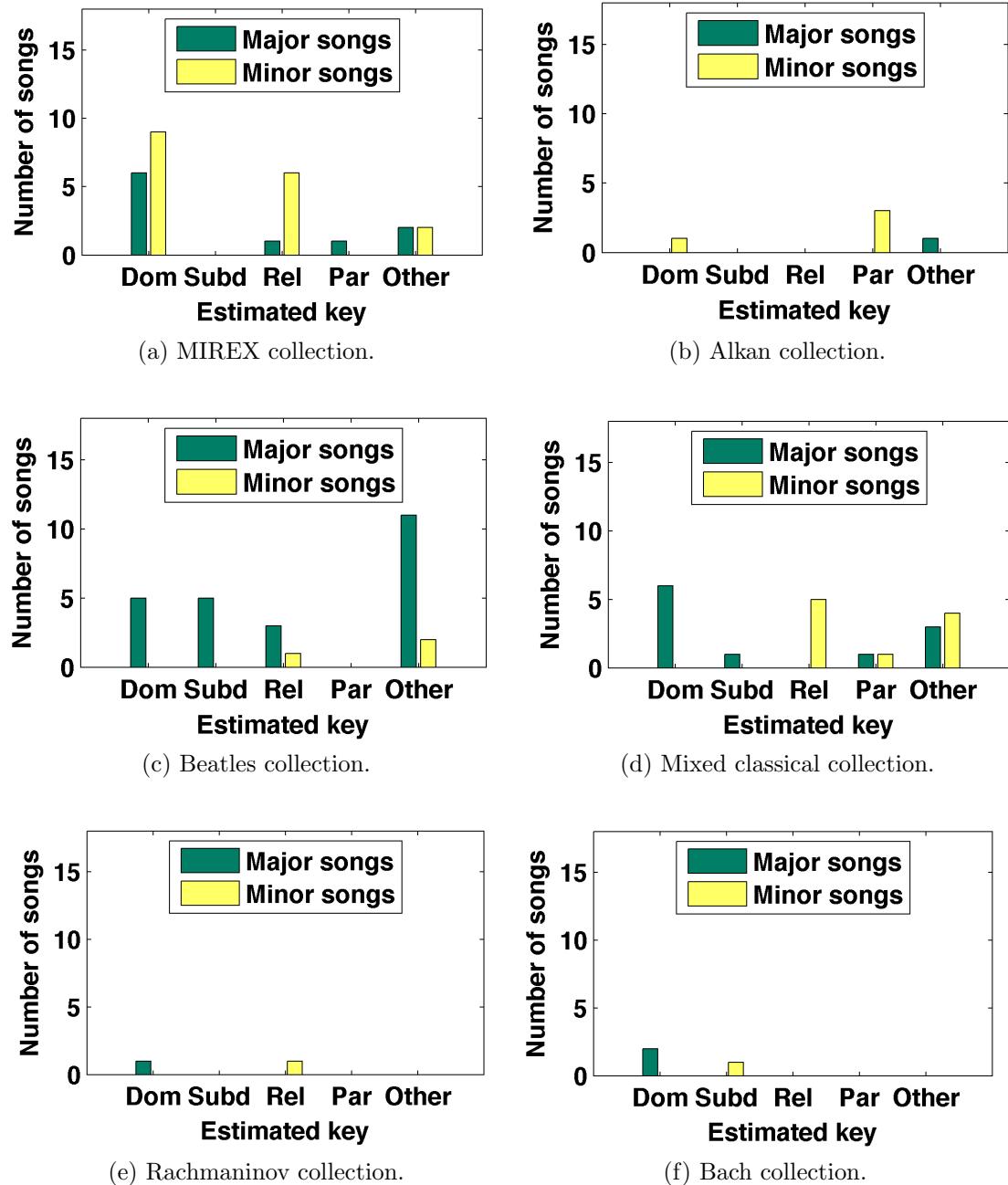


Figure 4.6: Distribution of errors with respect to the correct key for each test collection, when 3 upper partials are modelled in the chord templates with a decay factor, s , of 0.6. Dom = dominant, Subd = subdominant, Rel = relative, Par = parallel.

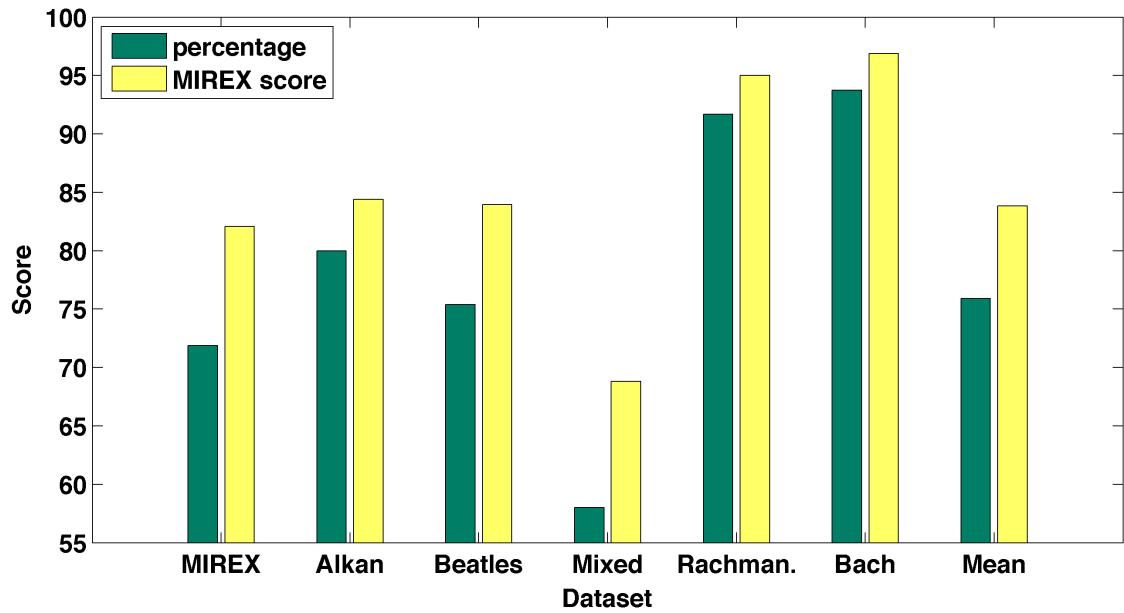


Figure 4.7: Percentage and MIREX scores for our best performing chord templates, with $s = 0.6$ and 3 upper partials modelled.

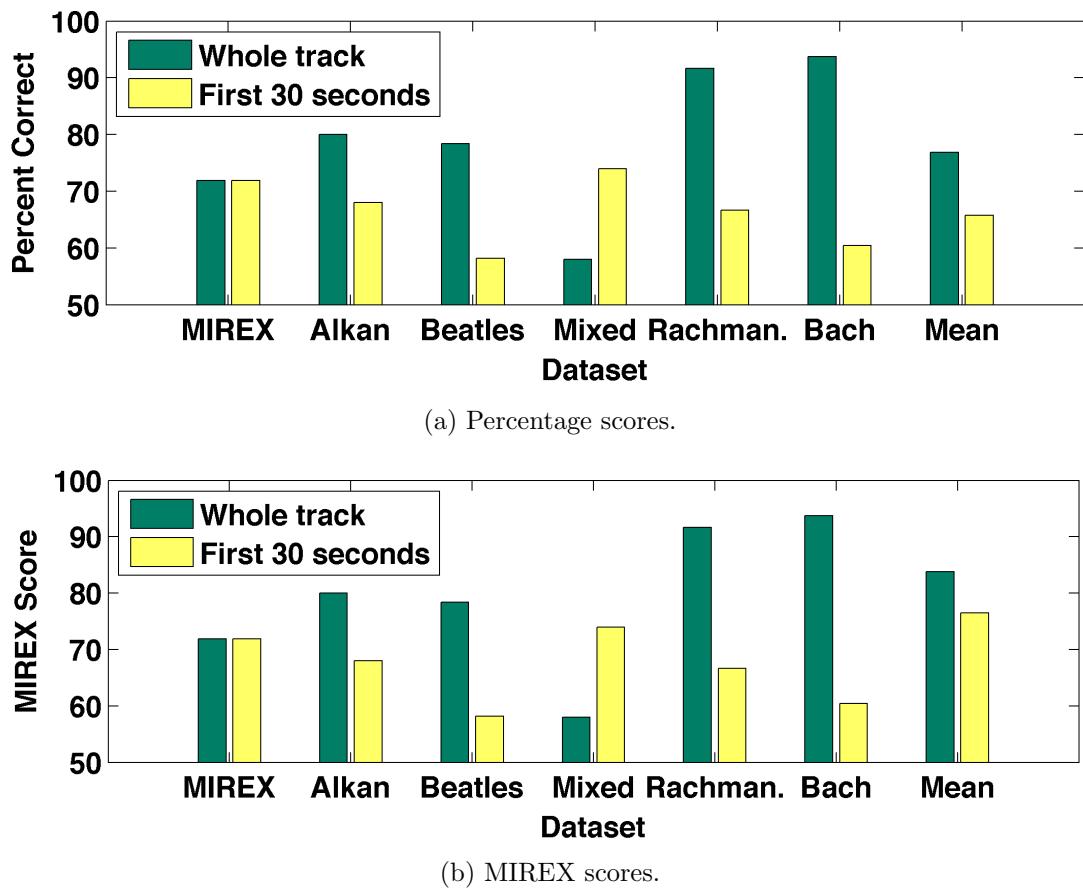


Figure 4.8: Main key estimation scores for our best performing model on the whole track and on only the first 30 seconds.

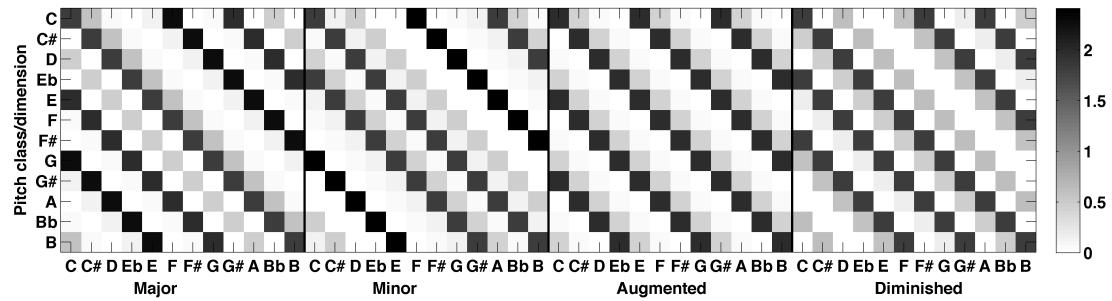


Figure 4.9: Visualisation of the mean values of the 12-dimensional Gaussian for each chord in the continuous model.

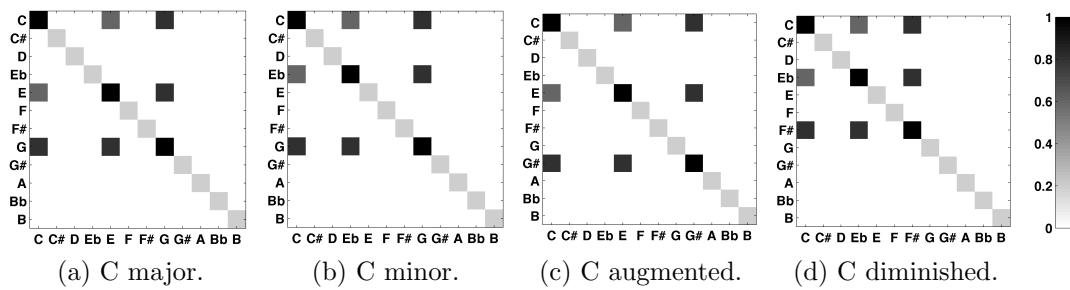


Figure 4.10: Visualisations of the covariance matrices for chords on C in the continuous model.

density function becomes a weighted Gaussian mixture of the 48 different chords. Our previous observation probabilities, that correspond to the likelihood of a chord occurring in a given key, become the mixture weights. They are the elements of the emission matrix shown in figure 3.6 on page 65.

The mean positions for the Gaussian functions, each representing one chord, are taken from our chord templates that model 3 upper partials. The templates are used directly for chords with C as the root, and rotated to give templates for chords with other roots. Figure 4.9 depicts the mean vectors for each chord.

For the covariance matrices we use values arrived at heuristically that have been shown to be successful by Bello and Pickens [2005]. The covariance represents the extent to which two pitches vary together, and the values are chosen such that the notes of the chord are more correlated than non-chord notes. We use values of 1 for the covariance of each note of the triad with itself, and 0.2 for the covariance of non-chord notes with themselves. The covariance of the root note with the fifth and of the third degree with the fifth are both set to 0.8, and the covariance of the root note with the third is set to 0.6. All other values are set to 0. Figure 4.10 depicts the covariance matrices for chords with C as the root. Training was carried out on a per-song basis, using the expectation-maximisation algorithm as implemented by Murphy [1998].

We evaluated the performance of the continuous HMM on all six test sets, and present the

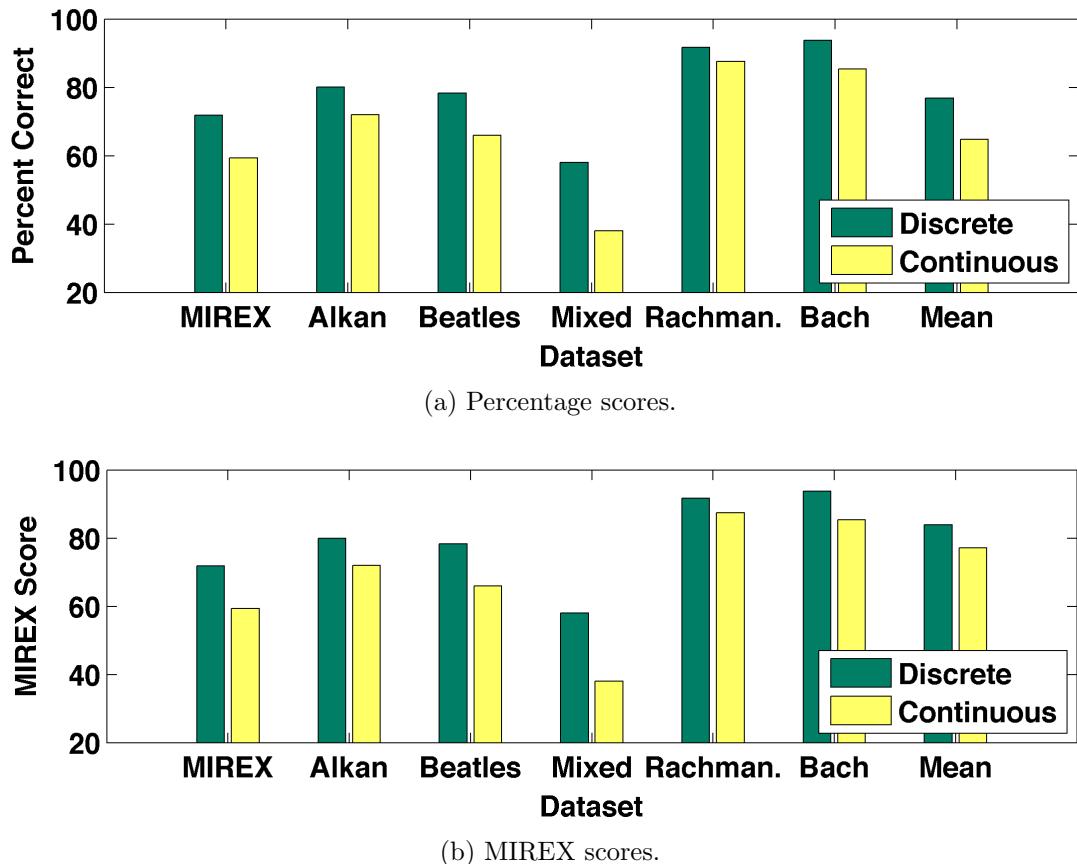


Figure 4.11: Main key estimation scores for the best performing discrete HMM and the continuous HMM.

results in figure 4.11 together with the results for our best-performing discrete model. We see that in all cases the discrete model performed better, particularly for the mixed classical dataset. Figure 4.12 shows the errors for each dataset separated by their relationship to the correct key. Dominant key errors are by far the most common (compare with figure 4.6 for the discrete model).

The strong tendency towards the dominant key could be a result of an imbalance in the mixture weights, for which the values are given in table 3.3 on page 62. For both major and minor keys the subdominant chord has a higher rating than the dominant, which seems counterintuitive at least in terms of Classical harmony where the focus is on tonic-dominant relationships and the perfect cadence is used as the most conclusive way to confirm a key. We investigated the effects of adjusting the mixture weights to give the dominant chord more importance in a key and the subdominant chord less importance. For major keys we change the subdominant rating (F major in the table) from 5.59 to 4.5 and the dominant rating (G major) from 5.33 to 6.5. For minor keys we change the subdominant (F minor) rating from 4.60 to 3.5, the dominant major rating (G major) from 4.38 to 5.5 and the dominant minor rating (G minor) from 3.48 to 4.5. The errors using these new ratings are shown in figure 4.13. Dominant errors are reduced, however the overall performance, shown more clearly in figure 4.14, is still not as good as for the discrete model. The

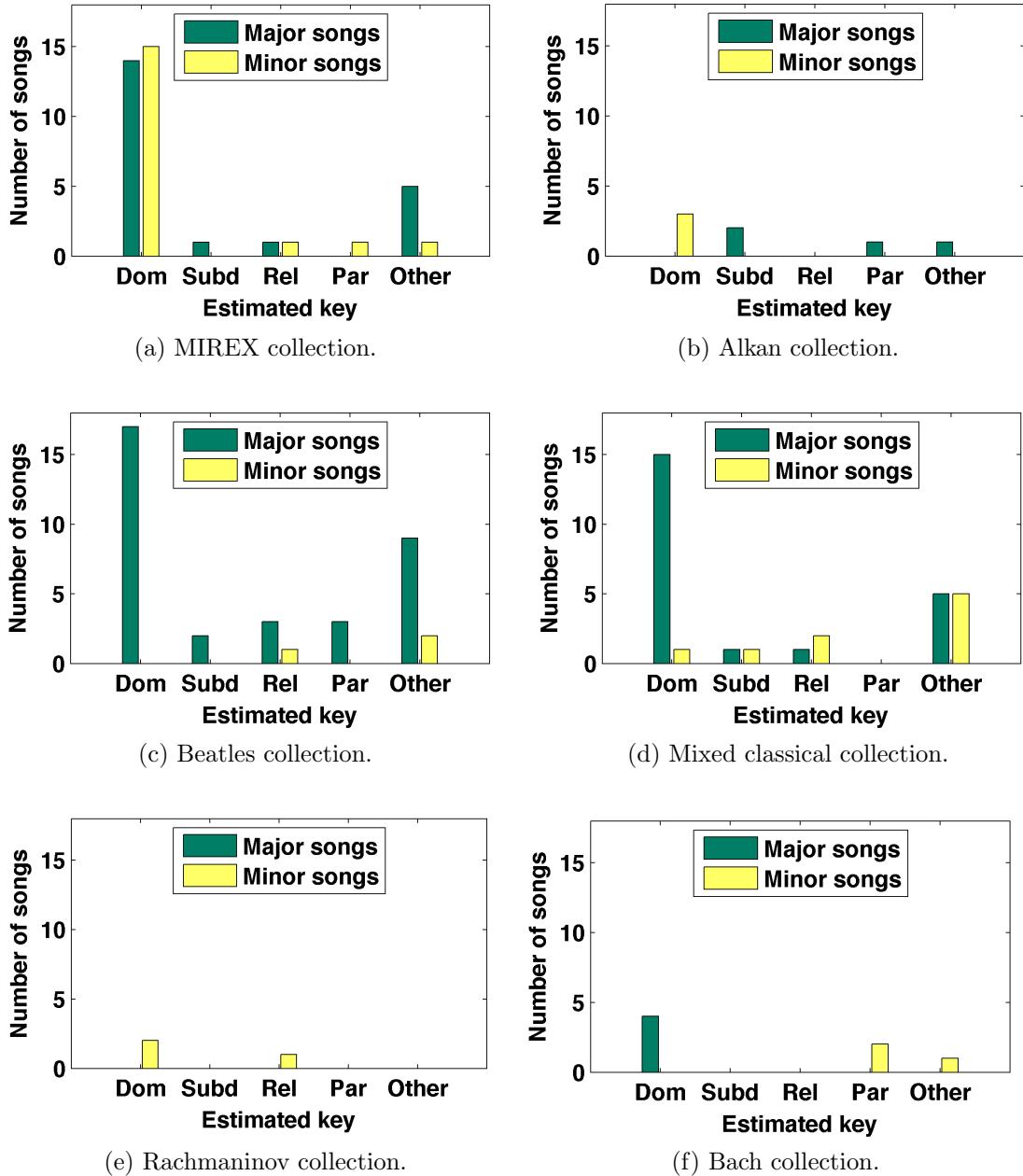


Figure 4.12: Distribution of errors with respect to the correct key for each test collection, when the HMM with a continuous observation probability density function is used. Dom = dominant, Subd = subdominant, Rel = relative, Par = parallel.

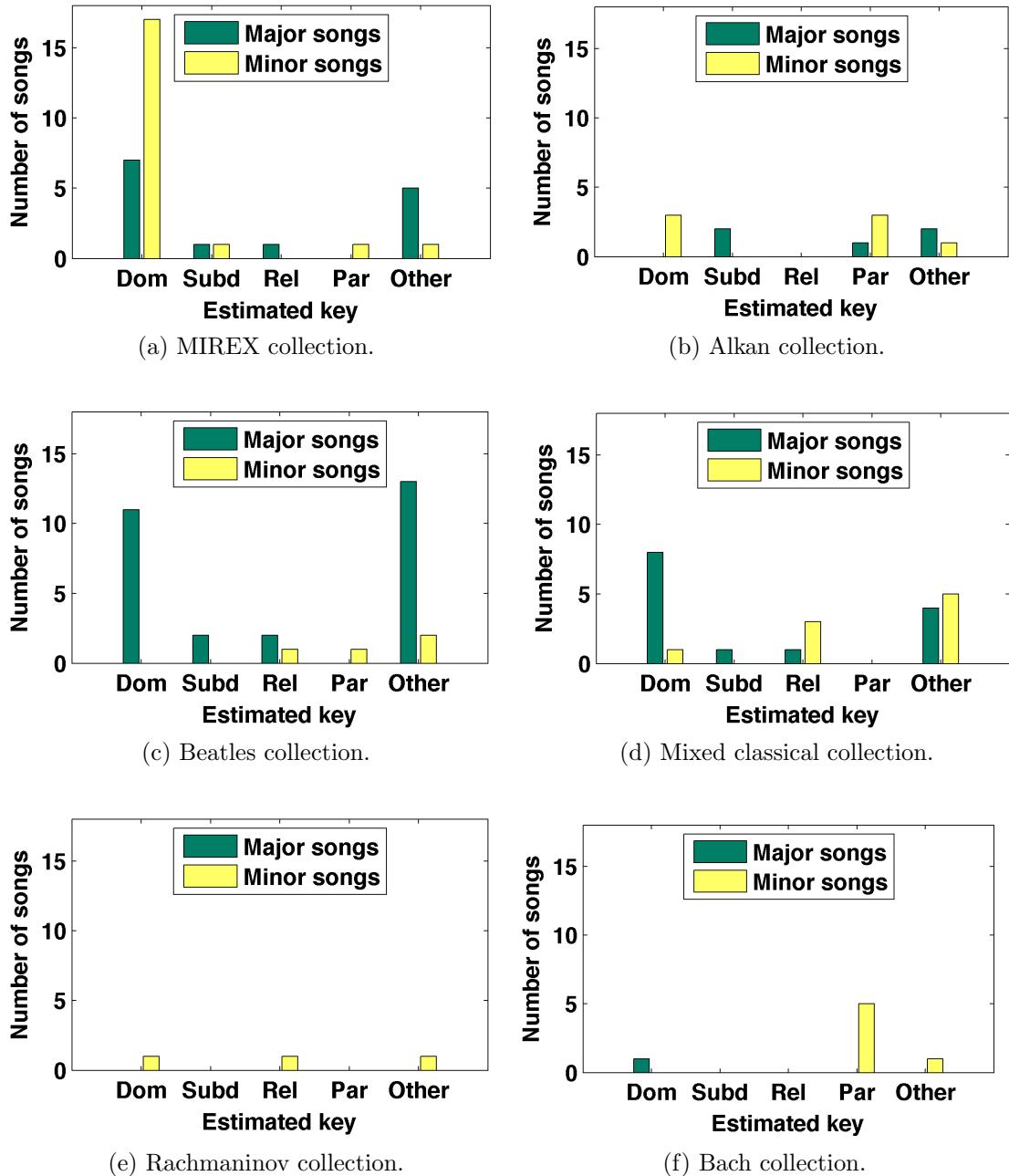


Figure 4.13: Distribution of errors with respect to the correct key for each test collection, when the HMM with a continuous observation probability density function and adapted mixture weights to reduce dominant errors is used. Dom = dominant, Subd = subdominant, Rel = relative, Par = parallel.

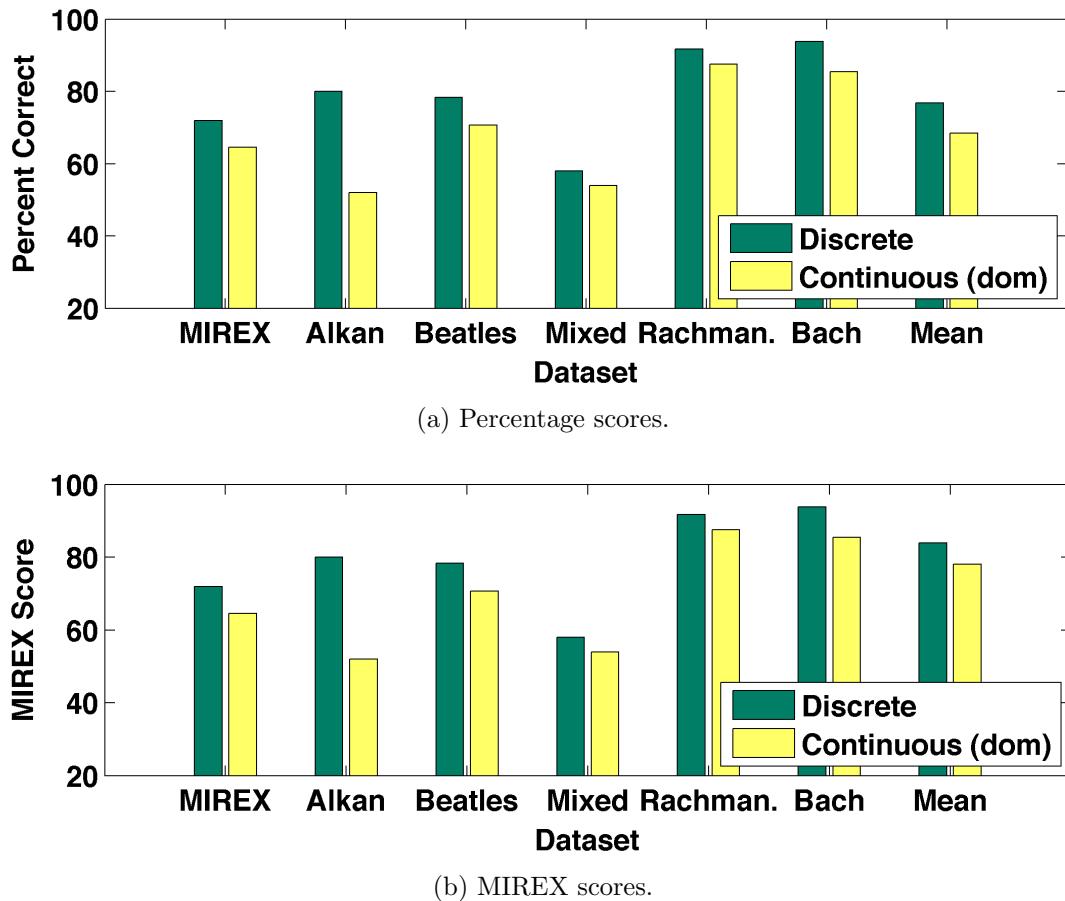


Figure 4.14: Main key estimation scores for the best performing discrete HMM and the continuous HMM with adapted mixture weights to reduce dominant key errors.

probability density function of the continuous model, although presented as a weighted mixture of chords, is nonetheless a single distribution in the chroma feature space, so the likelihood of each hidden state is determined by a pitch class distribution, with no chord estimation taking place. We have presented evidence that modelling tonality as a progression of discrete chords, as takes place when humans perform tonal analysis, is more appropriate. Hence we choose to use the discrete model for our further experiments. A more sophisticated model could include a continuous observation probability density function leading to a separate layer of discrete chords, which could act as the observations for our current discrete model that relates chords to keys. We leave these developments to future research.

4.5 Summary

In this chapter we have added a chord recognition step to our symbolic HMM to enable it to work from audio data. We used a template matching chord recognition method, and have investigated the effects of modelling upper partials in the chord templates. We found that the average key recognition performance was improved by modelling the upper partials, and we selected templates

that model 3 partials with a decay rate of 0.6 to use in our further investigations.

We also adapted the HMM to have a continuous observation probability density function, in order to avoid having to make a definite chord classification for every frame, but found that the discrete model performed better. Table 4.6 summarises the model on which we will continue our investigations.

Table 4.6: Parameters that produce the best main key estimation performance.

Parameter	Value
Observations	chord transitions
Relative state self-transition likelihood	1
Chord sampling interval	100 ms
Initialisation	Krumhansl 1
Training	Prior and transition probabilities
Upper partials modelled	3
Upper partial decay rate, s	0.6
Observation probability function	discrete

Chapter 5

DSP Parameters and Preprocessing

We have presented a method for tonality estimation from audio that relies on calculation of a chromagram before any tonal analysis takes place. The process of chromagram calculation requires several parameter choices to be made, such as the frame size and frequency range, and we have so far disregarded their effects. Such parameters are often chosen in an ad hoc fashion, with efforts concentrated on higher level parameters such as template values. However, the choice of low-level DSP parameters can significantly affect the results, so in this chapter we present an investigation into their effects (section 5.1). We also measure the computational savings associated with selected parameter choices (section 5.2). An earlier version of this work is to be published in Spring 2009 [Noland and Sandler, 2009].

We find that downsampling the audio before processing is an effective way of reducing the computation time without loss of performance, since the chord recognition relies primarily on analysis of the fundamental pitches, the upper partials being less important. We choose to continue to include all of the bass down to MIDI pitch E1, since the bass notes are very important for harmony analysis and there is little computational saving when omitting these frequencies. A range of suitable hop sizes is found, from about 0.05 s to 0.2 s, and the constant-Q transform kernels are found to be very robust to thresholding, which gives a large saving in computation time, although the best performance is achieved with the most accurate kernels.

In addition we investigate the effects of applying two pre-processing stages to the audio (section 5.3). The first is beat detection, which enables the calculation of a key estimate on every musical beat, instead of using a fixed time resolution that is not related to the music. The second is transient removal, which removes the noise-like note onsets, leaving behind the steady-state part of the music. The steady-state part contains the pitch information, and so transient removal may help to isolate the part of the signal needed for harmony analysis. However, we find that use of either pre-processing algorithm leads to considerably poorer key estimation performance.

We then compare the performance of our HMM-based key estimator to a simple tone profile correlation method of global key estimation (section 5.4). We find that the correlation method gives higher scores for the MIREX, Rachmaninov and Bach test collections, and our HMM performs better on the Alkan, Beatles and mixed classical collections.

5.1 Investigation into the Effects of Varying the Low-Level DSP Parameters

We investigate the effects of varying four low level parameters, which are described in chapter 2, sections 2.3.1–2.3.4. The parameters we consider are:

- The downsampling factor. Downsampling reduces the highest frequency that can be analysed but also produces fewer samples per second to pass to the constant-Q transform module.
- The lowest constant-Q frequency. This directly affects the shortest possible frame size, resulting in shorter FFTs and a smaller constant-Q transform kernel if lower frequencies are omitted.
- The hop size. This determines the number of frames per second to pass to the constant-Q transform and the HMM.
- The sparse kernel threshold. A high threshold reduces the accuracy of the constant-Q transform, but also reduces the number of required multiplications during the transform operation by setting small values to zero.

There are many other parameters that could be investigated, including the choice of kernel window, the position of the shorter high frequency kernels within the frame (see figure 2.8(c), page 41), and countless variations in the downsampling filter design. We choose these four because they all have the potential to reduce the processing time. Processing time measurements are presented and discussed in section 5.2.

The optimisation-by-parameter-tuning procedure is complex, with most parameters affecting at least one other. Investigation of all possible combinations of parameters would lead to a prohibitively large set of experiments, so we use the parameters from chapter 4, explained on page 88, as a starting point, and as each parameter is investigated we choose the most appropriate value for our subsequent experiments, based on the performance of the key estimation model. The initial parameter settings are given in table 5.1 together with the alternative values tested.

Table 5.1: Initial parameters for the key estimation model, and alternative values tested.

Parameter	Initial value	Alternative values tested					
Downsampling factor	1	2	4	8	16	32	64
Max. freq. (Hz)	21096	10548	5274	2637	1319	659	330
Max. freq. (MIDI)	E10	E9	E8	E7	E6	E5	E4
Min. freq. (Hz)	41.2	82.4	164.8				
Min. freq. (MIDI)	E1	E2	E3				
Hop size (frames)	1/8	1/64	1/32	1/16	1/4	1/2	1
Hop size (seconds)	0.19	0.023	0.046	0.093	0.37	0.74	1.5
Sparse kernel thresh.	0.0055	0.5	0.055	0.00055	0		

5.1.1 Downsampling

All of our original audio recordings are at CD quality, with 16 bits per sample at a sampling rate of 44.1 kHz, which gives an audio bandwidth of 22.05 kHz. When the audio is downsampled by a factor of 2, the bandwidth is halved, so we lose some audible information, and might expect the performance of the key recognition algorithm to deteriorate. However, the highest frequency bands contain only upper partials of notes whose fundamental pitches are much lower, so it is likely that downsampling by a only small factor will not seriously reduce performance and so could be a good method of reducing the running time of the algorithm.

For each part of the experiment the downsampling was performed on the original audio in one stage, so the downsampled audio will only contain one set of aliased components. The downsampling operation could be made more efficient by using a multistage structure [Porat, 1997, chapter 12], but even with a single stage downsampler the running time can be greatly reduced (section 5.2).

The anti-aliasing filter used is a finite impulse response (FIR) filter, designed using a Kaiser window, with a passband edge at $0.9 \times f'_s/2$ and stopband edge at f'_s , where f'_s is the new sampling rate. The passband and stopband ripple parameters are both 0.001, relative to a desired passband gain of 1. The magnitude response of the anti-aliasing filter for a downsampling factor of 2 is shown in figure 5.1. The stopband attenuation is 60 dB, less than the 96 dB range possible with 16 bit audio. However, it is desirable for the attenuation to be as small as possible without the aliased components becoming problematic, because less attenuation leads to a shorter filter, which takes less time to run and produces less smearing in the time domain. Results show that 60 dB is still sufficient to achieve good key recognition performance. We may be able to achieve better efficiency or better key estimation by using a different downsampling filter, but leave this experiment for future studies.

Figure 5.2 shows the percentage and MIREX scores (see section 2.6 in chapter 2 for a descrip-

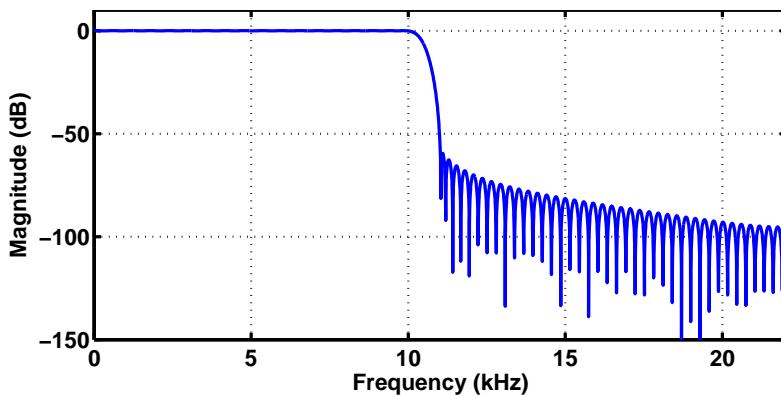
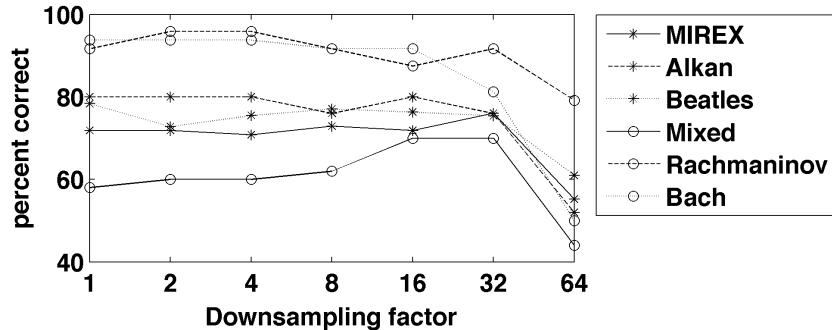


Figure 5.1: Magnitude response of the anti-aliasing filter for a downsampling factor of 2.

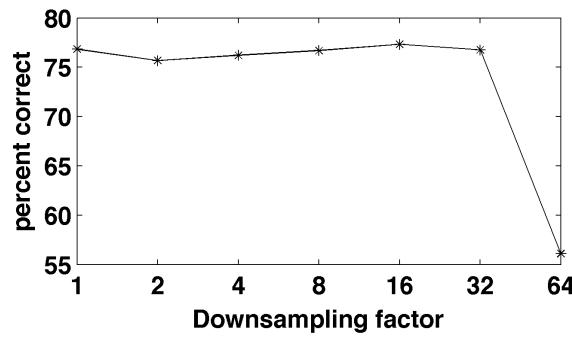
tion of the MIREX score) for the individual test collections, and the mean scores over all six test collections, for different downsampling factors. We see that there is in fact a small improvement in performance when downsampling, with a factor of 16 giving on average the best percentage and MIREX scores. Even downsampling by a factor of 32, with a corresponding highest frequency of 689 Hz (pitch E5, a major tenth above middle C) gave comparatively good key recognition performance, and it is not until we downsample by 64, cutting out all frequencies above 345 Hz (pitch E4), that performance seriously deteriorates.

These results indicate that only the fundamental pitches are important for key recognition, and the upper partials that occupy the higher frequency bands give us no additional help. It is only when the downsampling causes many of the fundamental pitches to be excluded from the analysis that key recognition becomes difficult. Restricting the frequency range to E5 still includes most of the fundamental pitches, and it seems that the missing higher notes are few enough in number for the harmony to remain largely the same. Key estimation is therefore unaffected in general, as might be expected for human analysis. However, restricting the note range to below E4 removes many more notes that are important to the harmony, leading to more chord estimation errors and hence more key estimation errors.

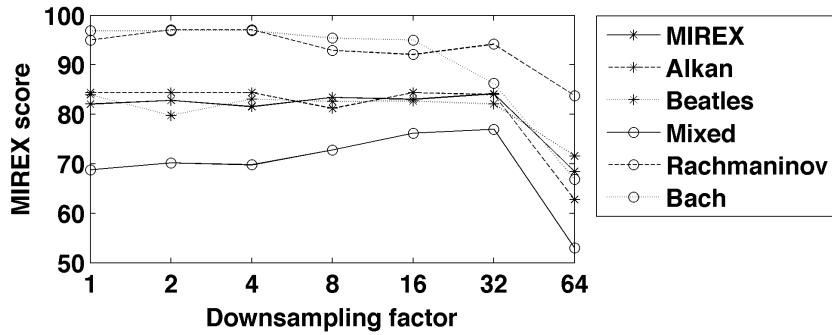
Downsampling was particularly beneficial for the mixed classical collection. Figure 5.3 shows the distribution of errors according to their relationship to the correct key for each dataset for the case with no downsampling, and figure 5.4 shows the same information for the case when the audio is downsampled by a factor of 16. We see that for the mixed classical collection the improvements when downsampling lie in the dominant and parallel error categories for pieces in a major key, and the relative major error category for pieces in a minor key, although it is difficult to draw any general conclusions from the distribution with such a small number of errors in each category. Separating the errors by composer yields no further information: the correct key was found after downsampling but not for the full sample rate version for one piece by each of Scarlatti, Vivaldi, Haydn, Schubert, Brahms and Shostakovich: composers from a wide range



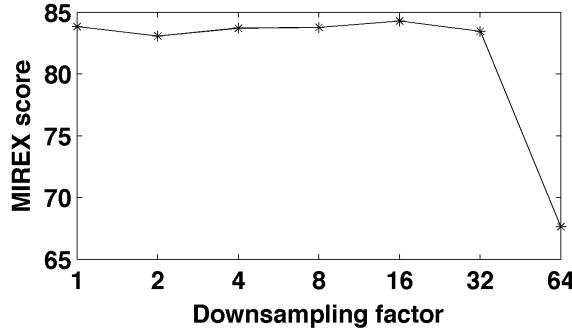
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 5.2: Main key estimation scores when using a downsampling factor of between 1 (no downsampling) and 64.

of musical periods. Four of the pieces were played on a piano, one was a guitar concerto with string orchestra and continuo, and the other was modern orchestra with full woodwind, brass and percussion: again a wide range of styles. We would require a larger test collection to better understand this result.

Although a downsampling factor of 16 gave the best average performance, less severe downsampling was preferred for the Rachmaninov and Bach test sets. The difference is however only three pieces across both test sets.

Since we see some improvement when downsampling the audio, we believe that the aliased components are at a low enough level to have no effect on performance. The loss of timing resolution caused by downsampling is also not of importance for this application. Even with a downsampling factor of 64, giving a sampling rate of 689 Hz, the samples are 0.0015 s apart: a negligible period in comparison to the shortest chord length which would be in the region of 0.3 s (1 chord per beat at 200 beats per minute).

For subsequent experiments we use a downsampling factor of 16, which restricts the frequency range to below 1378 Hz, or pitch E6, a little over 2 octaves above middle C. We continue to use the model of the first 3 upper partials in the chord templates as developed in chapter 4. This is still a valid approach when an upper limit is placed on the frequency range and some of the upper partials are missing, because the upper partials of the lower notes will still overlap the fundamental pitches of the higher notes.

5.1.2 Minimum Frequency

For a constant centre-frequency-to-bandwidth ratio at zero frequency an infinite bandwidth would be required, so the constant-Q transform requires that a minimum analysis frequency is set that is greater than zero. So far we have used 41.2 Hz as our lowest frequency, corresponding to the lowest note on a standard bass guitar. We now investigate how key estimation is affected by raising the minimum frequency. This will exclude some of the bass notes, but will also eventually allow the use of shorter frames to give better time resolution, and include fewer chords per frame. For this experiment however we hold the frame length constant, at 4096 samples, about 1.5 s, and simply exclude the lower octaves from the chromagram calculation.

It should be noted that the time-domain width of the constant-Q transform kernel for a given frequency is constant, no matter what the frame size. For the Fourier transform a long frame leads to averaging of the energy across a long period, but for the constant-Q transform increasing the frame size is equivalent to using a larger hop size, with the increased averaging only occurring for the additional low frequencies. This means that a long frame will smooth bass notes together, but sample the higher notes less often.

Figure 5.5 shows the percentage and MIREX scores for three different minimum constant-Q

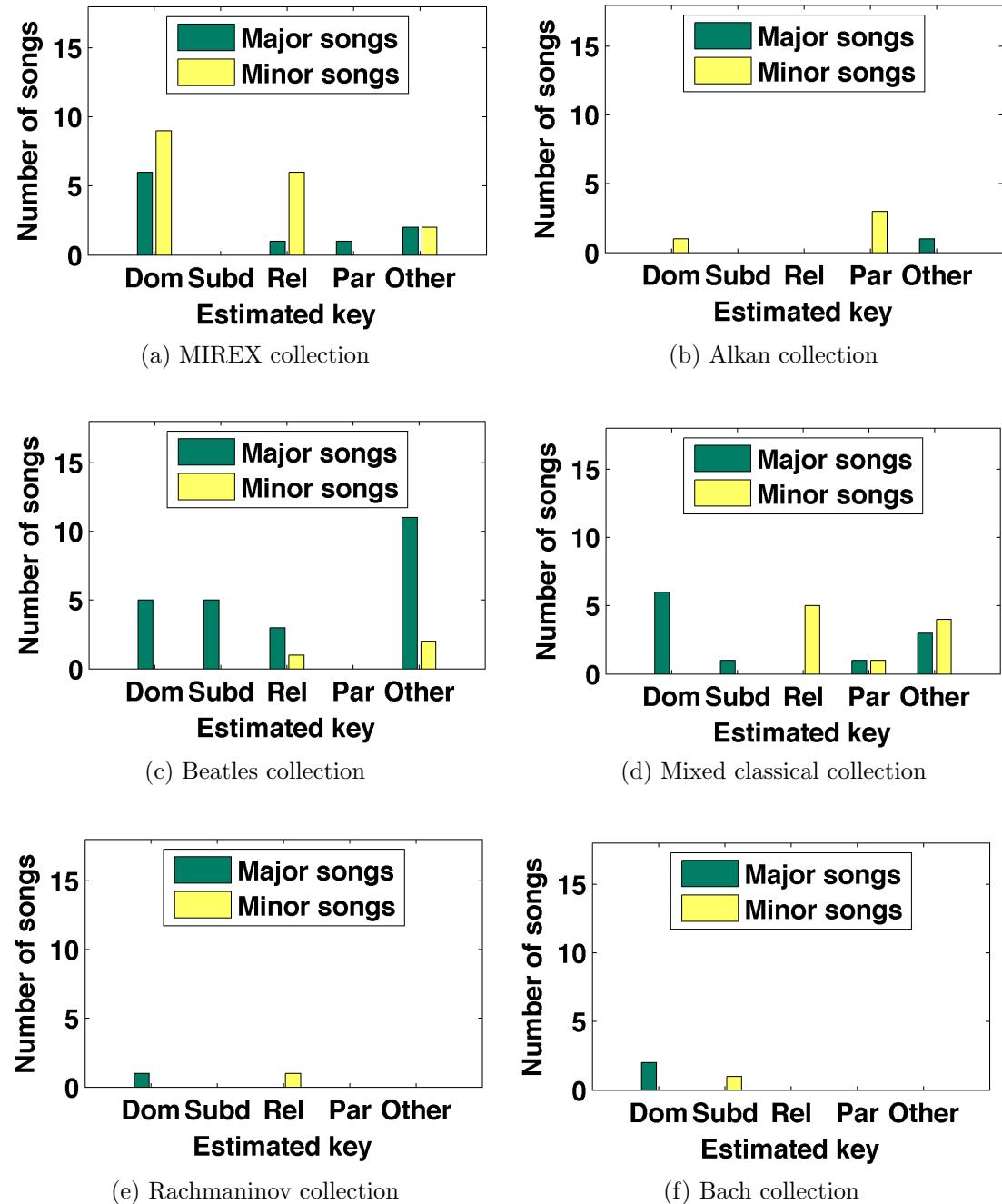


Figure 5.3: Distribution of errors with respect to the correct key for each test collection, when the audio is not downsampled. Dom = dominant, Subd = subdominant, Rel = relative, Par = parallel.

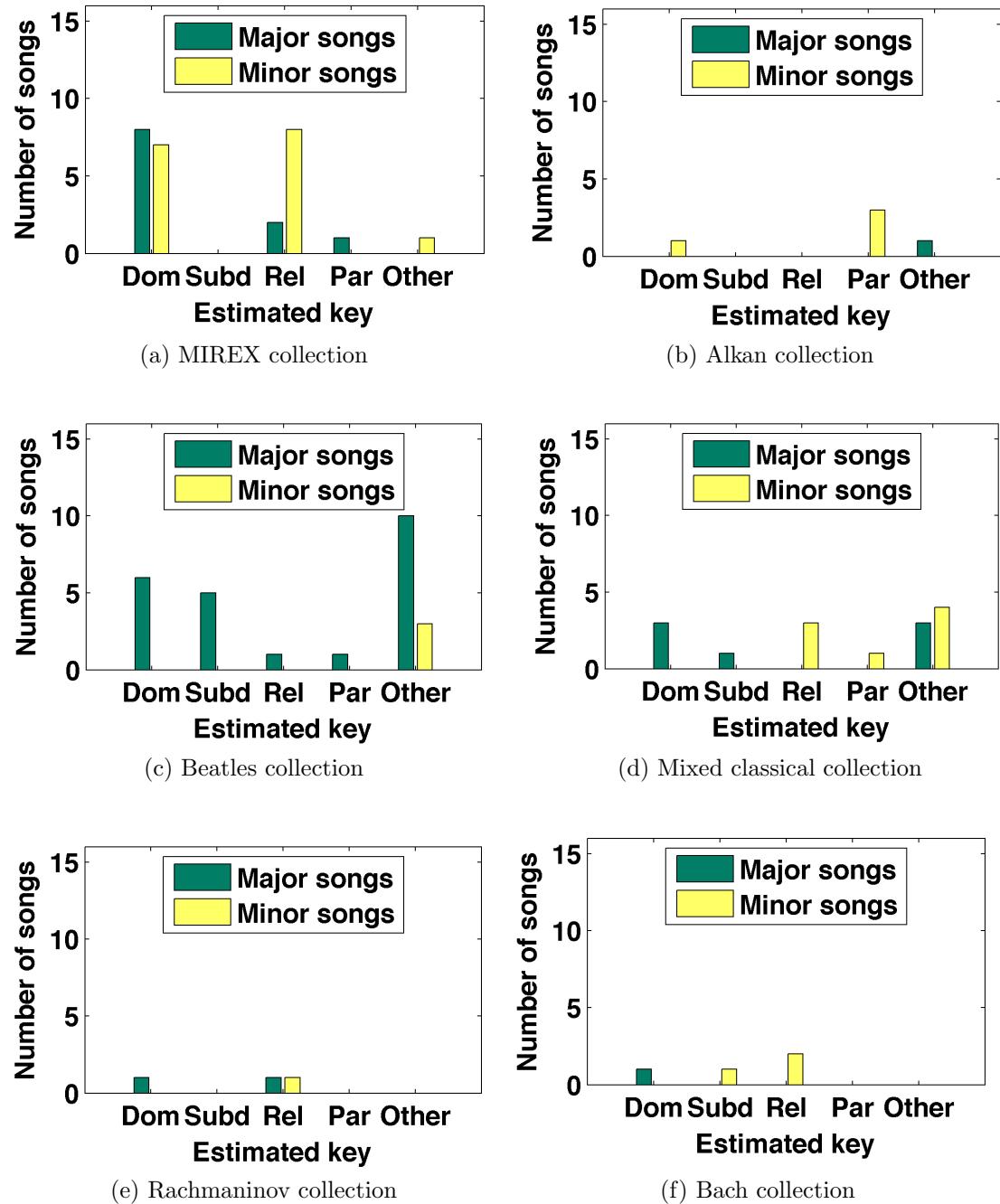
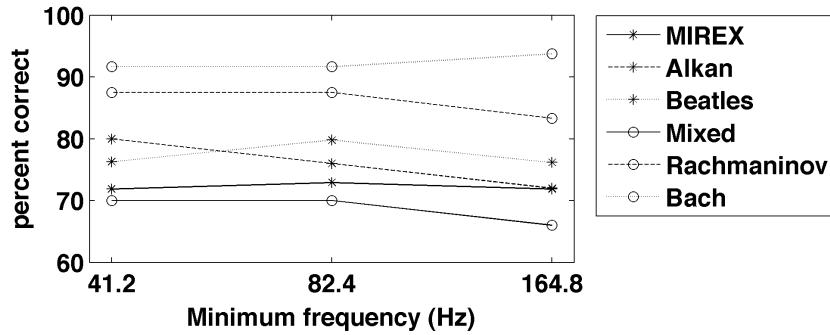
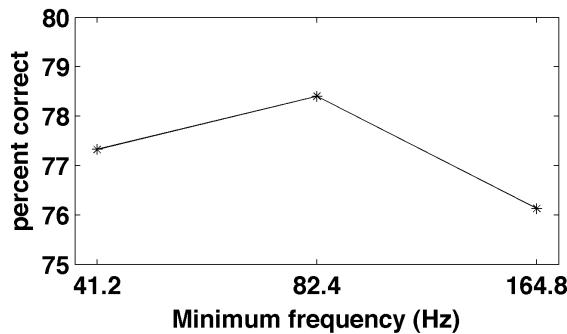


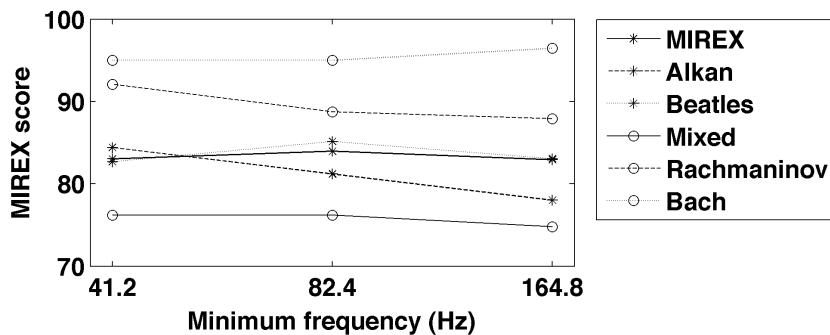
Figure 5.4: Distribution of errors with respect to the correct key for each test collection, when the audio is downsampled by a factor of 16. Dom = dominant, Subd = subdominant, Rel = relative, Par = parallel.



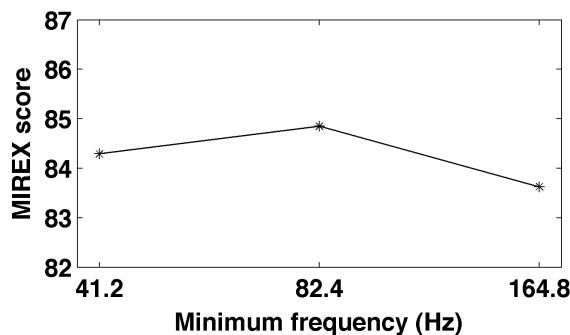
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 5.5: Main key estimation scores when using a minimum constant-Q frequency of between 41.2 Hz and 164.8 Hz. The frame size was held constant.

frequencies. We see that the differences in performance are very small, but there is nonetheless a peak in the average performance when the minimum frequency is set to 82.4 Hz.

The scores for all test collections were lower with a minimum frequency of 164.8 Hz than with 82.4 Hz. This degradation was expected since a limit of 164.8 Hz is less than an octave below middle C and so will exclude a significant number of bass notes, which are particularly important for harmony analysis.

It is more surprising that excluding only the lowest octave, between 41.2 (E1) and 82.4 Hz (E2), has improved performance. The improvement is almost entirely due to differences in performance on the Beatles test collection, which has the greatest influence on the average values since it contains the greatest number of tracks, with a small improvement in performance on the MIREX collection as well. In order to better understand these results we look in more detail at the particular tracks that cause the differences. Table 5.2 on page 117 shows the details.

Of the pieces in the lower part of the table, whose keys have been incorrectly estimated when the lowest octave is included, we see that the errors are mainly due to ambiguities in the music: either the music contains substantial sections in another key, or it is modal and so doesn't fit into our major-minor model. For *You Won't See Me* and *I Want To Tell You* a tendency towards the dominant using either chord V-of-V or the root note of V-of-V has tipped the estimation incorrectly to the dominant key. The choice of the subdominant key for *Yesterday* is more difficult to explain, however turning to the chord annotations we see that the subdominant chord is much more frequent than the dominant (19 occurrences of a B \flat chord compared to 8 of a C chord), and the main cadences on the word "yesterday" are plagal, chord IV to chord I.

We believe that the incorrect key estimates resulting from including the lowest octave are due to the model incorrectly decoding the harmonies, not due to smearing of the bass notes as a result of longer frames or to the addition of extraneous noise in this frequency band. Hence we conclude that it is important to include the lowest notes in the analysis, and higher scores should be achieved by improving the model, not by discounting frequencies that contain useful information about the harmony. We continue to use a minimum frequency of 41.2 Hz.

5.1.3 Hop Size

Reducing the hop size allows us to increase the number of frames per second without reducing the frame size, and therefore without excluding any of the lower frequencies. The most appropriate hop size is closely linked to the transition probabilities of the HMM, which encode the likelihood of staying in the same key from one frame to the next: for a smaller hop size the likelihood of changing key will be smaller. The most appropriate hop size is also dependent on the tempo and rate of harmonic change of the music itself, so we cannot expect to find a single value that is optimal for all music, or even for music within the same genre or by the same artist. The best

Table 5.2: Details of tracks for which key estimation is affected by excluding the lowest octave, between 41.2 and 82.4 Hz.

Track		Collection	Correct key	Key est. with lowest octave	Key est. without lowest octave	Brief analysis
Excerpt 33	MIREX	F major	F major	C major	Moves to C for a short time, and D minor for a short time.	
Not A Second Time	Beatles	G major	G major	E minor	Strong feel of E minor throughout.	
When I'm Sixty-Four	Beatles	D \flat major	D \flat major	A \flat major	Clearly in D \flat major.	
Excerpt 56	MIREX	B \flat minor	D \flat major	B \flat minor	Contains 2 phrases that begin in the major, and the final section is entirely major.	
Excerpt 81	MIREX	F minor	A \flat major	F minor	Ends in the major mode. Very chromatic in the upper parts.	
A Taste of Honey	Beatles	F \sharp minor	E major	F \sharp minor	Strong Dorian feel with many E naturals.	
I Wanna Be Your Man	Beatles	E major	A major	E major	Bluesy with many D naturals in the bass.	
Yesterday	Beatles	F major	B \flat major	F major	This leans much more to the relative minor than the subdominant.	
Norwegian Wood	Beatles	E major	G \sharp minor	E major	Strong Mixolydian feel. Main phrase over E drone. Bridge in E minor.	
You Won't See Me	Beatles	A major	E major	A major	Clearly in A but with lots of chord V-of-V.	
I Want To Tell You	Beatles	A major	E major	A major	Clearly in A with lots of B in the bass.	

we can achieve from a single, constant hop size is one that works well on average, and reasonably well for any kind of music. We would expect this value to be roughly 500 ms or less, based on our results from chapter 3 for varying the chord sampling interval (see figure 3.14 on page 77). With a downsampling factor of 16 and a frame size of 1.5 seconds to include frequencies down to 41.2 Hz, this corresponds to a hop size of no more than 1/3 of a frame.

Figure 5.6 shows the percentage and MIREX scores for all of our test collections when using hop sizes between 1/64 and 1 frame. The results show the same preference for short hop sizes as the discrete model, up to 1/32 of a frame (about 46 ms), after which the performance scores reduce again. The hop sizes giving peak performance for the individual music collections are between 1/32 and 1/8 of a frame, with variations most likely due to tempo differences in the music. The longer hop sizes are less suitable for any of the music collections, presumably because the time resolution is too poor to identify the chords accurately. We choose a hop size of 1/8 of a frame to continue our experiments, since it gave the highest percentage score, and within the range of suitable hop sizes it produces the smallest number of frames to process per second so will give the shortest computation time. In section 5.3.1 we also investigate the effects of using a hop size that is determined by the tempo of the music.

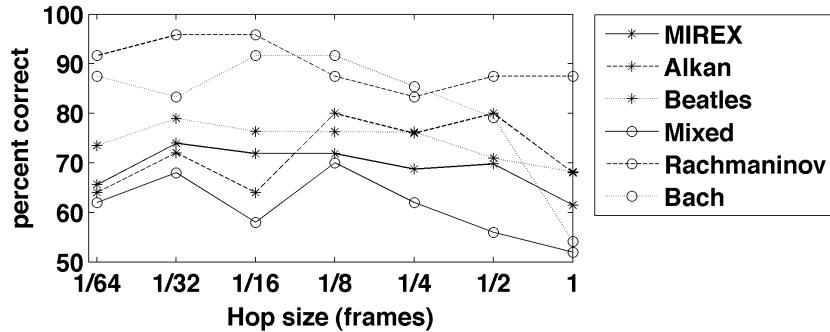
5.1.4 Sparse Kernel Threshold

In chapter 2, page 41, we described the efficient algorithm that we use for the constant-Q transform, including thresholding of the spectral transform kernels. We now investigate the effects of varying the threshold. A range of thresholds are tested, from no thresholding at all to a threshold of 0.5 which removes all but the highest peak of the kernel (peak value 0.54). Figure 5.7 shows the accurate spectral kernel for 1760 Hz and the same kernel with a threshold of 0.5 applied.

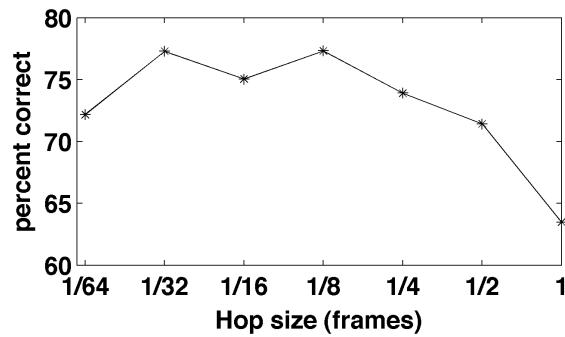
Figure 5.8 shows the percentage and MIREX scores for all of our test collections for different sparse kernel thresholds. As might be expected, the best performance is achieved with no thresholding, when the transform kernels are accurate to 24-bit floating point precision. However, the results are still good with thresholds of 0.0055 and 0.055 applied. There is a drop in the scores for the highest threshold of 0.5, but given that the average percentage score is still over 70 % we conclude that the constant-Q transform is very robust to thresholding.

Thresholds of both 0.0055 and 0.055 are sufficient to set all of the side-lobes to zero, so either would offer significant computational savings. The resulting difference in main lobe width between the two is approximately 2 cents, so they are essentially the same. For the MIREX, Beatles and mixed classical collections the performance was actually better with the higher threshold of 0.055.

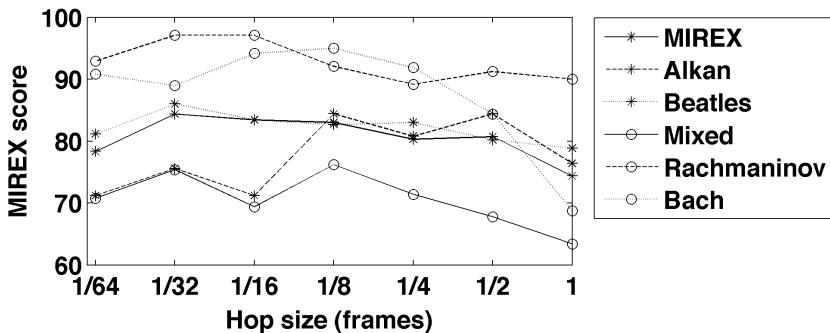
We have established that thresholding the transform kernels is a valid means of increasing computational efficiency (see section 5.2 for computation time figures). However, our current goal is to achieve the best key recognition performance, so for subsequent experiments we omit



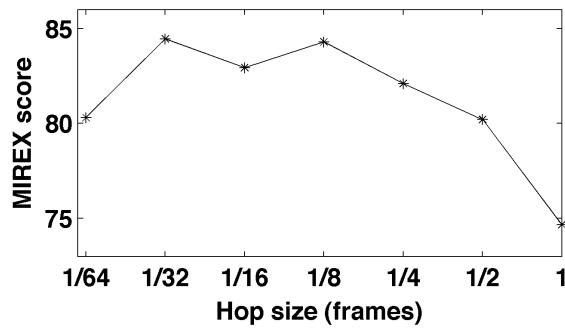
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 5.6: Main key estimation scores when using a hop size of between 1/64th of a frame (0.023 s) and a whole frame (1.5 s).

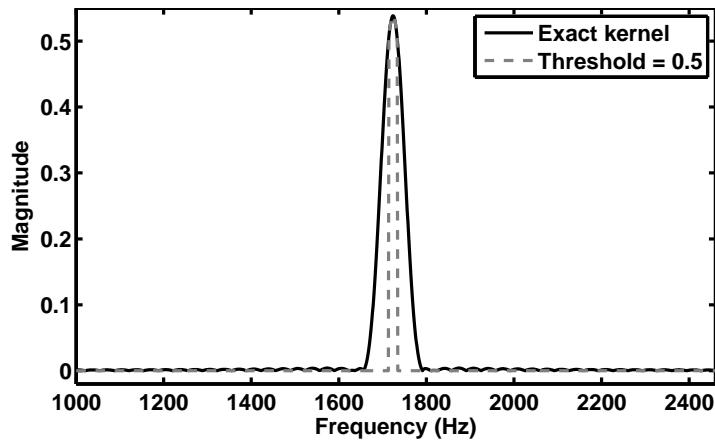


Figure 5.7: Accurate spectral kernel (solid) and a kernel with a threshold of 0.5 applied (dashed), for an analysis frequency of 1760 Hz.

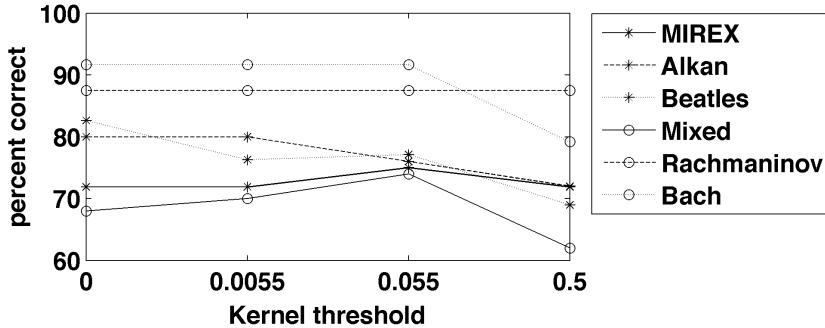
the thresholding operation and use accurate transform kernels.

5.2 Computation Time Comparison

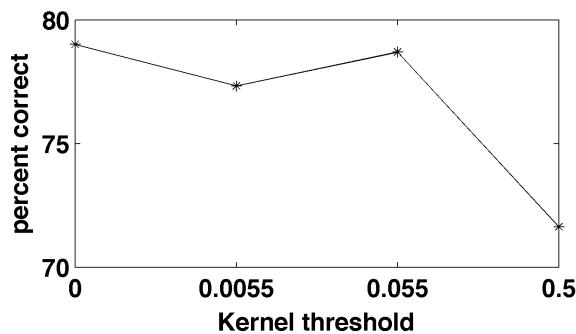
The experiments described in section 5.1 have been conducted using different computers with different specifications, so it is not possible to compare all of the computation times. However, we repeated selected experiments in a controlled fashion in order to demonstrate the computational savings possible. These repeated experiments were conducted on a PC with an Intel Xeon 3.0 GHz CPU (single core) and 2.5 GB of RAM, running Windows Server 2003, using MATLAB R2008a. The results are given in table 5.3 on page 122. The times are given in minutes, which can be considered approximately equivalent to CPU minutes since the computer was not performing any other tasks while the experiments were conducted.

We have separated the total computation times into the time taken to calculate the chromagram from digital audio, and the time taken to perform the key estimation from the chromagram. The greatest computational load is in the chroma feature extraction process, so this is where the greatest savings can be made.

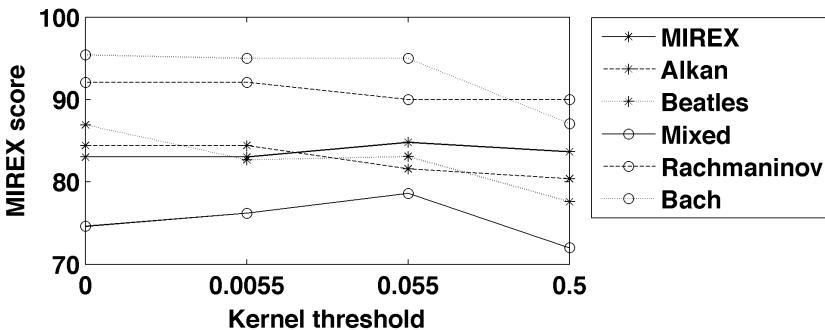
Parameter sets a and b in table 5.3 differ in their downsampling factors, set a with no downsampling and set b with a factor of 16. Downsampling by a factor of 16 has produced a saving of 16 % over the 522 minutes required to calculate a chromagram with no downsampling. One might expect a much greater saving, however we see later that the file reading, which always takes place at the original sampling rate and so does not vary with downsampling, takes the greatest proportion of the chromagram calculation time and so limits the time savings possible. In addition, the downsampling operation itself takes some processing time, and could be implemented more efficiently by using a multistage approach, leading to further computational savings [Porat, 1997, chapter 12]. There is no difference in the key estimation time when downsampling, since



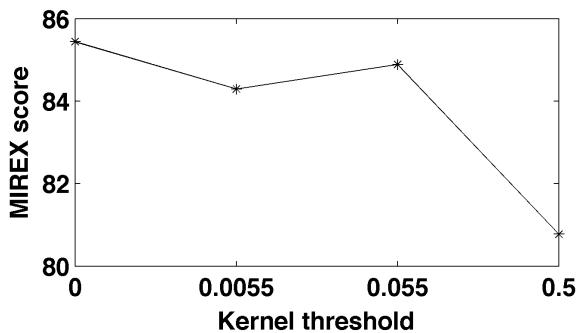
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 5.8: Main key estimation scores when using a constant-Q kernel threshold of between 0 (no thresholding) and 0.5.

Table 5.3: Computation times for all 6 test collections together, for extraction of chroma features from audio and main key estimation from chroma features. Five different sets of parameter values, *a–e*, were tested.

Parameter	Parameter values				
	set <i>a</i>	set <i>b</i>	set <i>c</i>	set <i>d</i>	set <i>e</i>
Downsampling factor	1	16	16	16	16
Maximum frequency (Hz)	21096	1319	1319	1319	1319
Maximim pitch (MIDI)	E10	E6	E6	E6	E6
Minimum frequency (Hz)	41.2	41.2	82.4	41.2	41.2
Minimum pitch (MIDI)	E1	E1	E2	E1	E1
Frame length (s)	1.5	1.5	0.74	1.5	1.5
Hop size (frames)	1/8	1/8	1/4	1/4	1/8
Hop size (s)	0.19	0.19	0.19	0.37	0.19
Kernel threshold	0.0055	0.0055	0.0055	0.0055	0
Chroma feature extraction time (minutes)	522	439	413	391	875
Main key estimation time (minutes)	36	36	36	20	36

the same number of chroma frames are produced.

The computation times for parameter sets *b* and *c* allow us to find the savings associated with missing out the lowest octave from our analysis. Here we have used shorter frames when missing out the lowest octave, but with the same hop size in seconds and so we maintain the same final number of frames. Using the shorter window length has produced a small saving in chromagram calculation time, a reduction of 6 % compared to the 439 minutes for a minimum frequency of 41.2 Hz. This saving is due to the need for fewer samples for each FFT calculation because of the shorter frame length, and fewer constant-Q values to calculate because of the higher minimum frequency. The saving is only small, however, so our decision to continue to include the lowest octave in calculations in order to reduce the likelihood of excluding some bass notes will not severely increase the running time. There is no reduction in the final number of frames, so the key estimation time is unchanged when the minimum frequency is raised.

The computation time saved by using a larger hop size can be found by comparing parameter sets *b* and *d*. Doubling the hop size to 1/4 of a frame (set *d*) has the effect of halving the final number of frames, and led to a chromagram calculation time of 391 minutes, a saving of 11 % over the version with a hop size of 1/8th of a frame (set *b*). Table 5.4 shows an approximate breakdown of the chroma feature extraction times for parameter sets *b* and *d*. We see that the saving comes partly from a small reduction in the FFT and constant-Q calculations, but mainly from a halving of the time required to tune the chromagram frames, which took about 13 % of the

Table 5.4: Breakdown of chroma feature extraction times for parameter sets b and d from table 5.3. Values are approximate, calculated from the time taken to process one piece only.

Operation	Feature extraction time			
	set b		set d	
	seconds	%	seconds	%
Reading audio samples from wave file	16.39	56	16.68	60
Downsampling	8.62	29	8.66	31
Calculating parameters	< 0.01	< 0.1	< 0.01	< 0.1
FFT	0.20	0.7	0.13	0.5
Constant-Q	0.23	0.8	0.17	0.6
Constant-Q to chromagram mapping	0.09	0.3	0.05	0.2
Tuning	3.93	13	1.98	7
Program control overheads	< 0.01	< 0.1	< 0.01	< 0.1

total chroma feature extraction time for parameter set b . The savings possible from increasing the hop size are limited because the two most costly operations, reading in the audio samples from a separate computer (56 % of the total feature extraction time for set b) and downsampling (29 % of the total feature extraction time for set b), are both independent of the final number of frames.

The hop size is the only parameter measured in this experiment that does affect the final number of frames, and we see a corresponding saving of 16 minutes in the key estimation time, or 44 %. The running time for almost all parts of the key estimation algorithm is dependent on the number of frames, with some control overheads, so this figure is approaching the 50 % we would see for a directly proportional relationship between the number of frames and running time.

Lastly, we measure the computational saving obtained by applying a threshold to the constant-Q spectral kernels. The algorithm using parameter set e does not perform any thresholding, and its computation times should be compared to the times for parameter set b , which has a kernel threshold of 0.0055. Applying the kernel threshold has produced the greatest reduction in chromagram computation time, from 875 minutes to 439, a saving of almost 50 %. Although we have chosen to continue without any thresholding in order to achieve the best key estimation performance, applying a threshold is clearly a good way of reducing the computation time with minimal performance deterioration for any application where time is important. The kernel threshold has no effect on the final number of frames, so the key estimation time is unaffected.

5.3 Investigation into the Effects of Using Two Preprocessing Algorithms: Beat Tracking and Transient Removal

We now investigate whether two preprocessing algorithms, designed to isolate the features that are useful for harmony analysis, can produce better key estimation performance when applied to

the audio data before the chromagram calculation.

5.3.1 Beat Tracking

In our experiments to find an appropriate hop size, described in section 5.1.3, we found that a wide range of hop sizes could be suitable. We believe that this wide range is a result of differences in musical style, in particular differences in tempo, between pieces in the music collections. The most appropriate hop size is determined by the rate of change of chords, which is likely to be strongly related to the tempo.

We now investigate the effects of using a hop size that is determined by the tempo of the music, which should have the effect of normalising the frame rate to the musical tempo. Assuming that chord changes usually occur on musical beats, a hop size of one beat with a frame size of less than one beat will be more likely to give frames that contain only one chord than a hop size that is unrelated to the music, which will lead to frames that span two or more chords. Including only one chord per frame should lead to more accurate chord recognition, and therefore better key estimation. This approach has been shown to improve chord recognition results by about 8% for Bello and Pickens' HMM-based chord estimation model [Bello and Pickens, 2005].

We first apply an automatic beat tracking algorithm to the audio data to give us sample-accurate estimates for the beat locations. We use the method described by Davies and Plumley [2007b], which uses a two-state model: a *general state* estimates the beat period based on autocorrelation in a note onset detection function, then if the beat period is stable control passes to a *context-dependent state* to enforce continuity in the estimated tempo. Using the two states in combination means that the beat tracker is able to follow sudden changes in tempo, but does not suffer badly from the common errors of switching to off-beats or switching to half or double tempo. The authors report accuracy equivalent to the state-of-the-art, with reduced computational complexity.

Having located the beat positions, we set the hop size to one beat. This means that the hop size can vary within one piece, but is constant relative to the estimated beat period. There is a conflict between including the lowest possible frequencies for analysis and keeping the frames shorter than one beat to avoid smearing of the bassline across beats, since most music has a tempo that is much faster than $1/\text{framesize} = 1/1.5$ beats per second, or 40 beats per minute. We choose to analyse only one beat of audio, and pad the beat with zeros to give the correct number of samples, with the audio in the middle of the frame. We apply a Hamming window of one beat length to the non-zero part of the frame in order to avoid block-end effects at the edge of the beat.

Reducing the number of audio samples analysed in one frame is equivalent to cropping the low frequency transform kernels, resulting in a lower Q and therefore greater spectral smearing

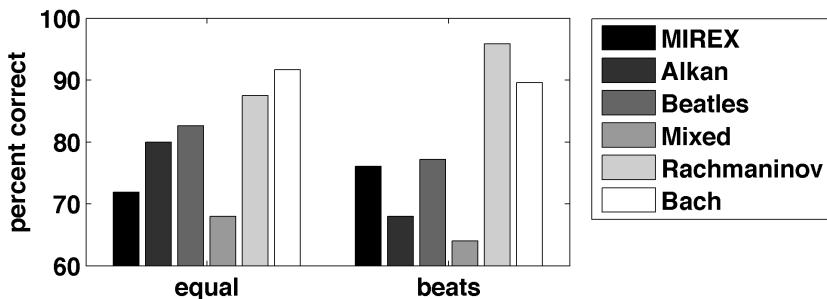
at the bass end. This means we no longer have a true constant-Q transform. However, we believe that the benefits of isolating single chords will overshadow errors due to lower spectral resolution for bass frequencies. For the few cases where the beat length is longer than the desired frame length we use the middle of the beat for analysis.

Figure 5.9 shows that using beat-synchronous frames results in poorer performance for most of the music collections. Listening to the tracks for which the correct key is found with equal-length frames but the incorrect key for beat-length frames, together with an audio rendition of the beat tracker, showed that the beat tracker is not finding the beats correctly. The authors of the beat tracking algorithm report accuracy of only 26.9 % for classical music [Davies and Plumley, 2007b], which accounts for 5 of our 6 test sets. Accuracy reported for two other beat-tracking algorithms tested was little better than 40 % for classical music so we would not expect a change of beat tracker to make a big difference. Beat tracking results for rock music were much better, at 77.9 %, however we find that even with the Beatles test collection our key recognition performance is worse when using the beat tracker. Similar findings are reported by Bello in a recent investigation into using beat-synchronous chroma features for chord estimation as part of a cover song retrieval system [Bello, 2007]. Hence we conclude that automatic beat-tracking algorithms are not yet reliable enough to be used as a preprocessing step for harmony analysis. A more suitable approach may be to use an algorithm that specifically searches for changes in harmony to segment the audio [Harte et al., 2006, Li and Bello, 2007].

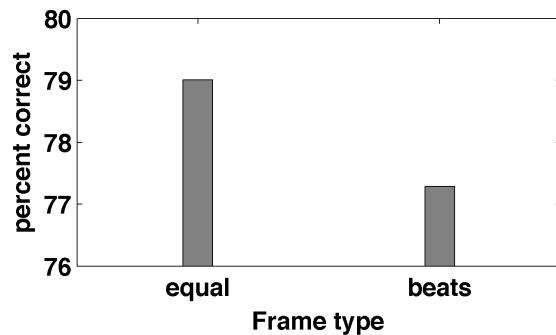
5.3.2 Transient Removal

Musical sounds can be separated into transient and steady-state parts. Transient parts occur particularly at note onsets and are noisy in character, and steady-state parts form the sustained section of musical notes, are approximately sinusoidal in character, and contain the pitch information. We hypothesise that if we discard the transient parts of the signal, thereby isolating the steady-state parts, we will see an improvement in the performance of our model. The chromagram will no longer show energy from the wideband transients, and so the harmonic peaks will be more pronounced.

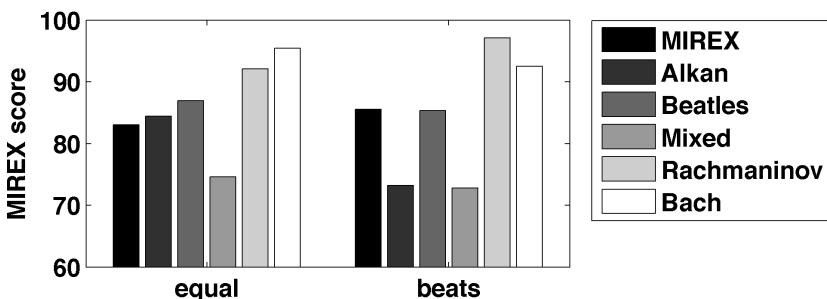
We apply a transient–steady-state separation algorithm after downsampling the audio, and then pass only the steady-state part of the signal to the constant-Q transform module. We use the separation method proposed by Duxbury et al. [2001]. Bins in a time-frequency grid are each classified as either transient or steady-state, depending on the difference in phase increment between adjacent windows for a given frequency. For a pure sinusoid at the bin frequency the phase increment is constant, so the difference in phase increment between adjacent frames is zero. Most signals will not be purely sinusoidal, so a bin is classified as steady-state if the difference in phase increment is below a certain threshold, set initially by the user. The threshold is then



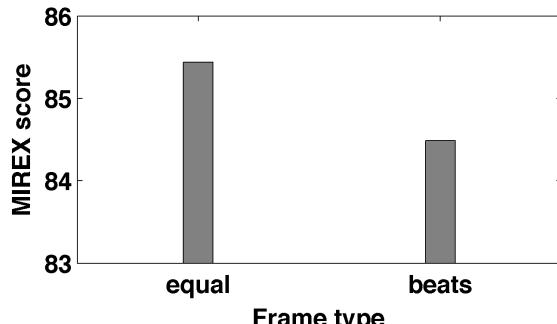
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.

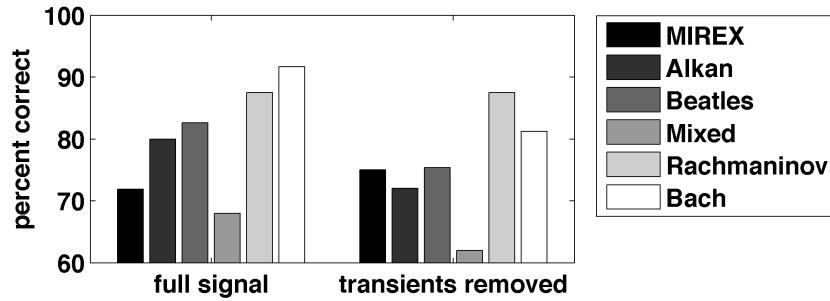


(c) MIREX scores for the separate music collections.

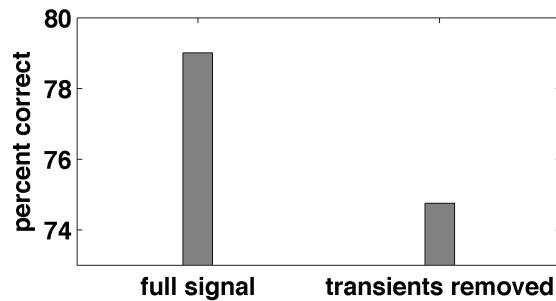


(d) MIREX scores for all of the collections together.

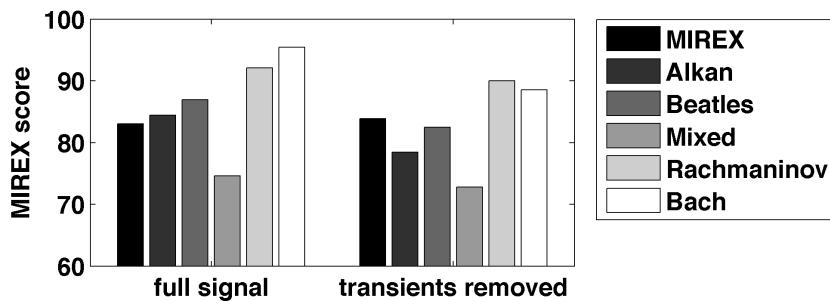
Figure 5.9: Main key estimation scores with equal-length frames and with beat-length frames.



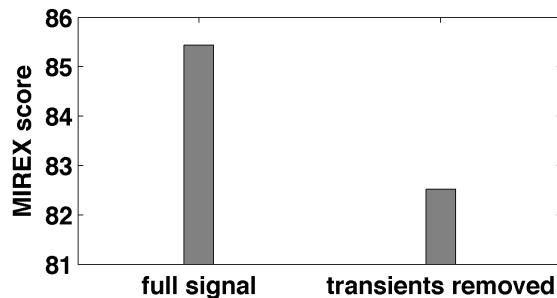
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 5.10: Main key estimation scores when using the full (downsampled) audio signal and the audio after transient removal.

adapted by the algorithm to improve detection of sinusoidal components that cross over the initial user-defined value. We set the initial threshold heuristically using informal listening tests on the separated, reconstructed outputs, and choose a value of 0.12.

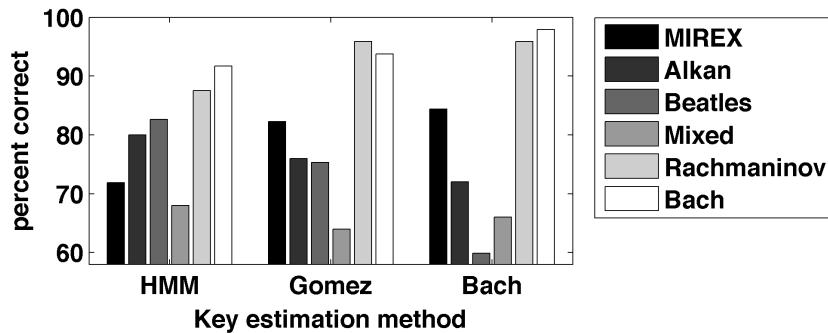
Figure 5.10 shows that automatic transient removal has improved key recognition performance only for the MIREX test collection, with no difference in the percentage score for the Rachmaninov collection and degraded performance for all other collections. The MIREX collection is the only one that is comprised of synthesised audio so it is likely to fit better into theoretical attack-sustain models. If this is the case the transient–steady-state separation algorithm would work better on the MIREX audio than on the other “natural” recordings. Transient signals are noise-like, and so add energy to all pitches equally, in effect slightly raising the noise floor. The peaks in the chromagram at note frequencies will therefore still show above the transients. Our results confirm this. It is also likely that with a downsampling factor of 16, and so an upper frequency limit of 1.3 kHz, much of the high frequency transient information has already been removed.

5.4 Comparison with a Tone Profile Correlation Method of Main Key Estimation

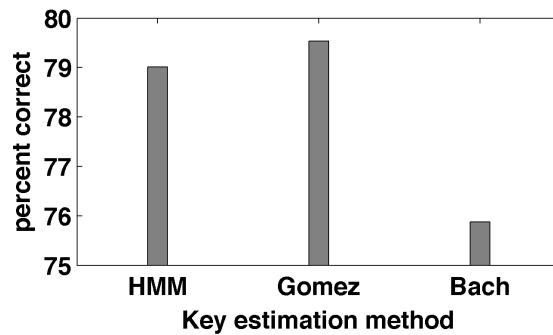
We now compare the performance of our model with a tone profile correlation approach to automatic key estimation. We take the approach described by Gómez [2004b], using her tone profile values as used in the `mirkeystrength` function in the MIR Toolbox for MATLAB [Lartillot and Toiviainen, 2007]. These are based on the Krumhansl probe tone ratings but adapted to give emphasis to notes of the tonic, dominant and subdominant triads and to take upper partials into account.

We also test tone profiles derived from the recordings in our Bach test collection, from Book I of the Well-Tempered Clavier, which we label the *Bach* profiles. They are shown in figure 2.3(c) on page 25. They were generated by calculating average chroma vectors for each Prelude or Fugue, which were then normalised to the title key. The normalised major and minor average chroma vectors were then summed to give a single major and single minor template, which can be rotated to produce a template for any major or minor key. These profiles will be specific to the note distribution used by Bach in the different tonal contexts, and to the timbre of the piano used for the recordings.

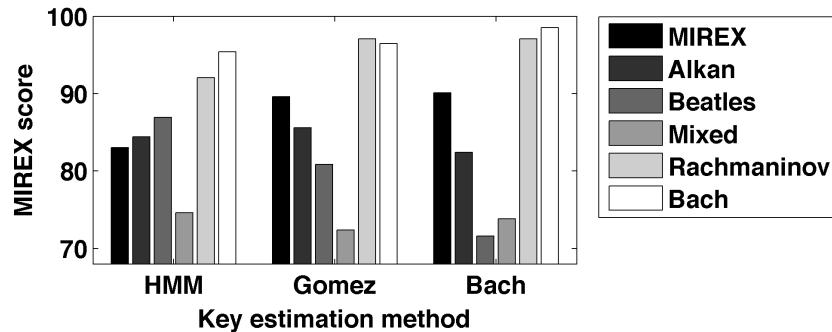
The tone profile correlation method of key estimation begins with a chromagram calculation, for which we use the same parameters as for our best performing HMM key estimator, as given in table 5.5. The correlation (see equation 2.11 on page 47) is calculated for each frame of the chromagram with each of 24 tone profiles, one representing each key. This gives a measure of the strength of each key at each time frame. To obtain an estimate of the overall most likely key, the



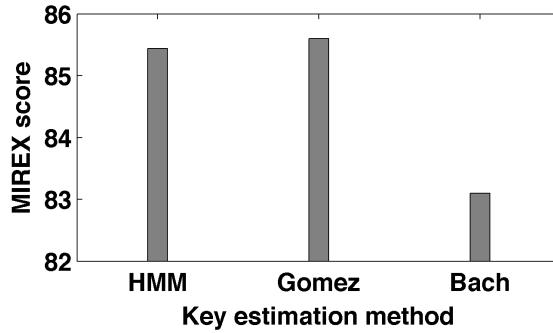
(a) Percentage scores for the separate music collections.



(b) Percentage scores for all of the collections together.



(c) MIREX scores for the separate music collections.



(d) MIREX scores for all of the collections together.

Figure 5.11: Main key estimation scores for the best-performing HMM method (see table 5.5 for the parameters) and two tone profile correlation methods, using profiles recommended by Gómez and profiles derived from recordings of Bach pieces.

key strength vectors are summed over time and the key with the highest score is taken to be the main key.

We calculate a main key estimate using the Gómez profiles and using the Bach profiles, for all the pieces in our six test collections, and compare the results to those from our best performing HMM key estimator. Figure 5.11 shows the results.

The mean percentage scores for the HMM key estimator and the profile key estimator for the Gómez profiles are only about 0.5 % apart, and the MIREX scores are even closer, with the profile method performing very slightly better. The scores that are separated by music collection reveal interesting results regarding the types of music that the different methods favour. The Gómez profile method performed better on the MIREX, Rachmaninov and Bach test collections, with the HMM outperforming the profile method on the Alkan, Beatles and mixed classical collections. The Beatles and mixed classical collections are the two which include the greatest variety of timbres and styles, and the HMM method is better at accommodating these differences. The HMM structure is able to ignore unlikely chord estimates, whereas the profile correlation method operates directly on the chromagram and includes all frames in the averaging process, so even occasional unusual harmonies exert an influence on the final key estimate.

The Bach profiles performed worse than the Gómez profiles on average, and in particular for the Beatles test collection, as might be expected. This is verification that a set of profiles that are specific to a particular type of music do not generalise well to other types, even compared to another profile correlation method. The Bach profiles did perform better than the Gómez profiles for some collections: the Bach collection, which we discount since the profiles were derived from these pieces, and the MIREX and the mixed classical collections, with equal performance on the Rachmaninov collection. For certain applications it may be possible for the user to select the most appropriate profile, but for fully automated systems the more general Gómez profiles would be more suitable.

5.5 Summary

In this chapter we have investigated the effects of varying parameters that relate to the very first stage of key estimation: the chromagram calculation. We have conducted our evaluation using measures of main key estimation accuracy. Table 5.5 summarises our best-performing model.

We found that it is possible to substantially downsample the audio signal before performing the frequency transform without loss of key estimation accuracy, and saw small improvements in key estimation performance up to a downsampling factor of 16. This results in an upper frequency limit of 1319 Hz, or MIDI pitch E6, which encompasses most of the fundamental pitches. We conclude that without more sophisticated ways of addressing upper partials in the signal, the

Table 5.5: Parameters that produce the best main key estimation performance.

Parameter	Value
Observations	chord transitions
Relative state self-transition likelihood	1
Chord sampling interval	100 ms
Initialisation	Krumhansl 1
Training	Prior and transition probabilities
Upper partials modelled	3
Upper partial decay rate, s	0.6
Observation probability function	discrete
Downsampling factor	16
Max. frequency (Hz)	1319
Max. frequency (MIDI)	E6
Min. frequency (Hz)	41.2
Min. frequency (MIDI)	E1
Hop size (frames)	1/8
Hop size (seconds)	0.19
Sparse kernel threshold	0
Beat tracking	none
Transient removal	none

additional energy in the high frequencies is detrimental to key estimation.

We found that limiting the lowest analysis frequency to 82.4 Hz, or MIDI pitch E2, gave a small improvement in key estimation performance. However, after looking more closely at the incorrect estimates we believe that the algorithm was finding the wrong key when there are ambiguities in the music, and so we continue to include the lowest frequencies for analysis since the bass is considered so important for determining harmony in music theory. Including the lowest frequencies does mean that the bass notes are smeared over time. It may be possible to achieve better performance by using a separate bass chromagram with a lower Q, similar to that implemented by Mauch and Dixon [2008].

A hop size of between 0.046 s and 0.19 s gave the best key estimation results, with variations in performance most likely due to differences in the rate of harmonic change in the music, which is not constant with respect to time in seconds. We select a hop size of 0.19 s for our final model because it produces the lowest number of frames out of the values within the suitable range, and so requires the shortest computation time.

The key recognition algorithm is very robust to thresholding of the transform kernels, but nonetheless the best performance was achieved with the most accurate kernels, so we use these for our final version of the algorithm.

We measured the total computation time required for key estimation for all of our test collections together, using selected parameter sets. The greatest computational saving was achieved by applying a threshold to the constant-Q spectral kernels, which means that small kernel values

are set to zero and multiplication by these values need no longer be performed. Using a threshold of 0.0055 resulted in almost a 50 % reduction in running time. Applying downsampling was also shown to offer good computational savings, of 16 % for a downsampling factor of 16. The implementation of the downsampling operation is not the most efficient, so further reductions in computation time could be achieved. Smaller savings were also shown to be possible by using shorter frames, with the side effect of excluding some low frequencies, and by using a longer hop size, with the side effect of losing time resolution.

We investigated the effects on key estimation performance of using frame lengths of one musical beat, and removing the transient parts of the signal before processing, but found that both of these preprocessing methods reduced the key estimation scores. Beat-tracking algorithms still perform relatively poorly on classical music, so do not work well on our largely classical test collections. Transients are wide-band signals that add noise to all pitch classes equally, so the peaks will still be detectable with the transients present. We verified this by showing that removing the transients is not useful for key estimation.

Having found our best-performing key estimator, we compared its performance to a tone profile correlation approach, using profiles recommended by Gómez, and profiles derived from recordings of Bach's Well-Tempered Clavier. The Bach profiles are style-specific and so performed comparatively poorly on different musical styles. The Gómez profile correlation approach gave better performance than the HMM approach for the MIREX, Rachmaninov and Bach test collections, and the HMM performed better on the Alkan, Beatles and mixed classical collections. The Beatles and mixed classical collections have the greatest variety of timbres and styles, which indicates that our HMM method of key estimation is better able to generalise to a variety of musical styles than are the simple tone profile correlation approaches.

We have presented a detailed investigation into our model for key estimation from audio, using its ability to find the single main key of a piece as an evaluation metric. In chapter 6 we turn our attention to alternative evaluation methods that are capable of measuring more detail in an automatically-generated tonal analysis of a piece.

Chapter 6

Evaluation Methods for Detailed Tonal Analyses

To date the majority of computational tonality estimation algorithms have been evaluated in terms of their ability to correctly estimate the single main key of a piece of music. However, our method and many alternative tonality estimation methods are capable of estimating the tonality of a piece in much more detail, including the relative strengths of all the different major and minor keys, and how these values change through time. Such detailed analyses are more interesting from a musicological point of view, and are likely to contain more useful information for music retrieval purposes than a single key estimate for the whole piece. However, since the analyses can be very complex, it is difficult to compare them and to determine which automatic analysis method should be used.

In this chapter we explore the possibilities for evaluating detailed tonal analyses. We first consider the requirements for an evaluation metric, and discuss how the requirements can vary depending on the desired application. We present a discussion of different possible methods of evaluation, and go on to conduct an experiment that compares a ground-truth comparison metric with retrieval-based metrics, and shows that the two classes of evaluation metric perform differently.

6.1 Requirements of an Evaluation Metric for Detailed Tonal Analyses

We wish to find a way of measuring the quality of a detailed analysis of tonal changes within a piece of music. Figure 6.1 shows an example of the kind of information we can obtain from a computer algorithm. The audio has been divided into frames, and the figure shows the relative strength of every major and minor key for each frame, so the progression of key strengths through the piece can be seen. In this case the piece is in classical sonata form, beginning in C minor, followed by a second subject in E \flat major. The first and second subjects are repeated, starting

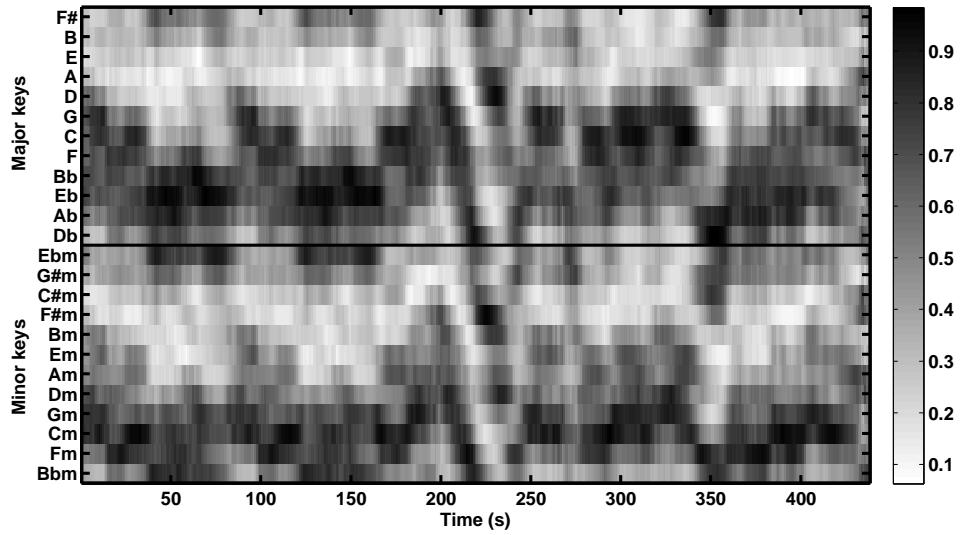


Figure 6.1: Automatically-generated detailed tonal analysis of the first movement of Beethoven’s Symphony number 5. The plot was generated from the output of a tone-profile correlation key estimation method, using Gómez’ profiles. The grey level indicates the relative key strength.

at about 100 s, then follows a development section. The development includes a rapid transition around the circle of fifths in the flat direction, seen as a steep descending pattern just after 200 s. The first subject in C minor returns at around 250 s, followed by the second subject now in C major, and the movement ends with an extended coda in C minor. These broad key changes can be traced in the plot, and we also see the influences of other keys throughout the piece. In particular there are strong links between relative major and minor keys.

When considering the requirements of a metric for measuring the quality of a detailed analysis of tonal changes within a piece of music, we must first consider for what purpose the analysis is intended. One possible application is to use the automatic analysis as an aid to a musicological study of a single piece, with no further processing, in which case the goal should be to produce an output that directly relates to human perception of tonality. This carries with it difficulties in determining how humans actually perceive tonality, discussed in section 6.2.

Another application is to use the tonal analysis for music classification and retrieval purposes, enabling queries on a large database such as “find me pieces that contain a similar tonal progression to this one”. In this case the goal is to produce an output that gives the best retrieval performance, which may or may not be the same output as best matches human perception.

Whatever the application, we require an evaluation metric to rate pieces as similar if they follow a similar tonal progression, with scope for tempo variations and key transpositions. The question of what makes two tonal progressions similar is in itself non-trivial, and our solution is described in section 6.5.2.

6.2 Comparison to Human Perception of Tonal Changes

One solution to measuring how well an automatically-generated analysis matches human perception of harmony is to compare the output against human annotations of the tonal progressions. However, machine-readable hand annotations of the required detail are very difficult to create, even with the development of a suitable syntax to represent them [Pastor Escuredo, 2008]. There exist many published harmonic analyses of various musical works by musicians such as Schenker [English edition 2004–05] and Riemann [1919], but these cannot be used directly for comparison with a computer algorithm. The first difficulty is that of time alignment. The analyses relate only to the score, not to the time of particular recordings, and generally relate to score reductions rather than the complete score that a player would use. Much of the analysis is in the form of verbal descriptions so timing information has to be further extrapolated. The second difficulty is that the analyses, both verbal and symbolic, are only comprehensible by a person with an advanced level of music theory training, substantially more advanced than that required for determination of the main key. These two issues together make the transformation of these analyses into a machine-readable format a long and difficult task that is in itself a substantial project.

Instead of transcribing existing analyses it is possible to enlist trained musicologists to create their own analyses, which are prescribed to be machine-readable and could even be aligned to a recording with the aid of a music visualisation tool such as Sonic Visualiser [Cannam et al., 2006]. However, this is perhaps a more arduous task than that of transcribing somebody else’s analysis, and is also difficult to achieve on a large scale. Krumhansl took this approach [1990, p. 96–100], asking two experts to give bar-by-bar key strength ratings from 0 to 10 for every key, for Bach’s Prelude in C minor from the Well-Tempered Clavier, Book II. We will use these ratings as ground truth for our experiments, described in section 6.5.

Both approaches to acquiring hand-annotated ground truth suffer from the problem that a tonal analysis is very subjective: it is unlikely that two people will agree on a single ground truth, so some means of generalisation is required. One approach could be to ask several people to analyse each piece in the collection, and either take an average or apply a statistical model to take account of the likely deviations. Obtaining multiple analyses in this way further increases the time taken to produce the ground truth, and raises questions of how best to combine the different analyses. Krumhansl obtained results from two people on one piece only, and the two analyses were treated separately.

6.3 Subjective testing

An alternative to using a ground truth to automatically rate various algorithms would be to ask music experts to do the rating, given the original recording, a musician’s analysis to function

as ground truth, and a plot to be assessed such as that shown in figure 6.1. Subjective tests have proved to be a highly valuable method of assessment where a ground truth is available and the human test subjects only function as comparators [ITU Recommendation, 1994–97], so the problem of automatically determining similarity is eliminated. Human subjects would not require the ground truth to be in a machine-readable format, but would need a high level of musical training to be able to follow published analyses such as those of Schenker in order to perform the similarity judgement. The problem of obtaining ground truth would also still be present.

It is possible to formulate a subjective test that does not require a ground truth, where the question posed would be not “how close is this tonal analysis to that of Schenker?”, but “is analysis A better or worse than analysis B?”. The advantage of subjective tests is that comparison words such as *better* and *worse* do not have to be more specifically defined: it is up to the test subject to decide which factors constitute a better analysis. In this way the responses are not constrained to the dimensions in which we expect to measure “goodness”.

In practice we expect to find low correlation between subjects, given the complex nature of a tonal analysis. Comparing two analyses thoroughly would require as much time as analysing the piece directly, as if to create a ground truth. To compare the analyses less thoroughly would mean that detail is overlooked.

Both automatic comparison against human ground truth and subjective comparison of tonal analyses are problematic and time-consuming. Our aim is to produce an automatic tonal analysis that is suitable for retrieval purposes, so we turn our attention to retrieval-based evaluation measures.

6.4 Application-based testing

We propose framing the evaluation problem in a retrieval context, posing the question, “can the automatically-generated tonal analyses be used to retrieve tonally similar pieces from a large, mixed database?”. By measuring how well the algorithm performs in its intended context we implicitly test the quality of the tonal analysis.

To measure tonal similarity between pieces once again requires human intervention to decide what constitutes a similar tonal structure, but there are some cases where the tonal structure is likely to be very similar in different recordings: theme and variations, different recordings of the same piece, and cover songs. With a collection of such tonally similar pieces and a mixed collection of pieces we can run a retrieval experiment that is equivalent to a cover song identification task, using standard retrieval measures and with no need for human quality judgements.

A retrieval-based metric measures the ability of an algorithm to produce consistent results for

music with the same harmonic pattern but differing in other dimensions such as instrumentation or tempo. We wish to know whether an automatic tonality estimation algorithm that best matches human perception of tonality is also the best algorithm for the purpose of retrieving tonally similar pieces from a mixed database, so we design an experiment to investigate.

6.5 Experiment Design

The experiment tests whether a ground-truth comparison evaluation technique and a retrieval-based evaluation will rank different tonal analysis algorithms in the same order. Our experiment is not exhaustive due to limitations in available ground truth, but is designed as a pilot study to prove the concept. We take nine different automatic tonal analysis algorithms, and rank them in order of quality as determined by a ground truth comparison measure, and by three different retrieval-based measures.

6.5.1 Tonal Analysis Methods Used in the Evaluation Experiments

We used two different types of automatic tonal analysis for the experiments, with various parameter settings to give a total of nine different analyses of each recording. The automatic methods are summarised in table 6.1.

Table 6.1: Tonal analysis methods used in the evaluation experiments.

Method number	Type	Frame length	
		chroma windows	seconds
1	profile	1	1.48
2	profile	2	2.96
3	profile	4	5.92
4	profile	8	11.84
5	profile	16	23.68
6	profile	32	47.36
7	profile	whole track	whole track

Method number	Type	Initialisation method
8	HMM	flat
9	HMM	probe tone

Tone Profile Correlation

The first seven techniques take the form of a sliding window-based tone profile correlation, as used in section 5.4. The audio is divided into frames, and a chroma vector is calculated for each frame to give a measure of the energy in each semitone. The chroma parameters used are shown

Table 6.2: Parameters used for chromagram calculation in the evaluation experiments.

Parameter	Value
Downsampling factor	4
Max. frequency (Hz)	1760
Max. frequency (MIDI note)	A6
Min. frequency (Hz)	55
Min. frequency (MIDI note)	A1
Frame size (seconds)	1.48
Frame size (samples)	16384
Hop size (frames)	1
Hop size (seconds)	1.48
Hop size (samples)	16384
Sparse kernel threshold	0.0055

in table 6.2¹. To estimate the key of one frame of a recording, the correlation (see equation 2.11 on page 47) between the chroma vector for the window and each of 24 key templates is calculated to give a key strength vector. This is done for each frame to give a time-varying key strength analysis.

The profile values used as key templates were derived from recordings of Bach Preludes and Fugues, as shown in figure 2.3(c) on page 25. They were generated by calculating average chroma vectors for each Prelude or Fugue, which were then normalised to the title key. The normalised major and minor average chroma vectors were then summed to give a single major and single minor template, which can be rotated to give a template for any major or minor key.

In order to create different levels of detail to test, we replaced groups of consecutive frames with one averaged frame, which increases the effective frame length. For methods 1 to 6 the number of consecutive chroma frames that were averaged ranged from 1 (equivalent to no averaging), to 32, giving effective frame lengths of between 1.48 s and 47.36 s, with no overlap. For method 7 we averaged the chroma features over the whole track, regardless of the track length. The resulting frame lengths are shown in table 6.1, and some example tonal analyses at different resolutions are shown in figure 6.2.

Hidden Markov Model

The final two tonal analysis techniques use the discrete HMM described in chapters 3 and 4. It models chord transitions as emissions from states that represent keys, with chord transitions estimated by a separate chord recognition step. The chroma parameters shown in table 6.2 were used, and the chord templates were binary, as in table 4.2 on page 89, without any modelling of the upper partials in the signal. The posterior state probabilities of the model are calculated for

¹These experiments were carried out before the parameter optimisation experiments of chapters 4 and 5, hence the difference in parameter settings.

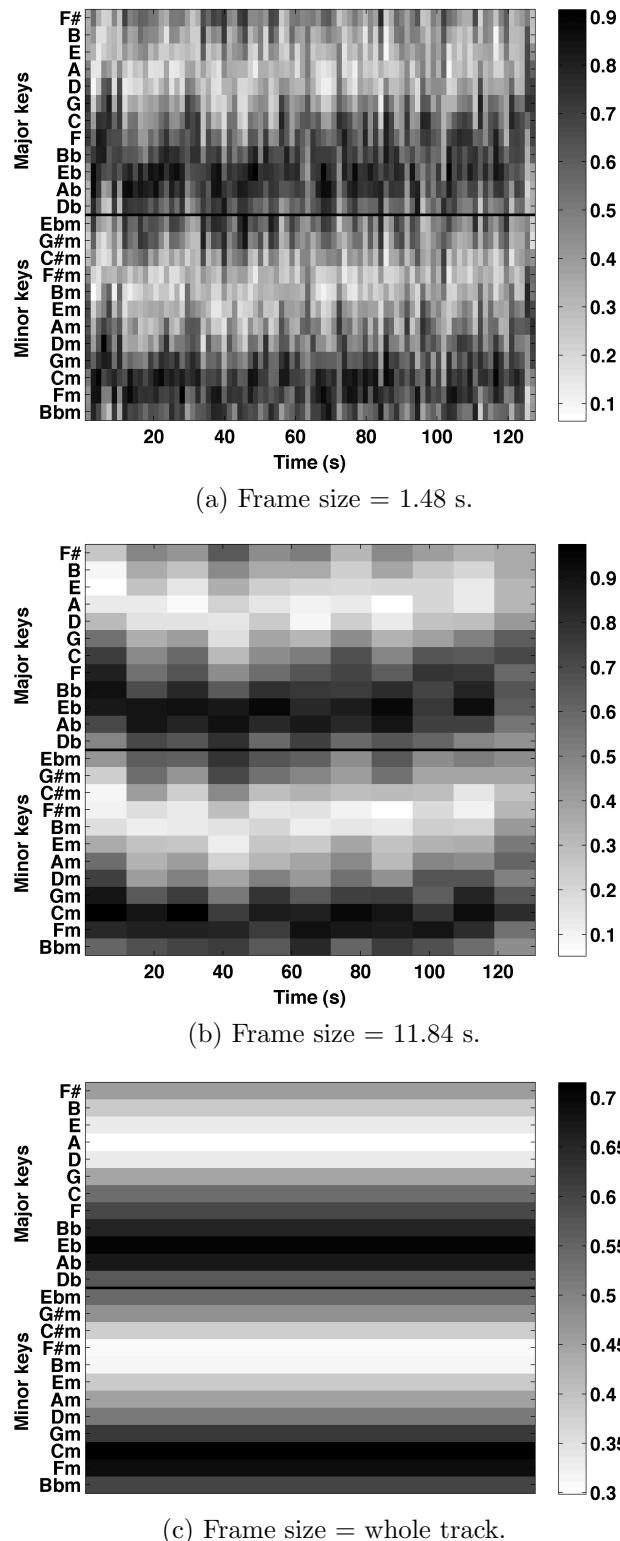
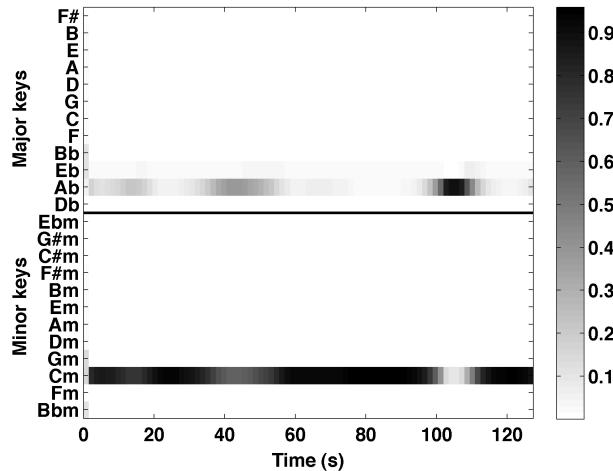
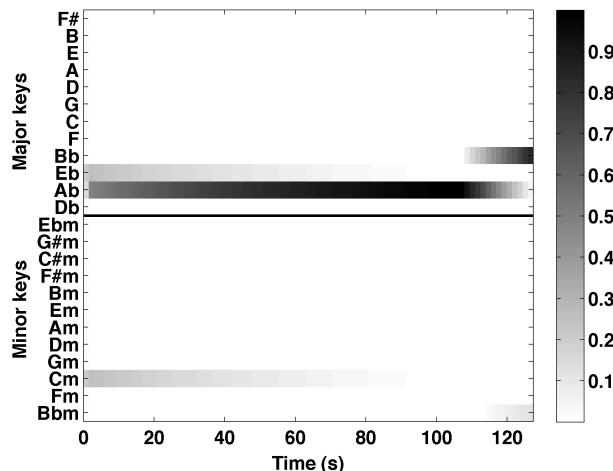


Figure 6.2: Example automatic tonal analyses of the Bach C minor Prelude using the profile correlation method with different frame sizes. The grey level indicates the relative key strength.

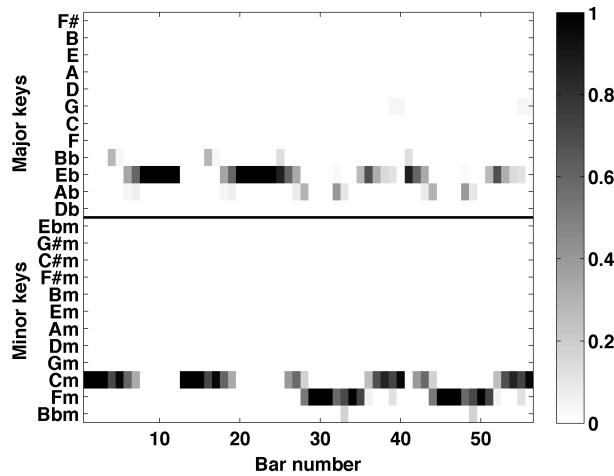


(a) Tone profile initialisation.

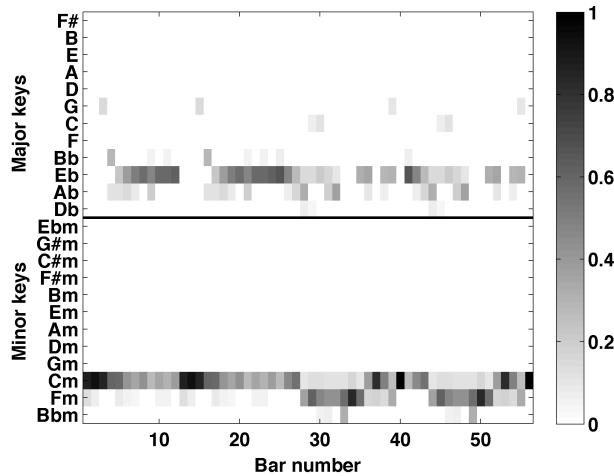


(b) Flat initialisation.

Figure 6.3: Example automatic tonal analyses of the Bach C minor Prelude using the HMM method, with tone profile and flat initialisations of the transition and observation probabilities. The grey level indicates the relative key strength. Note that the HMM with flat initialisation has incorrectly estimated the main key as Ab major.



(a) Ratings by Expert 1.



(b) Ratings by Expert 2.

Figure 6.4: Expert ratings for the Bach C minor Prelude. The grey level indicates the relative key strength.

each frame, and used as key strength values. We tested one version with flat initialisation profiles for the transition and emission probabilities, and a version with Krumhansl probe tone profile initialisation as for our final choice of discrete model. These initialiation values are described in section 3.3.2 starting on page 74. Figure 6.3 shows some example tonal analyses that were produced by the two HMM methods.

6.5.2 Ground Truth Comparison

Our first evaluation technique is to compare the nine different tonal analyses to an expert-annotated ground truth. The data we use as ground truth refers to Bach's Prelude in C minor from the Well-Tempered Clavier, Book II, annotated by two musical experts and reported by Krumhansl [1990, p. 98]. Each set of annotations gives a bar-by-bar key strength rating from 0 to 10 for every key. We normalise the ratings so that the key strengths sum to unity for each bar, then manually align the bar numbers to absolute time in a recording by Ashkenazy. The ratings of the two experts are shown in figure 6.4.

We create an automatic tonal analysis of the same recording by Ashkenazy using each of the nine different methods from table 6.1. We then find the similarity between each of the expert annotations and the nine different automatic annotations using the similarity measure described in the next section.

Similarity Measure

There are many possibilities for a similarity measure [Pampalk, 2006, chapter 2]. We select a method that fulfills our requirements of adapting to both tempo changes and transpositions, proposed by Gómez and used successfully for cover song retrieval based on tonal features [Gómez and Herrera, 2006].

We first address the need for transposition invariance. We find the main key of each track by summing the key strength vectors across the length of the track, and finding the key with the highest cumulative key strength. The key strength matrices are then rotated so that the main key always appears to be either C major or A minor. This technique to allow main key transpositions has been shown to improve retrieval performance [Gómez and Herrera, 2006].

The basis of the similarity calculation is to apply dynamic time warping to a dissimilarity matrix. Similarity matrices were first proposed by Foote [1999] to visualise musical structure, and consist of a measure of the similarity of every possible pair of frames within one piece of music, resulting in a square matrix in which repeated passages appear as diagonal lines. To produce a dissimilarity matrix the similarity measure is replaced by a dissimilarity measure. We calculate the dissimilarity between every pair of key strength vectors, not within a single piece but between the ground truth and the automatic annotation. Figure 6.5(a) shows an example

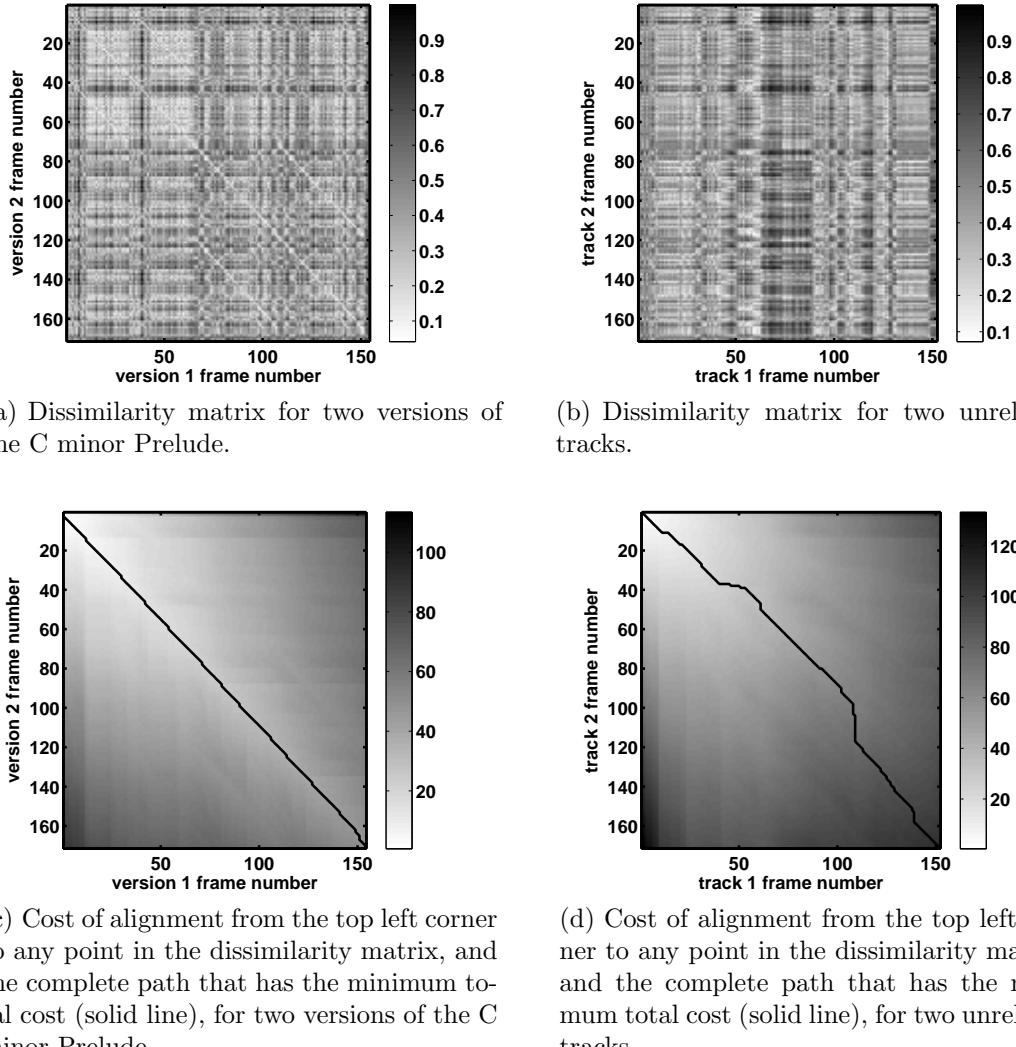


Figure 6.5: Example similarity matrices for similar and dissimilar pieces, with their alignment costs.

dissimilarity matrix for two different recordings of the same piece, in which some parallel diagonal lines are visible, indicating repeated sections. The dissimilarity matrix for two unrelated pieces in figure 6.5(b) shows no diagonal patterns.

To compute the dissimilarity between two key strength vectors, each 24-dimensional key strength vector is separated into two 12-dimensional vectors that contain the major and minor key strengths respectively. The major and minor vectors are transformed into the 6-dimensional tonal space devised by Harte et al. [2006]. The dissimilarity $\varepsilon_{m,n}$ between them is given by the sum of the Euclidian distances between the two major and two minor tonal centroids in the space, ζ_{maj}^a , ζ_{min}^a , ζ_{maj}^b and ζ_{min}^b .

$$\varepsilon_{m,n} = \|\zeta_{\text{maj}}^a - \zeta_{\text{maj}}^b\|_2 + \|\zeta_{\text{min}}^a - \zeta_{\text{min}}^b\|_2 \quad (6.1)$$

This dissimilarity measure accounts for the close relationship between keys that are adjacent on

the circle of fifths, although an alternative measure would be required to represent major-minor relationships.

We then apply dynamic time warping as implemented by Ellis [2005] to the dissimilarity matrix to give a measure of how difficult it would be to map one key strength matrix onto the other, taking into account variations in tempo within a piece. The dynamic time warping algorithm finds the path through the dissimilarity matrix, from the top left to the bottom right, that minimises the total cost of alignment. The total cost of alignment is the sum of dissimilarities between every pair of frames in this optimal path. Figures 6.5(c) and 6.5(d) show examples of the intermediate alignment costs together with the optimal complete path for two different pairs of tonal analyses. In figure 6.5(c), where the two tracks are different versions of the same piece, the optimal path is almost straight, whereas in figure 6.5(d), where the two tracks are unrelated, the optimal path is not straight and is therefore longer. The total cost of alignment is the value of the element in the bottom right corner of the figures. For longer pieces, the shortest possible path and therefore the lowest alignment cost is increased, so we normalise the cost function to the length of the longer track to make it less susceptible to large tempo changes between tonally similar pieces.

This similarity metric is one of many possible metrics that could be explored, but we believe that it is a good starting point since it addresses the most important aspects of a tonal analysis: a relative key progression through time that is invariant to tempo changes.

The cost of alignment between each of the two ground truth key strength matrices and the nine automatically-generated key strength matrices was calculated, and used to rank the automatic tonal analysis methods according to how closely they match the expert ratings.

6.5.3 Retrieval Experiments

We conduct the retrieval experiments on the same piece of music, the Bach C minor prelude from the Well-Tempered Clavier, Book II. For this we require a set of recordings that follow the same tonal progression, and a set of mixed pieces from which to retrieve them.

Recordings Used for Retrieval Experiments

We use different recordings of the same prelude to create our set of similar pieces, summarised in table 6.3. We used four different commercial recordings, which vary in tempo, instrument, tuning frequency and recording quality.

To supplement this we also make use of artificially-generated recordings that have been synthesised from MIDI using *Timidity++* [Izumo, 2008], which allowed us to further vary the instrument, key and tempo. The original MIDI file is part of the Mutopia Project [Mehrback, 2003]. Artificial data has been used very successfully for training a harmonic model [Lee and

(a) Original prelude.

(b) Triplet rhythm.

(c) Chords only.

Figure 6.6: First two bars of the original Bach prelude and the two variations.

Slaney, 2008], since it produces a set of files in which the harmonic progression is always the same but the audio features vary considerably. Although the use of artificially-generated test data may not be a perfect test for real audio applications, it is a useful way to produce a controlled experiment for initial comparisons of different algorithms.

The synthesised pieces allow us to vary the key, tempo and instrumentation, but they are all products of the same score. In order to further vary the note patterns we created two variations and recorded them ourselves on an electric piano: one that kept the same chord sequence but reduced each set of four semiquavers to a triplet, and another where only the chords were played. Figure 6.6 shows the score for the first two bars of the original prelude and the two variations.

The tonally similar recordings were added to a mixed database of 100 tonal classical pieces from the Baroque period through to the early 20th century, including other pieces by J. S. Bach. The pieces are played on various instruments, including piano solo, orchestra, and small chamber ensembles.

To thoroughly test a retrieval algorithm we would use more diverse variations and a much larger database from which to retrieve them. However, the purpose of this experiment is to test whether an evaluation metric based on retrieval performance gives equivalent results to an

evaluation metric based on hand-annotated ground truth. In order to make the comparison, we require the use of a query for which a human analysis is available. We also do not require a very large retrieval database since it is the ranking of the different tonal analyses that is important, not their absolute retrieval capabilities.

Table 6.3: Tonally similar recordings used in the retrieval experiment.

Recording number	Category	Information
1	CD	Ashkenazy, piano
2	CD	Asperen, harpsichord
3	CD	Barenboim, piano
4	CD	Fischer, piano, noisy recording
5	MIDI	harpsichord, as original
6	MIDI	violin, fast, A minor
7	MIDI	flute, slow, F minor
8	self	triplets, slow
9	self	chords only, slow

Retrieval Measures

Each variation was used in turn as a query on the database of mixed recordings plus the tonally similar recordings. The database items were ranked by the dissimilarity measure described in section 6.5.2. To quantify the retrieval performance we use the measure bpref^* proposed by Serrà [2007] for cover song retrieval, which takes into account the rank position of each relevant result up to a specified highest rank, and the number of possible relevant results in the whole database. The bpref^* value is given by

$$\text{bpref}^* = \frac{1}{|R_q|} \sum_{j=1}^{|R_a|} \left(1 - \frac{N_{nr}(j)}{|A| + |R_q|} \right) \quad (6.2)$$

where $|R_q|$ is the number of variations in the collection, $|R_a|$ is the number of variations that appear in the answer set, $|A|$ is the size of the answer set, and $N_{nr}(j)$ is the number of irrelevant answers ranked before the j -th variation in the answer set. We calculated values for a highest rank of 10 and 20, and for each the average bpref^* value was found over all 9 possible queries. The retrieval performance of the different tonal analysis methods was compared using this average bpref^* value. We also calculated the average rank of each of the tonally similar recordings as an alternative measure.

For each of the three retrieval metrics we use the tonal analysis of each of the nine similar pieces in turn as a query on the mixed database, which contained analyses of the remaining eight similar pieces. We calculated the average of the nine scores for each metric to give final retrieval performance scores for the current tonal analysis method.

6.6 Results of the Comparison of Evaluation Measures Experiment

Table 6.4 shows the ranking of the different tonal analysis algorithms, from best to worst, according to comparison against the two expert ratings and according to the three retrieval measures.

We note that the rankings from comparison against the two expert ratings, although similar, are not identical, which highlights the problem that tonality estimation is very subjective. Referring to figure 6.4 we see that the ratings of Expert 1 include fewer relevant keys at once than those of Expert 2, leading to a structure that is closer to an alternation between C minor and E flat major than the more scattered plot of Expert 2's ratings. Expert 1 has indicated several full key changes, to E flat major and F minor, shown by sections where these keys are the only ones exerting an influence. In contrast, Expert 2 has indicated that the main key, C minor, is always exerting an influence, even if it is not the strongest key at all times. The relative major key, E flat major, is also more persistent in the ratings of Expert 2. The constant presence of C minor in Expert 2's ratings should be best captured by an automatic analysis that is strongly smoothed, but Expert 2 also indicates some finer detail in his ratings of the more distant keys than did Expert 1, which could not be represented by a strongly smoothed automatic analysis. Hence neither expert's ratings is shown to be more or less suited than the other's to strong smoothing.

Table 6.4: Ranks given to the different tonal analysis techniques by different evaluation measures. Numbers in the right column are the automatic tonal analysis method numbers from table 6.1.

Measure type	Algorithm ranking (best to worst)
Expert 1	9 8 4 3 2 5 6 1 7
Expert 2	8 9 4 5 3 6 2 7 1
bpref*10	4 5 3 6 2 1 9 7 8
bpref*20	4 5 3 6 9 2 7 1 8
rank	4 5 3 9 2 6 1 7 8

Of the automatically-generated analyses we look first at the tone profile correlation methods, numbers 1–7. Some interesting relationships emerge when the evaluation metrics are plotted. Figure 6.7 shows the average of the two measures from ground truth comparison together with the three retrieval measures plotted against frame size. The ratings of the two experts show the same shape, so the average of the two is used for comparison against the retrieval measures. Although the different measures do not agree on the precise ordering of the tonal analysis methods, clearly all measures prefer the 11.84s frame size, with performance dropping as the resolution is either increased or decreased. This means that the frame size giving an automatic analysis that best matches the human judgements also gave the best retrieval performance. For the profile correlation analysis methods we would have been able to use a retrieval measure to optimise the

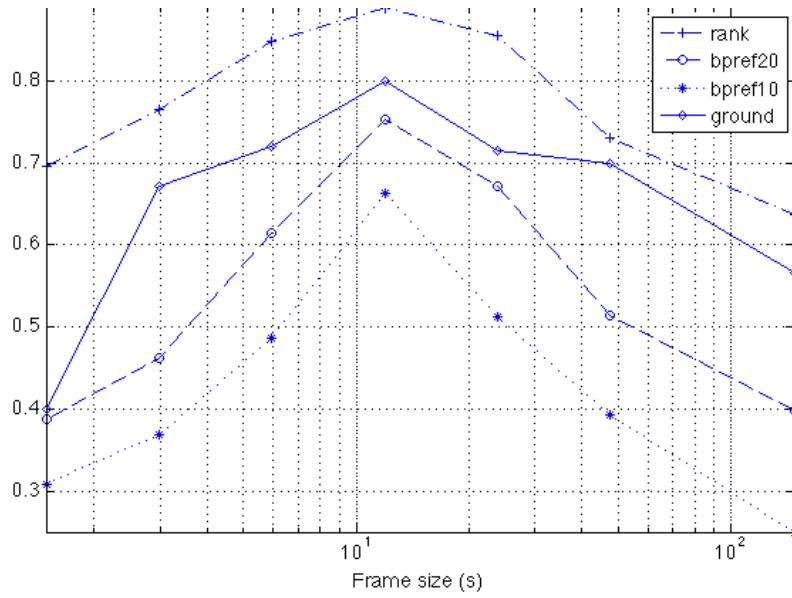


Figure 6.7: Evaluation scores for the profile-based tonal analysis methods, numbers 1–7, each method with a different frame size. The rightmost points on each line correspond to method 7 (frame size = whole track length). Each line represents a different evaluation metric: the (inverted) rank average, two bpref* metrics, and the similarity to the ground truth, as indicated. The average of the two experts’ ratings was used for the “ground” line. The vertical scale is different for each metric, so it is the shape of the lines that is of interest.

frame size, and arrive at the same answer as from optimisation using ground truth comparison.

The shape of the lines in figure 6.7 can be explained using a hierarchical view of harmony: a multi-level structure with chord changes at the lowest level and finest resolution, moving through short tonicisations and longer modulations to a single key for the whole piece at the highest level. In the Ashkenazy recording 11.84s corresponds to roughly 4 bars; a resolution that is able to smooth out the chord changes (there are typically 4 chords per bar) but still capture the larger key movements recorded by the experts. Shorter frames gave key regions that were much more disjunct than the expert analyses so were ranked lower in the ground truth comparison, and longer frames did not capture enough key variation to match the experts so were also ranked lower. Note that none of the profile methods was able to give both suitably smoothed blocks of the most important keys and discern the small key variations such as the hints of G major in Expert 2’s ratings (see figure 6.4(b)).

Similar reasoning can be used to explain the retrieval results. A single key vector for the whole track (method 7 in table 6.1) contains insufficient information to represent the piece, especially since we rotate the key strength plots so that they are all in C major or A minor. At the other end of the scale, with very fine time resolution, performance is also poor because it is at this low level that the differences between the variations occur. In order to disregard these differences and capture the tonal aspects that are constant across variations it is necessary to apply some smoothing, which can be achieved by increasing the frame size.

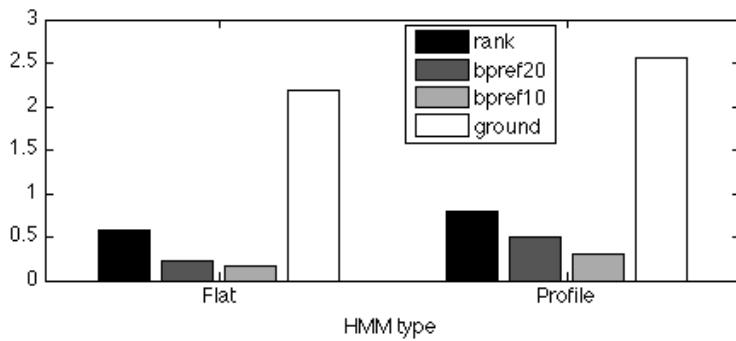


Figure 6.8: Evaluation scores for the HMM-based tonal analysis methods, numbers 8–9, which use flat initialisation and profile initialisation respectively. Each bar colour represents a different evaluation metric: the (inverted) rank average, two bpref^* metrics, and the similarity to the ground truth, as indicated. The vertical scale is different for each metric, so it is the relative height of the bars for each metric that is of interest. The average of the two experts’ ratings was used for the “ground” values.

Turning to the HMM-based methods, numbers 8 and 9 in table 6.1, we see that once again comparison against the ratings of the two experts did not rank the methods in the same order, showing that comparison against ground truth that has been created by one person is not a perfect measure. Expert 1’s ratings matched the probe tone initialised HMM better than the HMM with flat initialisation; Expert 2’s ratings matched the HMM with flat initialisation more closely. The analysis from the HMM with probe tone initialisation correctly shows the main key as C minor, and, given the rotation of the key strength matrices for transposition invariance, the analysis from the HMM with flat initialisation would show the main key as E flat major. We have already noted that E flat major is more persistent in Expert 2’s ratings than in Expert 1’s, which in this case has caused the automatic analysis that has incorrectly estimated the main key to be the best match. Figure 6.8 shows the evaluation scores for methods 8 and 9 on the same scale as figure 6.7.

The HMM key strengths were much closer to the experts’ analyses than the profile key strengths, the reason for which is clear from studying figures 6.2, 6.3 and 6.4, which show typical key strength plots from the profile method and the HMM method, and the expert ratings. The HMM applies smoothing to the chroma values in the time domain, and gives strong emphasis to the few most important keys at any time, so the key strength plots resemble the expert ratings much more closely than the more noise-like profile plots.

However, all three retrieval measures rated method 8 lowest, and method 9 as no better than 4th position. This tells us that the profile correlation analyses are much better for retrieval purposes than the smooth HMM analyses, even though the HMM versions match the ground truth more closely. The tonal analysis algorithm that most closely matched the ground truth was not the best algorithm for retrieval.

6.7 Summary

Many automatic tonal analysis algorithms are capable of estimating more than just the main key of a piece. However, suitable evaluation methods to compare different algorithms have not yet been developed. If the desired result is an automatic analysis that most closely matches human perception of tonality, the most obvious evaluation method is to compare the analyses against a hand-annotated ground truth. Suitable sources of ground truth are difficult to find, and it would also be necessary to find a measure that takes into account the discrepancies between the way different subjects perceive tonal progressions.

A likely application for an automatic tonal analysis algorithm is to retrieve pieces with a similar tonal progression from a large database. We have conducted a study to compare a ground truth comparison to a retrieval-based evaluation, and shown that although it was possible to optimise a particular type of algorithm using retrieval to get the same optimum as from a ground truth comparison, when diverse algorithms were compared ground truth and retrieval-based evaluation measures were not equivalent.

This means that striving for a tonal analysis method that best matches human perception will not necessarily produce the best retrieval results. If retrieval is the intended application of the tonal analysis, the most appropriate evaluation measure will be one that directly measures the retrieval performance.

Chapter 7

Conclusions and Future Directions

In this chapter we draw together our conclusions from the experiments described throughout the thesis. We go on to suggest directions for future research, including further development of the tonality models and applications of existing technology.

7.1 Conclusions

In chapter 2 we gave an overview of previous approaches to automatic key estimation. The most popular method is to compare chroma features to a set tone profile representing the pitch class distribution in a given key. This kind of method is capable of capturing much of the essence of a tonality, but we believe that its inherent restrictions of ignoring information from pitch progressions through time, and the requirement that a fixed pitch class distribution be set, prevent further improvements in performance.

Several researchers have turned to statistical modelling approaches, in particular the hidden Markov model, in order to introduce some information about time progression and to allow some flexibility in the prescribed harmonic relationships. The model we have developed is a hidden Markov model in which the hidden states represent keys, and the observation probabilities represent the relationships of chords to each key.

We tested many variants of the model in chapter 3, using hand-annotated chord symbols as observation data. We found that good initialisation of the transition and observation probabilities is extremely important, and much more important than extensive training for our case where we are limited to an unsupervised approach. We found that using the listener judgements of the relationships between chords and keys as the initial probabilities resulted in much better key estimation performance than either simple binary initialisation (the chord is in the key or not), or random initialisation with adaptation. Other parameters did not have such a great effect on performance, but we showed that better results could be achieved by using chord transitions as the observations instead of single chords, therefore including information about a longer chord progression. We also found that the inclusion of augmented and diminished chords as possibilities

for the observations offered only a small improvement in performance, so adapting the model to only allow major or minor chords would be a suitable method of reducing the complexity if required.

Having investigated tonality estimation at the perceptual level of chords, in chapter 4 we added an automatic chord recognition step to enable the model to function on audio data. We used a template matching chord recognition approach, which introduces many errors, but the HMM at the higher level is able to smooth out occasional incorrect chord estimates. We investigated an approach to modelling upper partials in the signal, and showed that even a simple exponential decay model improves the average key estimation performance. We also developed an HMM with a continuous observation probability density function in the chroma feature space, that does not require a definite chord classification, but found that our discrete model gave better performance. However, the continuous model was not extensively tested, so may still be able to offer some improvements.

In chapter 5 we turned our attention to the very first stages of feature extraction with investigations into the effects of low level parameter choices. We found that downsampling by a factor of 16, and so limiting the highest frequency analysed to MIDI pitch E6, gave a small improvement in key estimation performance, showing that the upper partials in the signal are in fact detrimental to key estimation even with a linear decay model of their amplitudes. This result may not hold for alternative approaches to chromagram calculation that make use of the upper partials to emphasise fundamental frequencies, such as the methods by Chuan and Chew [2005b] or Peeters [2006a]. We also saw a small improvement in key estimation performance when the lowest octave was removed from the analysis, but after inspection of the pieces in question we concluded that incorrect decoding of the harmonies was the cause of the errors, not the inclusion of the lowest notes. Such errors can only be correctly addressed by improving the higher level model.

We found that a range of hop sizes were suitable, but hypothesised that this range was due to differences in rates of harmonic change in the music. We later set the hop size to one beat, after applying an automatic beat detection algorithm to the audio, in order to normalise the frame rate to the musical tempo. However we found that the beat detection algorithm was not accurate enough for our purposes, and applying this step led to a large drop in key estimation scores. We also investigated the effects of applying an automatic transient removal algorithm before further processing, but this was also found to reduce performance scores.

We measured computation times for selected parameter sets, and found that applying a threshold to the constant-Q transform can give almost 50 % reduction in computation time with minimal loss of accuracy. Downsampling was also shown to be a good means of reducing the running time, especially since we found that it improves performance.

We compared our HMM method of key estimation to a tone profile correlation method, and found that although the difference in performance was not consistent for different datasets, the HMM performed better on the datasets with the greatest variation in timbre and style, which indicates that its ability to smooth over errors in chord recognition means it is better able to generalise for different types of music. This is an encouraging result, and suggests that further research into statistical models would prove fruitful.

In chapter 6 we discussed how to perform evaluation of a tonal analysis that expresses the relative strengths of all keys as they change over time. We compared a ground truth comparison measure to a retrieval measure, and found that the two measures gave different rankings to the different tonal analyses tested. The experiment was a preliminary study, but results indicated that the choice of evaluation measure is very important, and should match the intended application.

7.2 Future Directions

Our research has highlighted many directions in which further investigation would be worthwhile, which we will divide into four categories: developments of tonal models, improvements to the feature extraction process, development of suitable evaluation metrics, and application of the automatic tonality estimation algorithm to practical problems.

There are many possible ways in which the tonal analysis model could be improved. We believe that use of statistical models is an appropriate way of modelling harmonic relationships and the differences in perception of harmonic relationships between listeners: if hard chord and key classifications are converted to probabilities the model allows for the possibility of different interpretations whilst still differentiating between likely and very unlikely analyses. However, there are two important weaknesses in our approach of using hidden Markov models. The first is the Markov assumption that every state is dependent only on the previous state. Even with chord transitions as observations the maximum dependency spans only three frames, but in reality one hears longer progressions that lead to important cadence points. The second weakness is the requirement that the likelihood of remaining in a given state decreases exponentially with time, which is unlikely to represent the true nature of tonal progressions. There are many variations on the HMM structure that we would like to explore, and have been used for sequential data modelling in other fields, such as higher order Bayesian models [Du Preez, 1997] which allow for longer dependencies between states, and hidden semi-Markov models [Murphy, 2002] in which the state duration probability function can be manipulated.

There is also the possibility of using hierarchical models [Murphy and Paskin, 2001] to represent the hierarchical nature of tonality, from chords through tonicisations and modulations to a single main key. It seems a natural step to develop models that intrinsically represent these

different levels of tonal information. There have also been recent studies that include an element of rhythm modelling as part of a harmony analysis algorithm [Papadopoulos and Peeters, 2008], which is a useful first step towards an analysis that includes information about the phrase structure, perhaps to later bias the analysis towards cadence points.

The chroma feature extraction process may also be improved. In our model the chroma features take the place of pitch class distribution vectors, but we are using only a very simple technique to account for upper partials in the signal, meaning that we achieve better performance if high frequencies are excluded from the analysis. It is very difficult to perform exact note transcription given a recording of an unknown mixture of instruments (an evaluation of several state of the art polyphonic transcription algorithms was carried out as part of the 2008 Music Information Retrieval Evaluation Exchange (MIREX) [MIREX competition, 2008]), but perhaps applying an automatic transcription prior to folding of the spectrum into a chromagram will lead to features that resemble a pitch class distribution more closely. It would also be interesting to investigate alternative frequency transforms such as the wavelet transform [Qian, 2002, ch. 5] or filterbank approaches to feature extraction [Vaidyanathan, 1990, Patterson et al., 2003], that can eliminate the problems associated with the constant-Q transform of missing high frequency information due to the shorter length of high frequency windows, and of frequency windows that are not symmetrical on a logarithmic frequency scale.

We would like to expand our pilot study on evaluation methods, firstly to determine the most appropriate similarity measure for judging similarity to ground truth annotations, which will involve some subjective tests, and secondly to conduct the retrieval task using a much larger database.

The development of computational tonality estimation algorithms is interesting in its own right, but it would also be gratifying to see the algorithms used in real applications. Tonal descriptors have started to be used for music retrieval to rate tonal similarity between songs [Serrà and Gómez, 2007, Liu et al., 2008] and tonal similarity within songs to give structural descriptors [Chai, 2006, Levy et al., 2007], and systems have been proposed for storing the tonal descriptors [Gómez and Herrera, 2004a, Pastor Escuredo, 2008], but to our knowledge they have not yet been employed in a large scale search system.

It seems unlikely that computer software will be able to accurately judge all nuances of human perception of music in the near future, especially since culture and personal associations play such an important role [McEnnis and Cunningham, 2007]. However, with each development we are able to capture a little more information. Even with an incomplete understanding of perception we are able to build computer systems to perform tasks that humans cannot: to analyse and search an ever-growing agglomeration of audio data to help people find music that they like.

Appendix A

Important Musical Concepts

This appendix gives brief descriptions of the most important musical concepts that are referred to throughout this thesis.

A.1 Equal Temperament

This work is limited to the study of Western tonal music, which is based on combinations of notes that are spaced apart by one semitone. There are various systems in existence for defining the precise frequencies of each note, but this work is focused entirely on that of equal temperament. For equal temperament the frequencies of each semitone form a geometric progression with a multiplication factor of $2^{\frac{1}{12}}$, so

$$f_{i+1} = f_i \times 2^{\frac{1}{12}} \quad (\text{A.1})$$

where f_i is the frequency in Hertz of the i th note. The interval of an octave is formed by sounding a note together with another of twice its frequency, and within the octave there are 12 equally-spaced (in logarithmic frequency) semitones. The equal spacing means that some tones are not precisely consonant within a given key, but an instrument tuned in equal temperament can play equally well in any key, making key changes within a piece possible.

A.2 Major and Minor Scales

In Western tonal music the 12 semitones are grouped into subsets to form scales. The two main types of scale are major and minor, and are shown in figure A.1. The minor scale can include both the natural and raised sixth and seventh degrees ($A\flat$, $A\sharp$, $B\flat$ and $B\sharp$ in figure A.1), which means that the minor subset contains more notes than the major subset. The melodic minor scale differs depending on whether it is ascending or descending, but the harmonic minor scale uses the same notes in both directions. As the names suggest, the harmonic minor is used more often to form harmonies, and the melodic minor for melodies.

The scales in the figure can all be transposed to start on any of the 12 semitones, maintaining



Figure A.1: The C major scale, and harmonic and melodic variations of the C minor scale.

the same interval relationships between each note. Each note in the scale is called a scale degree, and is given a name from table A.1.

Table A.1: Names given to the degrees of the scale.

Degree	Name
I	tonic
II	supertonic
III	mediant
IV	subdominant
V	dominant
VI	submediant
VII	leading note

A.3 Key

The notes of a given scale define a harmonic context known as a key. In each scale the starting note is the most important and the most stable, and so is used together with the mode (major or minor) to name the key, for example C major or A minor. This means that there are 24 possible keys in Western tonal music (12 key notes each with 2 possible modes). For a melody consisting of notes from a given scale, the music will be drawn towards the key note, or tonic, and is likely to end on it. Interest is added by tension and resolution created by differing amounts of consonance with the tonic. The key can change during the course of a piece of music by changing the subset of pitches used: a process called modulation.

A.4 Chords

A chord is formed when two or more notes are sounded together. The most common chords are three-note chords called triads, where each note is an interval of a third (two scale degrees¹) above the last, for example the tonic, mediant and dominant notes together form the tonic triad of a key. A triad can be built on any note of the scale, and extended to include more notes. For example a seventh chord on the dominant is built by adding the note at an interval of a seventh

¹Musical intervals are measured inclusive of the lowest and highest notes.

from the dominant (the subdominant) to the dominant triad. The most important chords in a given key are the triads built on the tonic, subdominant and dominant notes.

The sequence of chords (or implied chords) in music is important for defining the key. A cadence is a section at the end of a musical phrase and is the most important part of the chord progression for confirming a particular key.

The notes of a chord can be sounded at any octave, and in any order. The lowest note of a chord defines its inversion, which affects the stability of the chord.

List of Abbreviations

CPU	Central Processing Unit
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
EM	Expectation-Maximisation
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
HMM	Hidden Markov Model
HPCP	Harmonic Pitch Class Profile
MIDI	Musical Instrument Digital Interface
MIREX	Music Information Retrieval Evaluation Exchange
PCM	Pulse Code Modulation
PCP	Pitch Class Profile
Q	Quality Factor (of a filter)
RAM	Random Access Memory
SOM	Self-Organising Map
SVM	Support Vector Machine

References

- Bret J. Aarden. *Dynamic Melodic Expectancy*. PhD thesis, Graduate School of the Ohio State University, 2003.
- Wolfgang Auhagen and Piet G. Vos. Experimental methods in tonality induction research: A review. *Music Perception, Special Issue in Tonality Induction*, 17(4):417–436, 2000.
- Juan P. Bello. Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, 2007.
- Juan P. Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in musical signals. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, 2005.
- Judith C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1), 1991.
- Judith C. Brown and Miller S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, 92(5), 1992.
- J. Ashley Burgoyne and Lawrence K. Saul. Learning harmonic relationships in digital audio with Dirichlet-based hidden markov models. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, 2005.
- Chris Cannam, Christian Landone, Mark Sandler, and Juan Pablo Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, 2006.
- Wei Chai. Semantic segmentation and summarization of music. *IEEE Signal Processing Magazine*, 23(2):124–132, 2006.
- Wei Chai and Barry Vercoe. Detection of key change in classical piano music. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, 2005.

- Elaine Chew. Modeling tonality: Applications to music cognition. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, August 2001. URL <http://www.hcrc.ed.ac.uk/cogsci2001/pdf-files/0206.pdf>.
- Elaine Chew. Slicing it all ways: Mathematical models for tonal induction, approximation, and segmentation using the spiral array. *INFORMS Journal on Computing*, 18(3):305–320, 2006.
- Elaine Chew and Alexandre R. J. François. Interactive multi-scale visualizations of tonal evolution in MuSA.RT opus 2. *ACM Computers in Entertainment*, 4(4), 2005.
- Ching-Hua Chuan and Elaine Chew. Polyphonic audio key finding using the spiral array CEG algorithm. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Amsterdam, July 2005a.
- Ching-Hua Chuan and Elaine Chew. Fuzzy analysis in pitch class determination for polyphonic audio key finding. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, 2005b.
- Ching-Hua Chuan and Elaine Chew. Audio key finding using FACEG: Fuzzy analysis with the CEG algorithm. MIREX Audio Key Finding entry (online, last accessed September 2008), 2005c. http://www.music-ir.org/evaluation/mirex-results/articles/key_audio/chuan.pdf.
- James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. URL <http://www.jstor.org/stable/2003354>.
- S. J. Cox. Hidden Markov models for automatic speech recognition: Theory and application. *Speech and Language Processing*, 1990.
- Matthew E. P. Davies and Mark D. Plumbley. Beat tracking with a two state model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–244, Philadelphia, 2007a.
- Matthew E. P. Davies and Mark D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007b.
- Karin Dressler and Sebastian Streich. Tuning frequency estimation using circular statistics. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, 2007.
- Johan A. Du Preez. *Efficient High-Order Hidden Markov Modelling*. PhD thesis, University of Stellenbosch, 1997.

- Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, second edition, 2001.
- Chris Duxbury, Mike Davies, and Mark Sandler. Separation of transient information in musical audio using multiresolution analysis techniques. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, Limerick, 2001.
- Dan Ellis. Dynamic time warp (DTW) in MATLAB. (Online, last accessed March 2008), 2005.
<http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw>.
- Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the 7th ACM International Conference on Multimedia (Part 1)*, pages 77–80, 1999.
- Takuya Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 464–467, Beijing, 1999.
- Emilia Gómez. Key estimation from polyphonic audio. MIREX Audio Key Finding entry (online, last accessed September 2008), 2005.
http://www.music-ir.org/evaluation/mirex-results/articles/key_audio/gomez.pdf.
- Emilia Gómez and Jordi Bonada. Tonality visualization of polyphonic audio. In *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, 2005.
- Emilia Gómez and Perfecto Herrera. Automatic extraction of tonal metadata from polyphonic audio recordings. In *Proceedings of the Audio Engineering Society (AES) 25th International Conference*, London, 2004a.
- Emilia Gómez and Perfecto Herrera. Estimating the tonality of polyphonic audio files: Cognitive versus machine learning modelling strategies. In *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, 2004b.
- Emilia Gómez and Perfecto Herrera. The song remains the same: identifying versions of the same piece using tonal descriptors. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006.
- Emilia Gómez Gutiérrez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.
- Christopher Harte. Chord tools and transcriptions. Centre for Digital Music Software (online, last accessed June 2006), 2005.
<http://www.elec.qmul.ac.uk/digitalmusic/downloads/index.html#chordtools>.

- Christopher Harte and Mark Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of AES 118th Convention*, Barcelona, 2005.
- Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, 2005.
- Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the Audio and Musical Computing for Multimedia Workshop 2006 (in conjunction with ACM Multimedia)*, Santa Barbara, California, 2006.
- David M. Howard and Jamie Angus. *Acoustics and Psychoacoustics*. Focal Press, third edition, 2006.
- David Huron and Richard Parncutt. An improved model of tonality perception incorporating pitch salience and echoic memory. *Psychomusicology*, 12:152–169, 1993.
- ITU Recommendation. ITU-R BS.1116. Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems, 1994–97.
- Özgür İzmirli. Tonal similarity from audio using a template based attractor model. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, 2005a.
- Özgür İzmirli. An algorithm for audio key finding. MIREX Audio Key Finding entry (online, last accessed September 2008), 2005b.
http://www.music-ir.org/evaluation/mirex-results/articles/key_audio/izmirli.pdf.
- Özgür İzmirli. Audio key finding using low-dimensional spaces. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, 2006.
- Özgür İzmirli. Localized key finding from audio using non-negative matrix factorization for segmentation. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, 2007.
- Masanao Izumo. Timidity++. GNU software (online, last accessed March 2008), 2008.
<http://timidity.sourceforge.net/>.
- Timothy C. Justus and Jamshed J. Bharucha. Music perception and cognition. In S. Yantis and H. Pashler, editors, *Stevens' Handbook of Experimental Psychology*, volume 1: Sensation and Perception, pages 453–492. Wiley, New York, third edition, 2002.
- C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.

- Carol L. Krumhansl. Tonality induction: A statistical approach applied cross-culturally. *Music Perception, Special Issue in Tonality Induction*, 17(4):461–479, 2000.
- Carol L. Krumhansl. The geometry of musical structure: A brief introduction and history. *ACM Computers in Entertainment*, 3(4), October 2005.
- Olivier Lartillot and Petri Toivainen. MIR in MATLAB (II): A toolbox for musical feature extraction from audio. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.
- Kyogu Lee. *A System for Acoustic Chord Transcription and Key Extraction from Audio using Hidden Markov Models Trained on Synthesized Audio*. PhD thesis, Stanford University, 2008.
- Kyogu Lee and Malcolm Slaney. Automatic chord recognition from audio using an HMM with supervised learning. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006.
- Kyogu Lee and Malcolm Slaney. A unified system for chord transcription and key extraction using hidden Markov models. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.
- Kyogu Lee and Malcolm Slaney. Acoustic Chord Transcription and Key Extraction From Audio Using Key-Dependent HMMs Trained on Synthesized Audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):291–301, 2008.
- Fred Lerdahl. *Tonal Pitch Space*. Oxford University Press, 2001.
- Mark Levy, Katy Noland, and Mark Sandler. A comparison of timbral and harmonic music segmentation algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawai'i, 2007.
- Ernest Li and Juan Pablo Bello. Key-independent classification of harmonic change in musical audio. In *Proceedings of AES 123rd Convention*, New York, 2007.
- Yuxiang Liu, Ye Wang, Arun Shenoy, Wei-Ho Tsai, and Lianhong Cai. Clustering music recordings by their keys. In *Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, 2008.
- Arpi Mardirossian and Elaine Chew. SKeFiS - a symbolic (MIDI) key finding system. MIREX Symbolic Key Finding entry (online, last accessed August 2008), 2005.
<http://www-scf.usc.edu/~mardiros/papers/MIREX2005.pdf>.

- Arpi Mardirossian and Elaine Chew. Visualizing music: Tonal progressions and distributions. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, 2007.
- G. Martens, H. De Meyer, B. De Baets, M. Leman, J.-P. Martens, L. Clarisse, and M. Lesaffre. A tonality-oriented symbolic representation of musical audio generated by classification trees. In *Proceedings of the EUROFUSE Workshop on Information Systems*, pages 49–54, 2002.
- G. Martens, H. De Meyer, B. De Baets, and M. Leman. Distance-based versus tree-based key recognition in musical audio. *Soft Computing (online)*, 2004.
- Matthias Mauch and Simon Dixon. A discrete mixture model for chord labelling. In *Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, 2008.
- Daniel McEnnis and Sally Jo Cunningham. Sociology and music recommendation systems. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, 2007.
- Jesse Mehrbach. J.S. Bach, Das Wohltemperierte Clavier II, Praeludium II. The Mutopia Project (Online, last accessed September 2008), 2003. <http://www.mutopiaproject.org>.
- MIREX competition. Audio key finding. 1st Music Information Retrieval Evaluation Exchange Contest (online, last accessed October 2008), 2005a.
<http://www.music-ir.org/evaluation/mirex-results/audio-key/index.html>.
- MIREX competition. Symbolic key finding. 1st Music Information Retrieval Evaluation Exchange Contest (online, last accessed October 2008), 2005b.
<http://www.music-ir.org/evaluation/mirex-results/sym-key/index.html>.
- MIREX competition. Multiple fundamental frequency estimation & tracking results. 4th Music Information Retrieval Evaluation Exchange Contest, 2008.
http://www.music-ir.org/mirex/2008/index.php/Multiple_Fundamental_Frequency_Estimation_%26_Tracking_Results.
- Sanjit K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, 2nd edition, 2001.
- Kevin Murphy and Mark Paskin. Linear time inference in hierarchical HMMs. In *Proceedings of the Neural Information Processing Systems Conference (NIPS)*, Vancouver, 2001.
- Kevin P. Murphy. Hidden semi-Markov models (HSMMs). (online, last accessed november 2008), University of British Columbia, 2002.
<http://www.cs.ubc.ca/~murphyk/Papers/segment.pdf>.

- Kevin P. Murphy. Hidden Markov model (HMM) toolbox for MATLAB. (Online, last accessed May 2008), 1998. <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>.
- Katy Noland and Mark Sandler. Key estimation using a hidden Markov model. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, 2006.
- Katy Noland and Mark Sandler. Signal processing parameters for tonality estimation. In *Proceedings of AES 122nd Convention*, Vienna, 2007.
- Katy Noland and Mark Sandler. Influences of signal processing, tone profiles, and chord progressions on a model for estimating the musical key from audio. *Computer Music Journal*, 33(1), 2009.
- Thomas Noll and Jörg Garbers. Harmonic path analysis. In Guerino Mazzola, Thomas Noll, and Emilio Lluis-Puebla, editors, *Perspectives of Mathematical and Computational Music Theory*, pages 395–427. epOs-Music, Osnabrück, 2004.
- H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the AIEE*, pages 617–644, February 1928.
- Elias Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. PhD thesis, Vienna University of Technology, March 2006. URL http://www.ofai.at/_elias.pampalk/publications/pampalk06thesis.pdf.
- Hélène Papadopoulos and Geoffroy Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proceedings of the 2007 International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 53–60, Bordeaux, 2007.
- Hélène Papadopoulos and Geoffroy Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–124, Las Vegas, 2008.
- Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49, 2002.
- David Pastor Escuredo. The tonality ontology. (Online, last accessed March 2008), 2008. <http://purl.org/ontology/tonality>.
- Roy D. Patterson, Masashi Unoki, and Toshio Irino. Extending the domain of center frequencies for the compressive gammachirp auditory filter. *Journal of the Acoustical Society of America*, 114:1529–1542, 2003.

- Steffen Pauws. Musical key extraction from audio. In *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, 2004.
- Steffen Pauws. Keyex: Audio key extraction. MIREX Audio Key Finding entry (online, last accessed September 2008), 2005.
http://www.music-ir.org/evaluation/mirex-results/articles/key_audio/pauws.pdf.
- Geoffroy Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, 2006a.
- Geoffroy Peeters. Musical key estimation of audio signal based on hidden Markov modeling of chroma vectors. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFX-06)*, Montreal, 2006b.
- Jeremy Pickens, Juan Pablo Bello, Giuliano Monti, Mark Sandler, Tim Crawford, Matthew Dovey, and Don Byrd. Polyphonic score retrieval using polyphonic audio queries: A harmonic modelling approach. *Journal of New Music Research*, 32(2):223–236, 2003.
- A. W. Pollack. Notes on ... series, soundscapes.info. (Online, last accessed November 2008), 2000. http://www.icce.rug.nl/~soundscapes/DATABASES/AWP/awp-notes_on.shtml.
- Boaz Porat. *A Course in Digital Signal Processing*. John Wiley & Sons, Inc., 1997.
- Hendrik Purwins. *Profiles of Pitch Classes Circularity of Relative Pitch and Key - Experiments, Models, Computational Music Analysis, and Perspectives*. PhD thesis, Technischen Universität Berlin, 2005.
- Hendrik Purwins and Benjamin Blankertz. CQ-profiles for key finding in audio. MIREX Audio Key Finding entry (online, last accessed September 2008), 2005.
http://www.music-ir.org/evaluation/mirex-results/articles/key_audio/purwins.pdf.
- Hendrik Purwins, Benjamin Blankertz, and Klaus Obermeyer. A new method for tracking modulations in tonal music in audio data format. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'00)*. IEEE Computer Society, 2000.
- Shie Qian. *Introduction to Time-Frequency and Wavelet Transforms*. Prentice Hall PTR, New Jersey, 2002.
- Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, no. 2, February 1989.

- Christopher Raphael and Josh Stoddard. Harmonic analysis with probabilistic graphical models. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, Baltimore, MD, 2003.
- Hugo Riemann. *L. van Beethovens sämtliche Klavier-Solosonaten: Ästhetische und formal-technische Analyse mit historischen Notizen (3 volumes)*. Max Hesse, Berlin, 1919.
- David Rizo and José M. Iñesta. Tree symbolic music representation for key finding. MIREX Symbolic Key-Finding entry (online, last accessed October 2008), 2005.
http://www.music-ir.org/evaluation/mirex-results/articles/key_symbolic/rizo.pdf.
- Stanley Sadie, editor. *The New GROVE Dictionary of Music and Musicians*. Macmillan Publishers Limited, London, 1980.
- Craig Stuart Sapp. Harmonic visualizations of tonal music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 423–430, Havana, Cuba, 2001.
- Craig Stuart Sapp. Visual hierarchical key analysis. *ACM Computers in Entertainment*, 4(4): 1–19, 2005.
- Heinrich Schenker. *Der Tonwille: pamphlets in witness of the immutable laws of music: offered to a new generation of youth (2 volumes)*. Oxford University Press, English edition 2004–05. Ian Bent et al., translators; William Drabkin, editor.
- Arnold Schönberg. *Structural Functions of Harmony*. Faber and Faber, London, second edition, 1969.
- Joan Serrà. A qualitative assessment of measures for the evaluation of a cover song identification system. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, 2007.
- Joan Serrà and Emilia Gómez. A cover song identification system based on sequences of tonal descriptors. MIREX Audio Cover Song Identification entry (online, last accessed March 2008), 2007.
http://www.music-ir.org/mirex/2007/index.php/Audio_Cover_Song_Identification_Results.
- Marie-Hélène Serra. Introducing the phase vocoder. In Curtis Roads, Stephen Travis Pope, Aldo Piccialli, and Giovanni De Poli, editors, *Musical Signal Processing*, pages 31–90. Swets & Zeitlinger, 1997.
- Alexander Sheh and Daniel P. W. Ellis. Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the 4th International Conference on Music Information Retrieval*, Baltimore, 2003.

- Arun Shenoy and Ye Wang. Key, chord and rhythm tracking of popular music recordings. *Computer Music Journal*, 29(3):75–86, 2005.
- Arun Shenoy, Roshni Mohapatra, and Ye Wang. Key determination of acoustic musical signals. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2004.
- Roger Shepard. Circularity in judgements of relative pitch. *Journal of the Acoustical Society of America*, 36:2346–2353, 1964.
- Roger Shepard. Pitch perception and measurement. In Perry Cook, editor, *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*, pages 149–165. MIT Press, 1999.
- Ilya Shmulevich, Olli Yli-Harja, Edward Coyle, Dirk-Jan Povel, and Kjell Lemström. Perceptual issues in music pattern recognition: Complexity of rhythm and key finding. *Computers and the Humanities*, 35(1):23–35, 2001.
- Eric Taylor. *The AB Guide to Music Theory, Part I*. The Associated Board of the Royal Schools of Music (Publishing) Ltd., 1989.
- Eric Taylor. *The AB Guide to Music Theory, Part II*. The Associated Board of the Royal Schools of Music (Publishing) Ltd., 1991.
- David Temperley. *The Cognition of Basic Musical Structures*. MIT Press, 2001.
- David Temperley. Bayesian models of musical structure and cognition. *Musicae Scientiae*, 8(2):175–205, 2004.
- David Temperley. A Bayesian key-finding model. MIREX Symbolic Key-Finding entry (online, last accessed October 2008), 2005.
http://www.music-ir.org/evaluation/mirex-results/articles/key_symbolic/temperley.pdf.
- David Temperley. *Music and Probability*. MIT Press, 2007.
- Petri Toiviainen. Visualization of tonal content with self-organizing maps and self-similarity matrices. *ACM Computers in Entertainment*, 3(4), 2005.
- Petri Toiviainen and Carol L. Krumhansl. Measuring and modeling real-time responses to music: The dynamics of tonality induction. *Perception*, 32(6):741–766, 2003.
- P. P. Vaidyanathan. Quadrature mirror filter banks, M-band extensions, and perfect-reconstruction techniques. *Proceedings of the IEEE*, 78(1):56–93, 1990.
- Piet G. Vos. Tonality induction: Theoretical problems and dilemmas. *Music Perception, Special Issue in Tonality Induction*, 17(4):403–416, 2000.

Yongwei Zhu. An audio key finding algorithm. MIREX Audio Key Finding entry (online, last accessed October 2008), 2005.

http://www.music-ir.org/evaluation/mirex-results/articles/key_audio/zhu.pdf.

Yongwei Zhu and Mohan Kankanhalli. Music scale modeling for melody matching. In *Proceedings of the 11th ACM International Conference on Multimedia*, pages 359–362. ACM, 2003.

Yongwei Zhu and Mohan Kankanhalli. Key-based melody segmentation for popular songs. In *Proceedings of the Pattern Recognition, 17th International Conference on Pattern Recognition (ICPR'04)*, volume 3, pages 862–865, Washington, DC, USA, 2004. IEEE Computer Society.

Yongwei Zhu and Mohan S. Kankanhalli. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Transactions on Multimedia*, 8(3):575–584, 2006.

Yongwei Zhu, Mohan S. Kankanhalli, and Sheng Gao. Music key detection for musical audio. In *Proceedings of the 11th International Multimedia Modelling Conference (MMM'05)*. IEEE Computer Society, 2005.