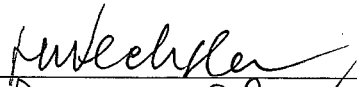
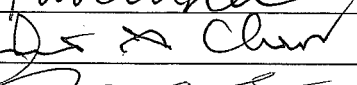
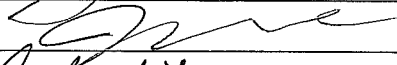
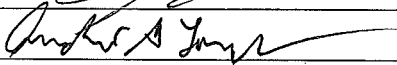
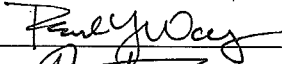
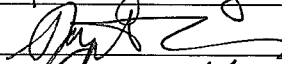
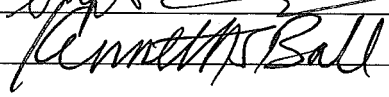


UNSUPERVISED BAYESIAN MUSICAL KEY AND CHORD RECOGNITION

by

Yun-Sheng Wang
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Information Technology

Committee:

	Dr. Harry Wechsler, Dissertation Director
	Dr. Jim Chen, Committee Member
	Dr. Jessica Lin, Committee Member
	Dr. Andrew Loerch, Committee Member
	Dr. Pearl Wang, Committee Member
	Dr. Stephen Nash, Senior Associate Dean
	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date: April 9, 2014

Spring Semester 2014
George Mason University
Fairfax, VA

Unsupervised Bayesian Musical Key and Chord Recognition

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

by

Yun-Sheng Wang
Master of Science
George Mason University, 2002

Director: Harry Wechsler, Professor
Department of Computer Science

Spring Semester 2014
George Mason University
Fairfax, VA

Dedication

To my parents.

To Lindsey, Justin, and Tammy.

Acknowledgements

It was hard for me to imagine that I would finally be on the verge of finishing my PhD degree requirements. It has been a long journey and I have quit the program numerous times. So many times – most unofficially but one officially – that I lost count. I distinctly remember the periodic crushing pressure coming from multiple fronts testing my ability to find a balance between my family of four, work, overseas family, and the program. Life gets in the way. However, Professor Wechsler's patience allowed me to progress. He pulled me back after I quit the program and showed me the path of fruitful research. For my convenience, he often opened his home to me and we discussed my progress on the weekends at his kitchen table or study. I am thankful for his guidance, support, and encouragement; without him, this dissertation cannot be born. My sincere appreciation to my committee members, Professors Jim Chen, Jessica Lin, and Pearl Wang, who carved out their precious time and energy to provide me with much needed feedback. Last but not least, given the musical nature of my dissertation which intersects the arts, science, and technology, I am privileged to have Professor Loerch, a bassoonist, on my committee; he went the extra mile with his time. His critique and advice were instrumental (no pun intended) in my preparation of the dissertation.

During this all uphill marathon, my wife, Tammy, and my two children, Justin and Lindsey, were the three people who staffed the one-and-only mobile aid station providing me with unconditional love and cheer to keep me going. Other people may see runners pass each milestone, but my family ran with me, and we crossed the finish line together. This dissertation is written for them – my best friend and wife for almost two decades, and two young budding musicians. They are my anchors and without them, I would be lost.

Table of Contents

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS OR SYMBOLS.....	ix
ABSTRACT	x
CHAPTER 1 INTRODUCTION.....	1
1.1 MOTIVATION AND APPLICATIONS	2
1.2 RESEARCH GOALS	4
1.3 THESIS ORGANIZATION	6
1.4 CONTRIBUTIONS AND PUBLICATIONS.....	9
CHAPTER 2 BACKGROUND AND RELATED WORK.....	11
2.1 MUSICAL FUNDAMENTALS	11
2.1.1 <i>Pitch and Frequency</i>	11
2.1.2 <i>Tonality and Harmony</i>	13
2.1.3 <i>Chroma and Key Profiles</i>	21
2.2 MUSIC SIGNAL PROCESSING AND PREVIOUS WORK.....	24
2.3 PREVIOUS KEYS AND CHORDS ANALYSIS	29
2.3.1 <i>Bharucha's Model</i>	30
2.3.2 <i>Summary of Previous Work</i>	33
2.3.3 <i>Recent Work After 2008</i>	41
2.4 MIXTURE MODELS.....	50
CHAPTER 3 METHODOLOGY	57
3.1 OVERVIEW OF THE METHODOLOGY	58
3.2 INFINITE GAUSSIAN MIXTURE MODEL.....	61
3.3 SYMBOLIC DOMAIN	68
3.3.1 <i>Feature Extraction</i>	68
3.3.2 <i>Keys and Chords Recognition</i>	71
3.4 AUDIO DOMAIN	73

3.4.1	<i>Wavelet Transformation</i>	74
3.4.2	<i>Chroma Extraction and Variants</i>	89
3.4.3	<i>Local Keys Recognition</i>	92
3.4.4	<i>Chord Recognition</i>	96
3.5	EVALUATION METRICS	98
CHAPTER 4	EXPERIMENTAL RESULTS	101
4.1	THE BEATLES ALBUMS	101
4.2	SYMBOLIC DOMAIN	104
4.2.1	<i>Keys Recognition</i>	104
4.2.2	<i>Chords Recognition</i>	108
4.3	AUDIO DOMAIN	113
4.3.1	<i>Key Recognition</i>	114
4.3.2	<i>Chord Recognition</i>	121
4.4	PERFORMANCE COMPARISON.....	126
4.5	TONAL HARMONY AND MACHINES.....	130
CHAPTER 5	APPLICATIONS AND EXTENSIONS	134
CHAPTER 6	CONCLUSIONS AND FUTURE WORK	141
6.1	SUMMARY.....	141
6.2	CONTRIBUTIONS	143
6.3	FUTURE WORK	144
BIBLIOGRAPHY	145
BIOGRAPHY	153

List of Tables

Table	Page
Table 1: Natural, harmonic and melodic Minor scales	16
Table 2: Formation of triads	19
Table 3: Previous work and commonly used STFT specification	26
Table 4: Previous work and commonly used CQT specification	28
Table 5: Previous work of key and chord analysis	35
Table 6: Publication count for key and chord analysis since 2008	40
Table 7: Gaussian coding examples for IGMM.....	67
Table 8: Sampling algorithm using IGMM for symbolic key and chord recognition	72
Table 9: Four stages of extracting keys and chords from audio	73
Table 10: Sampling rate for CQT	90
Table 11: Specification of frequency, bandwidth, and Q	90
Table 12: Variants of chroma features used in experiments	92
Table 13: Key sampling algorithm using IGMM (audio).....	95
Table 14: Correction rule for sporadic chord labels	98
Table 15: 12 albums of the Beatles	103
Table 16: Experimental results of key finding using K-S and IGMM	107
Table 17: Precision, recall, and F-measure for the IGMM key-finding task	107
Table 18: Sample Euclidean distance of chords	109
Table 19: Six types of chords.....	121
Table 20: Performance comparison of similar work published after 2008.....	127
Table 21: Segmentation cues	138

List of Figures

Figure	Page
Figure 1: Neuro-cognitive model of music perception (Koelsch & Siebel, 2005)	3
Figure 2: Fundamental frequencies of human voices and musical instruments and their frequency range.....	13
Figure 3: C major scale	15
Figure 4: Cardinality of chords (Hewitt, 2010)	17
Figure 5: Octave and pitch classes. Each letter on the keyboard represents the pitch class of the tone (Snoman, 2013).....	17
Figure 6: Names of musical intervals (Hewitt, 2010).....	18
Figure 7: Notation of C major, minor, diminished, augmented chords (Hewitt, 2010)....	19
Figure 8: Four types of suspended triads with c as the root (Hewitt, 2010)	20
Figure 9: (a) Pitch tone height; (b) Chroma circle; and (c) Circle of Fifth; ((a) and (b) are from Loy, D. (2006, pp. 164-165))	22
Figure 10: Krumhansl and Kessler major and minor profiles.....	23
Figure 11: Temperley key profiles.....	24
Figure 12: Framework of chromagram transformation (diagram extracted from (Müller & Ewert, 2011))	25
Figure 13: Bharucha's model (1991, p. 93)	31
Figure 14: Network of tones, chords, and keys (Bharucha, 1991, p. 97)	31
Figure 15: Gating mechanism to derive pitch invariant representation (Bharucha, 1991, p. 97)	32
Figure 16: System developed by Ryynanen and Klapuri (2008).....	43
Figure 17: (a) Dynamic Bayesian network developed Mauch & Sandler (2010); (b) DBN modified by Ni et al. (2012).....	44
Figure 18: Rule-based tonal harmony by de Hass (de Haas, 2012).....	46
Figure 19: Latent Dirichlet allocation for key and chord recognition (Hu, 2012). Left model: symbolic music; right model: real audio music	47
Figure 20: Chord recognition model developed by Lee and Slaney (2008)	49
Figure 21: A basic Dirichlet Process Mixture Model	51
Figure 22: A standard DPMM for key and chord modeling.....	54
Figure 23: Methodology overview.....	59
Figure 24: A conceptual generative process for keys and chords.....	60
Figure 25: Types of mixture models (Wood & Black, 2008). (a) Traditional mixture, (b) Bayesian mixture, and (c) Infinite Bayesian mixture. The numbers at the bottom right corner represent the number of repetitions of the sub-graph in the plate.	62
Figure 26: Specification of Infinite Gaussian Mixture Model.....	63

Figure 27: MIDI representation of "Let It Be"	70
Figure 28: ADSR envelop (Alten, 2011, p. 16)	76
Figure 29: Fundamental frequency and harmonics of piano, violin, and flute (Alten, 2011, p. 15).	77
Figure 30: Wavelet transform with scaling and shift (Yan, 2007, p. 28)	78
Figure 31: Discrete Wavelet Transform (DWT).....	79
Figure 32: Undecimated Discrete Wavelet Transform (UWT)	80
Figure 33: Four-level discrete wavelet transform (Yan, 2007, p. 36).....	80
Figure 34: Daubachies scaling functions	81
Figure 35: Symlet scaling functions	82
Figure 36: Decomposition wavelets. Top two: Low-pass and high-pass filters for db8; Bottom two: Low-pass and high-pass filters for sym8.	82
Figure 37: Frequency allocation of wavelet transform.	83
Figure 38: Amplitude and time representation of 1.5 seconds of “Let it be.” Top row represents the original signal.	84
Figure 39: Frequency and time representation of 1.5 seconds of “Let it be.” Top row represents the original signal.	85
Figure 40: Chord type distribution for the Beatles' 12 albums (Harte, 2010)	104
Figure 41: Similarity matrix for the song titled “Hold Me Tight”	111
Figure 42: Euclidean distance of IGMM chords to ground truth.....	112
Figure 43: Average chord Euclidean distances between IGMM and GT.	113
Figure 44: Overall keys distribution	114
Figure 45: Distribution of global keys.....	115
Figure 46: Distribution of local keys	115
Figure 47: Overall key finding.....	116
Figure 48: Single key finding	117
Figure 49: Multiple key finding.....	117
Figure 50: Precision improvement over CUWT-4.....	119
Figure 51: Recall improvement over CUWT-4	119
Figure 52: F-measure improvement over CUWT-4.....	120
Figure 53: Chord recognition rates	122
Figure 54: Chord recognition overlap rate (box and whisker).....	123
Figure 55: Chord recognition improvement over CUWT-4	124
Figure 56: Combined improvement over CUWT-4.....	124
Figure 57: Effect of bag of local keys on chord recognition	126
Figure 58: Music segmentation through harmonic rhythm.....	137

List of Equations

Equation	Page
Equation 1: Short-term fourier transform	26
Equation 2: Constant Q transform	27
Equation 3: Sampling rate determination	27
Equation 4: Q determination	28
Equation 5: Size of analysis frame.....	28
Equation 6: Chroma summation	29
Equation 7: Chroma vector	29
Equation 8: Normalized chroma vector	29
Equation 9: Posterior distribution of Gaussian parameter	52
Equation 10: Sampling function 1	52
Equation 11: Sampling function 2	53
Equation 12: Sampling function for an existing index variable	56
Equation 13: Sampling function for a new index variable	56
Equation 14: Sampling function for alpha	56
Equation 15: Distribution for the proportional variable	64
Equation 16: Distribution for the indexing variable	64
Equation 17: IGMM joint distribution.....	66
Equation 18: Prior for Gaussian covariance	66
Equation 19: Prior for Gaussian mean	66
Equation 20: Shannon entropy.....	87
Equation 21: Wavelet similarity measure	88
Equation 22: Adjusted chroma energy.....	97
Equation 23: Precision	99
Equation 24: Recall.....	99
Equation 25: F-measure	100
Equation 26: Chord symbol recall	100

Abstract

UNSUPERVISED BAYESIAN MUSICAL KEY AND CHORD RECOGNITION

Yun-Sheng Wang, Ph.D.

George Mason University, 2014

Dissertation Director: Dr. Harry Wechsler

Butler Lampson once said “All problems in computer science can be solved by another level of indirection.” Many tasks in Music Information Retrieval can be approached using indirection in terms of data abstraction. Raw music signals can be abstracted and represented by using a combination of melody, harmony, or rhythm for musical structural analysis, emotion or mood projection, as well as efficient search of large collections of music. In this dissertation, we focus on two tasks: analyzing tonality and harmony of music signals. Tonality (keys) can be visualized as the “horizontal” aspect of a music piece covering extended portions of it while harmony (chords) can be envisioned as the “vertical” aspect of music in the score where multiple notes are being played or heard simultaneously. Our approach concentrates on transcribing western popular music into its tonal and harmonic content directly from the audio signals. While the majority of the proposed methods adopt the supervised approach which requires scarce manually-transcribed training data, our approach is unsupervised where model parameters for

tonality and harmony are directly estimated from the target audio data. Our approach accomplishes this goal using three novel steps. First, raw audio signals in the time domain are transformed using undecimated wavelet transform as a basis to build an enhanced 12-dimensional pitch class profile (PCP) in the frequency domain as features of the target music piece. Second, a bag of local keys are extracted from the frame-by-frame PCPs using an infinite Gaussian mixture which allows the audio data to “speak-for-itself” without pre-setting the number of Gaussian components to model the local keys. Third, the bag of local keys is applied to adjust the energy levels in the PCPs for chord extraction.

The main argument for applying unsupervised machine learning paradigms for tonal and harmonic analysis on audio signals follows the principle of Einstein’s “as simple as possible, but not simpler” and David Wheeler’s corollary to Butler Lampson’s quote “..., except for the problem of too many layers of indirection.” From experimental results, we demonstrate that our approach – a much simpler one compared to most of the existing methods – performs just as well or outperforms many of the much more complex models for the two tasks without using any training data. We make four contributions to the music signal processing and music information processing communities:

1. We have shown that using undecimated wavelet transform on the raw audio signals improves the quality of the pitch class profiles.
2. We have demonstrated that an infinite Gaussian mixture can be used to efficiently generate a bag of local keys for a music piece.

3. We have ascertained that the combination of well-known tonal profiles and a bag of local keys can be used to adjust the pitch class profiles for harmony analysis.
4. We have shown that an unsupervised chord recognition system – without any training data or other musical elements – can perform as well, if not exceed, many of the supervised counterparts.

Chapter 1 Introduction

The ability to use machines to understand music has many potential applications in the area of multimedia and music information retrieval. For most of us, at a high level and without formal musical training, we can recognize whether the music being played is classical or popular as well as the mood the music piece conveys. At the middle level, listeners can easily determine whether a part being played is the chorus or refrain even with little or no formal musical training. At a low level, our brain not only can easily distinguish whether a music piece contains instruments such as piano, strings, woodwind, or percussion but is also capable of getting our foot to tap along with the rhythm of the music piece. These tasks of recognizing certain properties of a music piece are seemingly simple tasks for humans, but they remain to be difficult problems for machines to achieve a high accuracy similar to that of humans' ears and brains.

In this dissertation, we focus on developing a new methodology for machines to extract tonality (keys) and harmony (chords) from both symbolic and audio wave music. On a small scale, due to the lack of music scores of most popular music, musicians often want to extract these two elements for their own play or transcribe the piece into some other form that can be more appropriately played by different instruments or singers with different vocal ranges. On a large scale, the ability to use machines to extract keys and chords can be used to perform music segmentation, an important intermediate step to

retrieve music using machines. However, manual transcription is often a very laborious process and therefore it would be desirable for machines to perform such tasks given the large quantity of music that is available to us. Recognizing keys and chords of a music piece are two very much related tasks since knowing one would greatly help the other. In this dissertation, we present our research in key and chord recognition for popular music.

1.1 Motivation and Applications

As an amateur musician playing with a band in the past and currently with young children playing different instruments in the household, I always have the need to extract keys and chords by ears so that a music piece can be played by various instruments after transposing music. Manual analysis of tonal harmony on a few pieces is enjoyable but using machines to perform automated transcription would be much more desirable for large quantities of music media. Furthermore, the advancement of the internet and mass availability of various hand-held devices create the demand to efficiently retrieve music for listeners under different circumstances. As described by Yang and Chen (2011, p. 187), chord notations are one of the most important “mid-level” features of music and such representation can be used to identify and retrieve music with similarity. From the neuro-cognitive perspective of music perception, such “mid-level” features lay the foundations for our auditory systems and brain to interpret and analyze the structure of the music being played and move our emotions, as described in Figure 1 (Koelsch & Siebel, 2005).

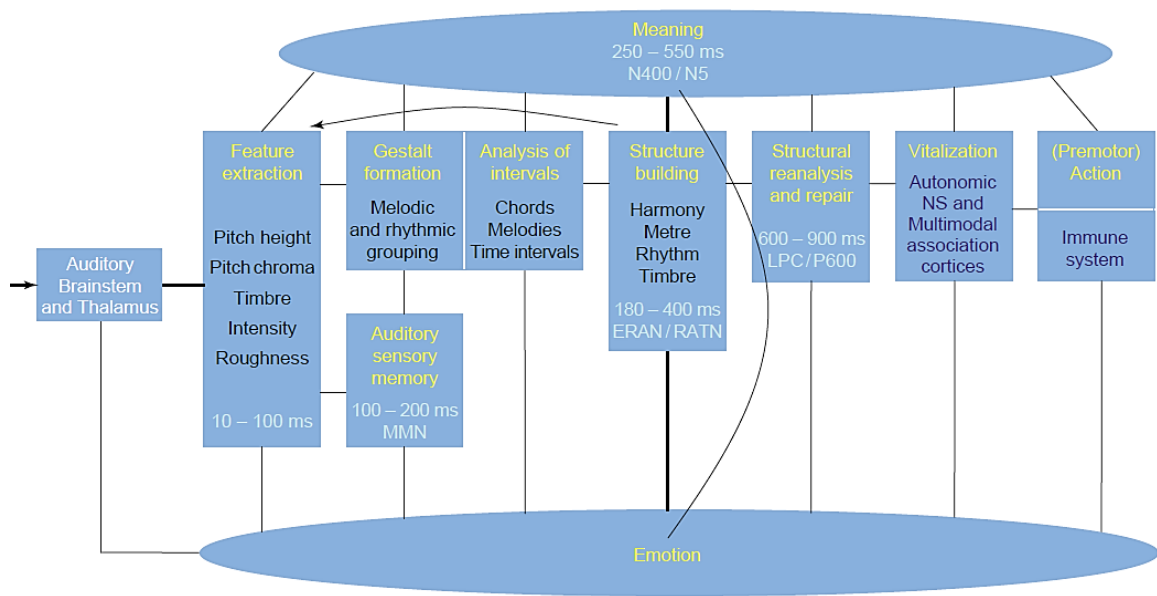


Figure 1: Neuro-cognitive model of music perception (Koelsch & Siebel, 2005)

Using machines to transcribe music with chord sequences and key information not only provides a useful compacted representation of a music piece but also facilitates upper-level analyses in the areas of summarization, segmentation, and classification (Chai, 2005). These three areas have implication in music searches and applications for music information retrieval (MIR). In the area of music classification, tonal structure and harmonic progression are strongly related to the perceived emotion while similar chord sequences are often observed in songs that are close in genre; therefore, they are good features for classifying music in terms of their emotion or genre (Cheng, et al., 2008; Anglade, et al., 2009). Koelsch and Siebel (2005) also state that “structurally irregular musical events, such as irregular chord functions, can elicit emotional (or affective) responses such as surprise; a fact that is used by composers as a means of expression.”

Summarization and segmentation are two sides of the same coin for music structural analysis where the summarized representation, as chord progressions, can also help segment a music piece into parts such as intro, chorus, refrain, bridge, and outro. Proper segmentation of a music piece can also improve the search process if the end user has high confidence in terms of the “segment” of his approximate query (Noland & Sandler, 2009). Following this train of thought, we propose a novel music segmentation mechanism in Chapter 5.

1.2 Research Goals

The tasks of analyzing tonality and harmony are very much related for tonal music since knowing the key of a music piece greatly helps the determination of chords and vice versa. We review this relationship in more detail in Chapter 2. However, analyses of keys and chords of a music piece are subjective and two analysts will not necessarily analyze a music piece exactly the same way (de Clercq & Temperley, 2011). With regard to key analysis, some musicians might hear a modulation in many sections of the piece while others might not. This kind of disagreement is even more pronounced in chord analysis – is it a major or minor triad when we can only detect the root and the fifth of a chord or should we label a section with a minor or seventh chord? Therefore, we propose to use a probabilistic framework to address uncertainties where latent variables – keys and chords – are estimated using a generative process and sampling techniques. Furthermore, we aim to bypass the model selection problem typically encountered in various machine learning

paradigms by having the target music “speak for itself” instead of using predetermined model parameters.

We approach the two tasks (key and chord recognition) using machine learning techniques. In a supervised learning setting, properly labeled training data (annotated keys and chords, in our case) are used to train a classifier so that it is capable of giving labels, i.e., keys or chords, to a given music piece. For unsupervised learning, there is no training data involved; it simply clusters sections of musical notes with the same characteristics such as those belonging to the same modulations or chords without giving them specific labels. The main differentiators between these two paradigms are model training and specifics of output labels. In our case, we argue that supervised learning is not suitable for music due to the scarcity of labeled training data which leads to the high possibility of over-fitted supervised models. Therefore, it would be more desirable to directly perform the two tasks on a target music piece in an unsupervised manner. However, a pure clustering-based unsupervised learning method (clustering musical notes into key and chord segments) is also undesirable since the goal of analyzing the tonality and harmony of a target music piece is to output specific key and chord labels. Thus, a better fit for our purpose is unsupervised learning guided by constraints which, in our case, is to use the unsupervised learning as a framework but incorporating relevant music theory into the framework so that it is capable of outputting the correct key and chord labels.

We test the key and chord recognition algorithm of popular music in both symbolic form and real audio recordings. Symbolic music in the format of MIDI (Musical Instrument Digital Interface) is event-based which contains all information that is necessary for machines to communicate and hence, generate the prescribed music as specified in the symbolic format. Real audio recordings are those stored on CDs (compact discs) as musical albums which can be played by CD players. Music from audio CDs can be extracted and converted to Waveform Audio file format (WAV) which contains a sequence of samples of audio sound waves. We test our proposed key and chord recognition algorithm with the above two data formats.

To summarize, our research goals are to develop a novel method to recognize keys and chords of symbolic and real music. Specifically, we aim to achieve the following:

1. Simultaneously recognize keys and chords of a music piece
2. Lay a foundation of using harmony for music segmentation and structural analysis
3. Adopt an unsupervised learning method to avoid the use of labeled training data
4. Use a probabilistic framework to address issues of uncertainties

1.3 Thesis Organization

Chapter 2: Background and Related Work

We first review the fundamentals of music theory related to tonality and harmony as well as define musical terms that we use throughout this dissertation. Secondly, we review the

most commonly used signal processing techniques for extracting features that are useful for key and chord finding. Third, we discuss important previous work of key and chord recognition in symbolic and audio domain, concentrating on work after the year 2008. Finally, we review the concept and fundamentals of infinite mixtures, the basis for the infinite Gaussian mixtures that we employ to extract a bag of local keys.

Chapter 3: Methodology

In the beginning of Chapter 3, we provide a “roadmap” of the methodology that outlines the contribution of each component to the overall tasks of key and chord finding. Since, in our method, extracting a bag of local keys using an infinite Gaussian mixture is a common component for the symbolic and audio track, we first concentrate on discussing the specifications of the model in the musical context. After the common thread is explored, we divide the discussion into two tracks – symbolic and audio – and provide specific treatment for each musical data format. In our discussion, we put more emphasis on the audio track due to its ubiquitous dominance in real audio recordings that we hear every day. Specifically, we discuss a wavelet based signal processing technique that we adopt in “regularizing” the raw audio signals before useful features are extracted. We conclude this chapter with a discussion on evaluation mechanisms for key and chord recognition in the symbolic and audio domains.

Chapter 4: Experimental Results

The dataset that we use is from the Beatles’ 12 albums of 175 songs. Therefore, at the beginning of this chapter, we describe the characteristics of the recordings in terms of their keys and chords. We move on to discuss our experimental results for the symbolic and audio tracks, respectively. Since the symbolic versions of the Beatles’ music are certainly different from the original Beatles’ recordings in terms of their audio content and length, experiments performed on the MIDI files are primarily served to improve the extraction of local keys for real audio files. Emphasis is placed on the audio track and the performance of various audio features are analyzed and compared.

Chapter 5: Applications and Extensions

With the ability to extract keys and chords described in the previous chapters, we propose a segmentation method based on “harmonic rhythm” that only involves the extracted tonal and harmonic information. Five dimensions – texture, phenomenal, root, density, and function – of harmonic rhythm are discussed in terms of how they can be used as segmentation cues. We further discuss the possibility of turning the segmentation boundary recognition problem into a change detection using a non-parametric martingale based method.

Chapter 6: Conclusions and Future Work

In this final chapter, we summarize the work we performed and highlight the main contributions of this undertaking. Future direction of improving the framework to turn a

bag of local keys into local key recognition on a frame-by-frame basis as well as future work for music structural segmentation is discussed.

1.4 Contributions and Publications

The thesis is organized based on the following three publications:

- Wang, Y.-S. & Wechsler, H. Musical keys and chords recognition using unsupervised learning with infinite Gaussian mixture. *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, ICMR 2012, Hong Kong, China.*
- Wang, Y.-S. Toward segmentation of popular music. *Proceedings of the 3rd ACM International Conference on Multimedia Retrieval, ICMR 2013, Dallas, Texas, USA.*
- Wang, Y.-S. & Wechsler, H. Unsupervised Audio Key and Chord Recognition. *Proceedings of the 16th International Conference on Digital Audio Effects, DAFx 2013, Maynooth, Ireland.*

Specifically, we make four contributions to the music signal processing and music information processing communities:

1. We have shown that using undecimated wavelet transform on raw audio signals improves the quality of the pitch class profiles.
2. We have demonstrated that an infinite Gaussian mixture can be used to efficiently generate a bag of local keys for a music piece.

3. We have ascertained that the combination of well-known tonal profiles and a bag of local keys can be used to adjust the pitch class profiles for harmony analysis.
4. We have shown that an unsupervised chord recognition system – without any training data as well as other musical elements – can perform as well, if not exceed, many of its supervised counterparts.

Chapter 2 Background and Related Work

In this chapter, we review the fundamentals of music theory and musical terms that are pertinent to the discussion of this dissertation as well as previous work in key and chord recognition. In Section 2.1, the relationship between frequency and pitch is covered, followed by the discussion of tonality (key) and how harmony (chord) is constructed under a tonal center. Section 2.2 reviews the most commonly used signal processing method for analyzing tonal harmony. Starting with one of the earliest models proposed by Jamshed Bharucha, we review, in Section 2.3, methods proposed in the literature while putting emphasis on more recent work since year 2008. In the last section of this chapter, we review early work of mixture models to lay the foundation for more in-depth model discussion at the beginning of Chapter 3.

2.1 Musical Fundamentals

2.1.1 Pitch and Frequency

From the Columbia Electronic Encyclopedia, 6th Edition, pitch is defined as the following:

Pitch, in music, the position of a tone in the musical scale, today designated by a letter name and determined by the frequency of vibration of the source of the tone. Pitch is an attribute of every musical tone; the fundamental or first harmonic, of any tone is perceived as its pitch. The earliest successful attempt to standardize pitch was made in 1858, when a commission of musicians and scientists appointed by the French government settled upon an A of 435 cycles per second; this standard was adopted by an international conference at Vienna in 1889. In the United States, however, the prevailing standard is an A of 440 cycles per second.

Based on the above definition, we see that three musical terms – musical scale, fundamental frequency, and harmonic – play an integral role in defining pitch and its relationship to frequency. A musical scale, explained in detail in Section 2.1.2, is a set of musical notes ordered by fundamental frequency (f_0) which is defined as the lowest frequency of a periodic waveform. The f_0 of each piano note is depicted in the bottom of Figure 2. Since sounds generated by musical instruments or human voices are rarely pure tones – those with one sinusoidal waveform of a single frequency – but a mixture of harmonics or overtones of twice, three, or n times of the fundamental frequency, such mixture of harmonics give rise to “timbre.” Timbre, also known as tone color, characterizes a unique mix of harmonics which allows us to distinguish different voices or sound produced by human or musical instruments. In general, periodicity – a periodic acoustic pressure variation with time – is the most important determinant of whether a sound is perceived to have a pitch or not. Therefore, pitched sounds, when represented in waveform (time domain), are periodic with regular repetitions while non-pitched sounds

Tonality, in music, quality by which all tones of a composition are heard in relation to a central tone called the keynote or tonic. In music that has harmony the terms key and tonality are practically synonymous, embracing a hierarchy of constituent chords, and a hierarchy of related keys.

Atonality, in music, systematic avoidance of harmonic or melodic reference to tonal centers (see key). The term is used to designate a method of composition in which the composer has deliberately rejected the principle of tonality.

From the above definitions, three terms – tonal center (central tone, tonic), hierarchy, and harmony (harmonic) – appear at least twice so we will first discuss them to see how they relate to tonality. A tonic is the most important and stable tone in which a music piece typically “resolves to” at the end or otherwise it gives the listeners the feeling of “unresolved” tension. Centering at the tonic, other tones form a hierarchy of pitches that are most frequently used and such hierarchy indicates the functions of different tones and their importance to the tonal center. Such musical relations within the hierarchy of pitches and tonal stability enable a listener to perceive and appreciate tension and release from a music piece. Harmony is the use of simultaneous tones which form varieties of chords and is one of the key ingredients in polyphonic music. Similar to the tonic of a music piece, chords and their progression create tension or resolution throughout the music piece. Though we have not finished the discussion of for key and chord, it should be clear that the tasks of extracting them (tonality and harmony) only apply to tonal music and therefore, we will not discuss atonality in this dissertation. The

remaining of the section provides more background information and concepts related to keys.

The key of a music piece contains two elements: tonic (discussed above) and mode. The mode of a key -- major or minor – are frequently referred in the title of classical music such as “Minuet in G Major” by Bach where the tonic is G and mode is major so the overall key is G Major. The most important distinguishing factor between a major and minor mode is the presence of major-third or minor-third interval above the tonic. A major third interval spans four semitones while a minor third consists of three semitones. The concept of intervals and semitones in a major or minor mode can be fully explained through major or minor scales, respectively. A major scale is defined by the interval pattern of T-T-S-T-T-T-S where T stands for whole tones and S stands for semitones. A whole tone is comprised of two semitones. Figure 3 depicts the C-major scale where C is the tonic with a major third (four semitones from the tonic C to E).

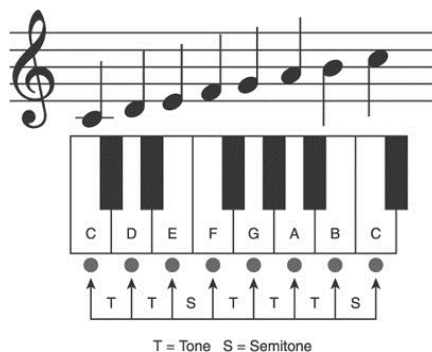





Figure 3: C major scale

There are three minor scales: natural, harmonic and melodic minor, all of which have a minor-third interval above the tonic. We summarize the interval patterns of the three minor scales in Table 1.

Table 1: Natural, harmonic and melodic Minor scales

C Minor Scale	Staff Notation	Intervals
Natural		T-S-T-T-S-T-T T-T-S-T-T-S-T
Harmonic		T-S-T-T-S-T+S-S S-T+S-S-T-T-S-T
Melodic		T-S-T-T-T-T-S T-T-S-T-T-S-T

A chord is a set of two or more notes that are played simultaneously or sequentially. The cardinality of chords, using C as the root, can be visualized in Figure 4. The most frequently used chords are triads which consist of three distinct pitch classes. A pitch class is a set of pitches or notes that are an integer number of octave apart. An example that two notes (C4 and C5) are one octave apart but belong to the same pitch class (C) is described in Figure 5. Since an octave contains 12 semitones, we use integer notation, starting from 1 to 12 where degree 1 indicates the root pitch class, to describe pitch classes as whole numbers. Such integer notation represents the scale degree of a particular note in relation to the tonic. The tonic is considered to be the first degree of the scale.

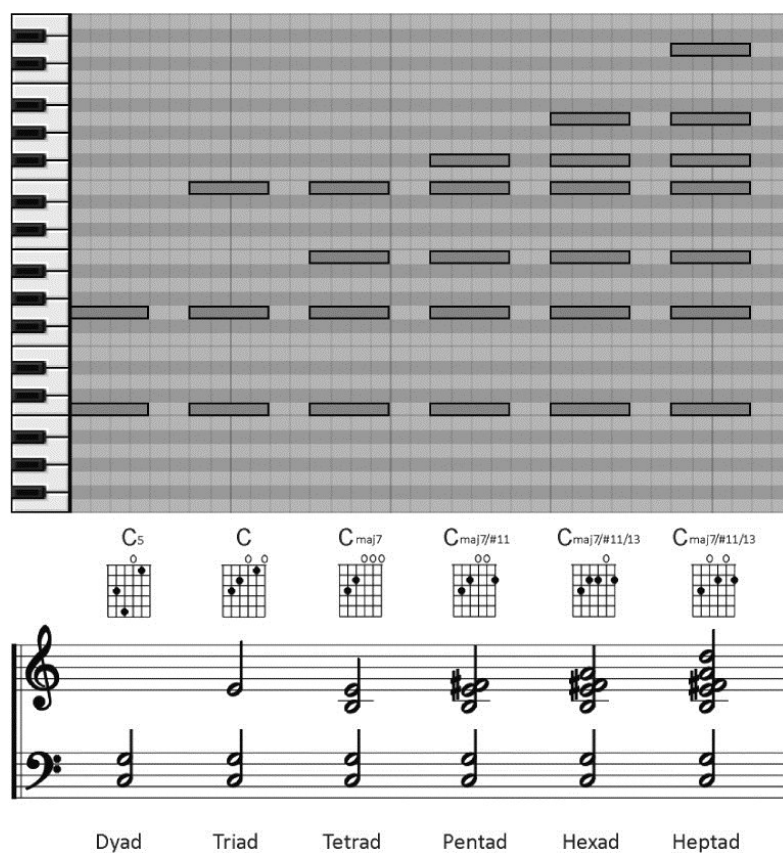


Figure 4: Cardinality of chords (Hewitt, 2010)

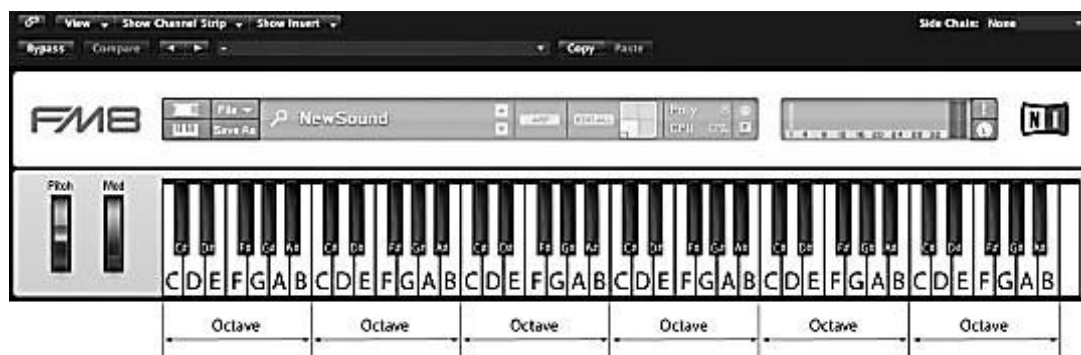


Figure 5: Octave and pitch classes. Each letter on the keyboard represents the pitch class of the tone (Snoman, 2013).

Using the 12 semitones within an octave, an interval is the distance from the root to each semitone. The root of a chord is the pitch upon which other pitches are stacked against to form a chord. For example, the root of an F-major chord is F pitch while the root of E-minor chord is the E pitch. Figure 6 tabulates and gives names of all intervals within an octave that we use to discuss the formation of chords.

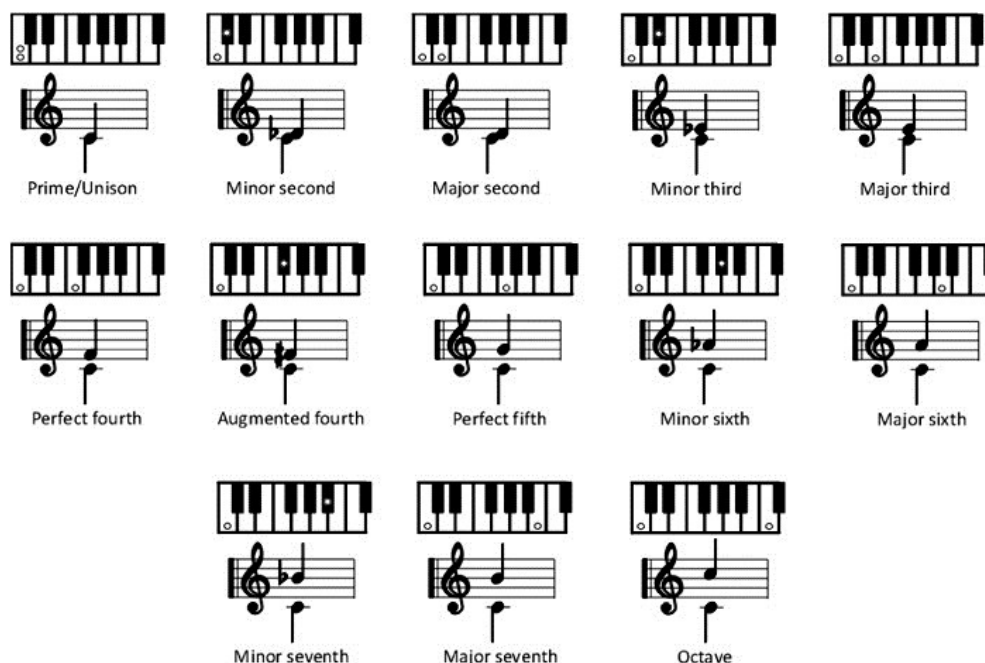


Figure 6: Names of musical intervals (Hewitt, 2010)

We will limit our review to five types of chords, namely, major, minor, diminished, augmented, and suspended (2nd and 4th), which our chord detection task mostly focuses on in this dissertation. These five types of chords all consist of three pitch classes. Table 2 summarizes the intervals that make up the five types of chords and

illustrates examples with roots in C pitch class using staff notation. Figure 7 and Figure 8 depict the five types of triads with C as root using piano roll, guitar fret board, and staff notation.

Table 2: Formation of triads

Name	Intervals
Major	Root, major 3 rd , and perfect 5 th
Minor	Root, minor 3 rd , and perfect 5 th
Diminished	Root, minor 3 rd , and diminished 5 th (augmented 4 th)
Augmented	Root, major 3 rd , and augmented 5 th
Suspended 4 th	Root, perfect 4 th , and perfect 5 th
Suspended 2 nd	Root, 2 nd , and perfect 5 th

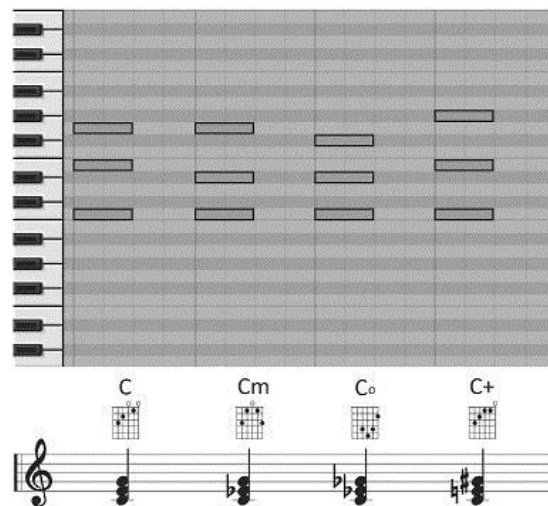


Figure 7: Notation of C major, minor, diminished, augmented chords (Hewitt, 2010)

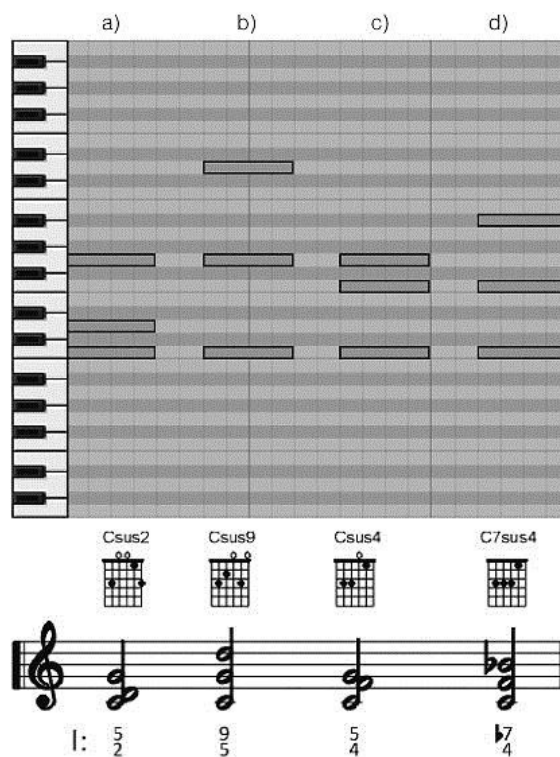


Figure 8: Four types of suspended triads with c as the root (Hewitt, 2010)

Other than the notations described above, musicians often use Roman numerals to denote triads within a major or minor key of their respective scale (collectively we denote as diatonic scales) as described in Figure 3 and Table 1. A triad is of the n th degree when the root of the chord is the n th degree note of the diatonic scale employed by the music piece. Therefore, triads formed within the diatonic scale are called in-key chords. For example, the C major and F major triads in a music piece with the key of C major is denoted as Roman numerals ‘I’ and ‘IV’ respectively since its root is the tonic and fourth degree of the C major scale. The most important in-key triad is the tonic chord which is the first degree chord (“I” chord) and it is the best representative chord of the key for

three reasons. First, the root tone of the chord is the also the root tone of the key. Second, the tonic chord contains the perfect fifth interval (such as the G in C major chord) which is also the third harmonics of the root tone of the key. Third, and most importantly, the tonic chord contains the third of the key – three intervals (minor third) or four intervals (major third) above the tonic – which determines the mode of the key (minor or major).

2.1.3 Chroma and Key Profiles

According to Revesz and Shepard, a pitch has two dimensions: tone height and chroma. Tone height is the sense of high and low pitch while chroma refers to the position of a tone within an octave (Loy, 2006, p. 163). Figure 9 (a) and (b) visualize the concept of tone height and chromatic circle (abbreviated chroma) where the chroma circle is the projection of tone height along the y-axis. The concept of chroma is the same as that of a Pitch Class depicted in Figure 5. Due to human ears' logarithmic frequency sensitivity, the tone height component is represented using the logarithm of the frequency of a pitch. In the chroma circle, neighboring pitches are a tonal half step apart which we refer to as "semitone" in Figure 3. Circle of Fifth (CoF), as depicted in Figure 9 (c), represents musically significant intervals, such as perfect fifth (clock-wise) and perfect fourth (counter-clockwise). CoF is often used to measure "distances", such as Lerdahl's distance (Lerdahl, 2001), among different keys as well as explain the concept of consonance and dissonance for chord formations, dated back to as early as Pythagoras' time (Benson,

2007). Perfect fifth and perfect fourth have a frequency ratio -- all of them simple ratio -- of 3:2 and 4:3, respectively, while notes of an octave apart has a simple ratio of 2:1.

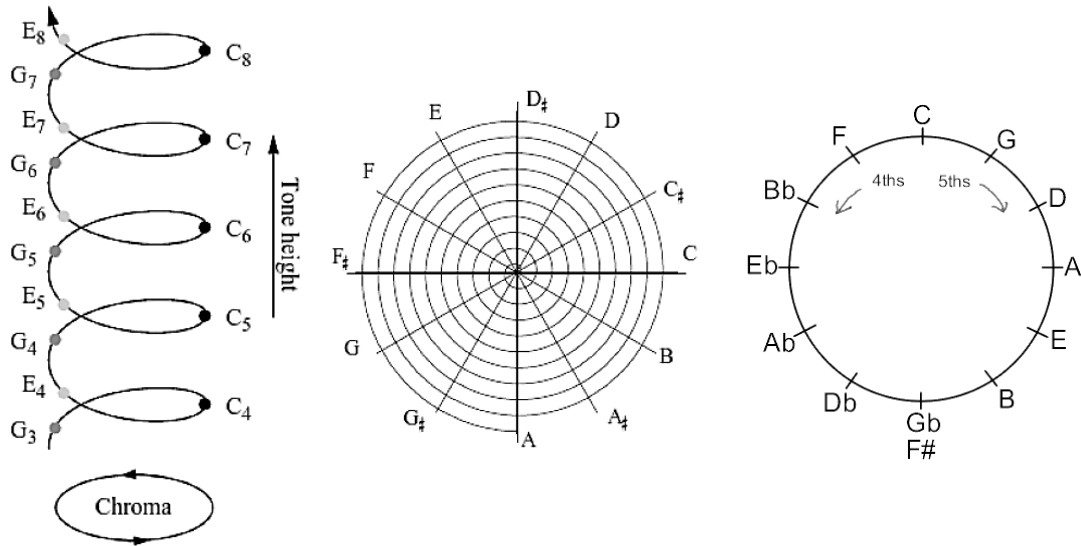


Figure 9: (a) Pitch tone height; (b) Chroma circle; and (c) Circle of Fifth; ((a) and (b) are from Loy, D. (2006, pp. 164-165))

The most influential key-finding work was developed by Krumhansl and Schmuckler (Krumhansl, 1990) which is widely known as K-S key-finding algorithm. The algorithm uses a set of 12 major and 12 minor key profiles, depicted in Figure 10, developed by Krumhansl and Kessler (Krumhansl & Kessler, 1982). Ranking values of these profiles describe how well the probe-tone “fits” in the context on a scale of one to seven where higher values represent better goodness-of-fit in terms of stability and compatibility. Many key and chord finding implementations are based on the K-S algorithm and K-K profiles where target music pieces are encoded as a 12-dimensioned

vector to be compared with these 24 key profiles. The key profile that best correlates with the target 12-dimensional vector is the found key.

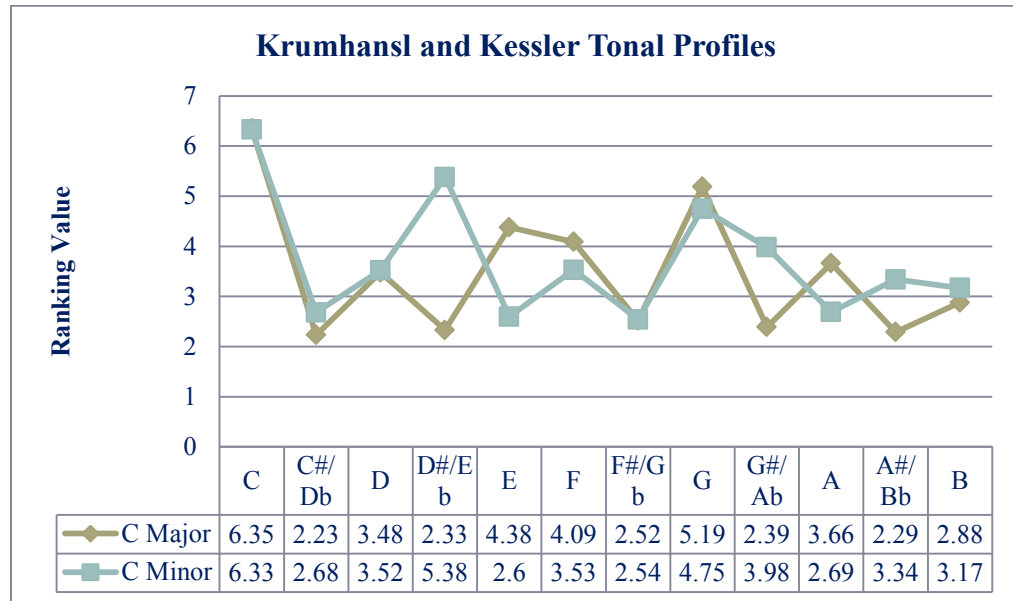


Figure 10: Krumhansl and Kessler major and minor profiles

Instead of gathering responses to the probe-tone from listeners as a way to represent each tone's ranking in a tonal structure, Temperley (2007) uses the Kostka-Payne corpus of 46 musical excerpts to determine each scale degree's presence, using probability distributions, in major and minor scales in the corpus. For example, scale degree 1 (the tonic) and scale degree 7 occur in 74.8% and 40% of the segments in major scales, respectively. The Temperley tonal profile is depicted in Figure 11.

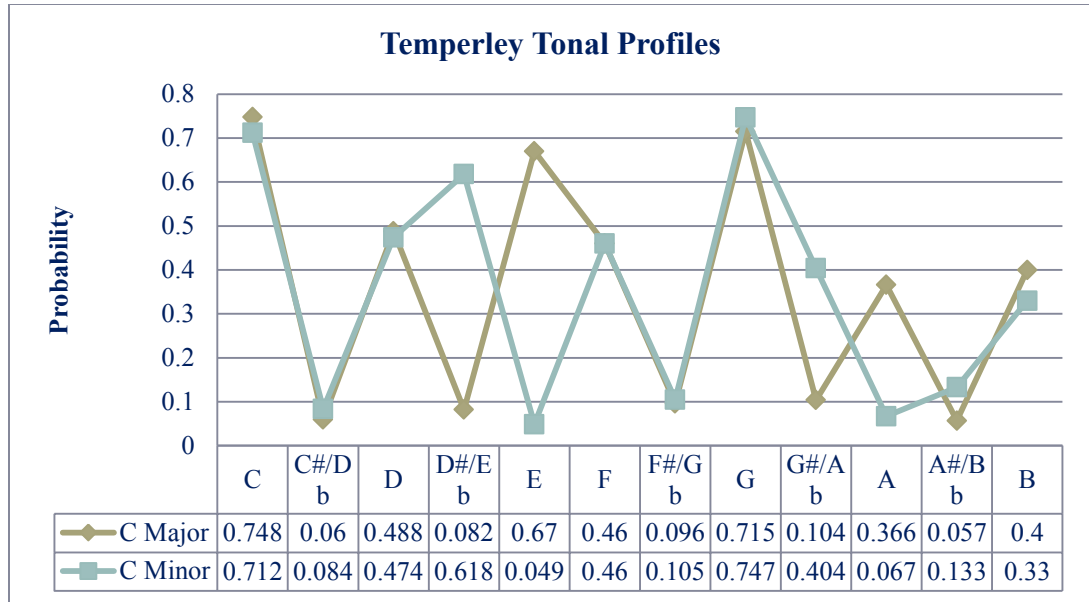


Figure 11: Temperley key profiles

2.2 Music Signal Processing and Previous Work

The symbolic representation (i.e. MIDI) of music, similar to a musical score composed by a composer, contains explicit information of musical notes played by computers. Since the 1970s, much of the tonal or harmonic analyses have been performed on the symbolically notated western classical music which we review in Section 2.3. Due to the differences between the data format of symbolic and waveform audio music, a signal processing front end is required to transform the raw audio waves into a format suitable for the tasks at hand. For key and chord analysis, the most popular format is a chromagram, also known as chroma vectors or Pitch Class Profile (PCP), which is a frame-by-frame chroma-based representation of the target music piece. In this section, we

review the most commonly used signal processing techniques to extract the PCP. Figure 12 depicts the general framework of a two-stage process to convert waveform audio signals to a frame-by-frame chromagram. In our discussion of specific methods of the signal processing front end, we mainly follow the notation used in (Loy, 2007).

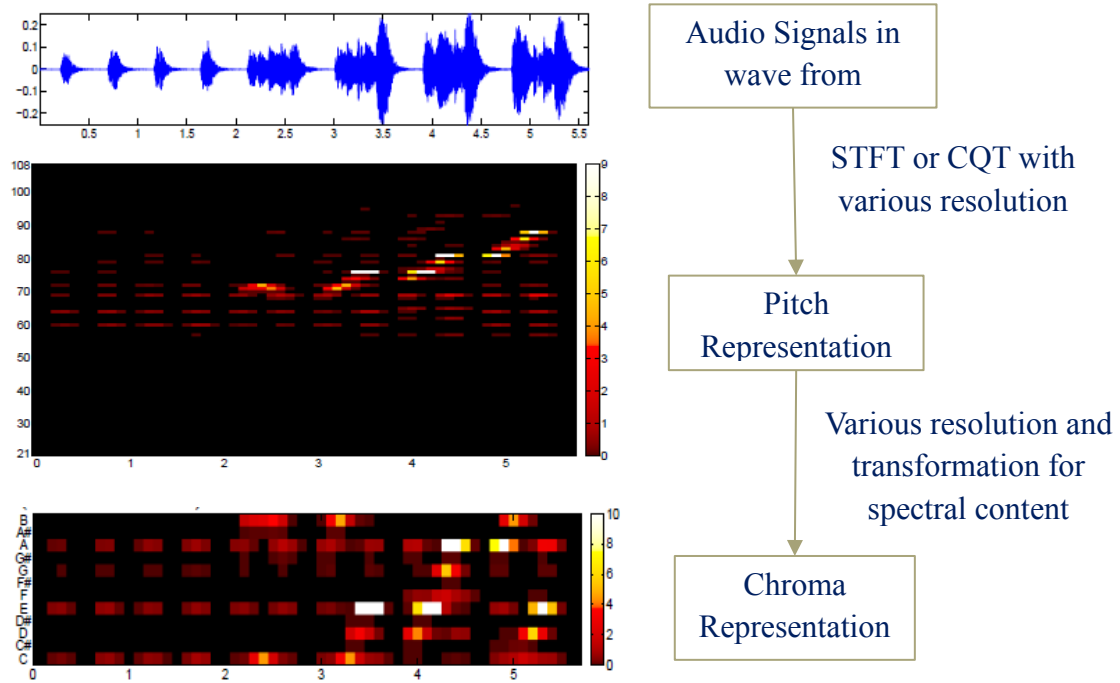


Figure 12: Framework of chromagram transformation (diagram extracted from (Müller & Ewert, 2011))

The first stage transforms signals from the time domain into frequency domain using discrete Short-Time Fourier Transform (STFT) which splits the sampled input signals, $x(i)$, into successive block of frames of size N and hop size r . Equation 1 describes the STFT and Table 3 lists a few commonly used STFP specifications.

Equation 1: Short-term Fourier transform

$$X_k^{STFT}(sR) = \sum_{r=-\infty}^{\infty} x(r)w(sR - r)e^{-j2\pi kr/N}$$

where k indexes discrete frequency over the range of $0 \leq k < N$, s denotes the index of the analysis frame, and $w(\cdot)$ is a suitable windowing function.

Table 3: Previous work and commonly used STFT specification

	Analysis Type	Analysis Window	Frame Size	Sampling Rate	Hop Size
Sheh and Ellis (2003)	Harmony and Segmentation	Hann	4096	11025 Hz	100 ms
Gomez (2006)	Keys	Blackman Harris	4096	44.1 KHz	11 ms
Khadkevich and Omologo (2009)	Harmony and Segmentation	Hamming	2048	11025 Hz	185.7 ms

STFT is suitable for analyzing frequency resolution that is constant throughout the frequency range, i.e., it divides the spectrum of the sound into bins of constant bandwidth. However, due to human ears' logarithmic frequency sensitivity, the pitch perception of the ear is proportional to the logarithm of frequency rather than to the frequency itself. Therefore, the constant bandwidth of STFT overspecifies high

frequencies and underspecifies low frequencies. A Constant Q Transform (Brown, 1991) is designed so that the bandwidths of analysis bins, denoted as δf_k , increase in constant proportion to the center frequency, f_k , of each band which overcomes the insufficient frequency resolution for low frequencies. Quality Factor, abbreviated Q , is therefore defined as the ratio of the center frequency to the bandwidth of a bandpass filter. Furthermore, since a frequency ratio of two is a perceived pitch change of one octave and a semitone interval is $\sqrt[12]{2}$, we can express f_k in terms of the minimum center frequency f_{min} (such as C0 at 16.35Hz, see Figure 2) and the number of bins (β) per octave. The last piece of information that is required to complete the specification of CQT is the length of the analysis frame, $N(k)$, which can be determined by the sampling rate f_s , f_k , and Q . Equation 2, Equation 3, Equation 4, and Equation 5 describe CQT in a similar notation to that of STFT. Table 4 lists a few commonly used STFP specifications.

Equation 2: Constant Q transform

$$X_k^{CQT}(sR) = \sum_{r=0}^{N(k)-1} x(r)w(k,r)e^{-j2\pi f_k r}$$

Equation 3: Sampling rate determination

$$f_k = 2^{k/\beta} f_{min}$$

Equation 4: Q determination

$$Q = \frac{f_k}{\delta f_k}$$

Equation 5: Size of analysis frame

$$N(k) = \frac{f_s}{f_k} Q$$

Table 4: Previous work and commonly used CQT specification

	Analysis Type	f_{min}	f_{max}	β	Q	Sampling Rate	Hop Size
Bello and Pickens	Harmony and Segmentation	98 Hz	5250 Hz	36	51	11025Hz	1/8
Harte (2005)	Chord	110 Hz (A2)	1760 Hz (A6)	36	51	11025 Hz	1/8
Muller (2011)	Harmony	27.5 Hz (A0)	4186 Hz (C8)	72	25	High: 22050 Hz Middle: 4410 Hz Low: 882 Hz	1/2

The second stage is to sum up the energy level of pitch representation from the first stage into a two-dimensional chromagram based on Equation 6 (Lerch, 2012) where j represents the index of chroma (0 ~ 11) and n denotes the index of each analysis frame in Equation 7. They are frequently normed as described in Equation 8.

Equation 6: Chroma summation

$$v(j, n) = \sum_{o=o_l}^{o_u} \left(\frac{1}{k_u(o, j) - k_l(o, j) + 1} \sum_{k=k_l(o, j)}^{k_u(o, j)} |X(k, n)| \right)$$

Equation 7: Chroma vector

$$\mathbf{v}(n) = [v(0, n), v(1, n), v(2, n), \dots, v(11, n)]^T$$

Equation 8: Normalized chroma vector

$$\mathbf{v}_N(n) = \mathbf{v}(n) \cdot \sqrt{\frac{1}{\sum_{j=0}^{11} v(j, n)^2}}$$

where in Equation 6, o_l and o_u designate the indices of the first and last octaves in the pitch representation while $k_l(o, j)$ and $k_u(o, j)$ represent the low and high cut-off frequencies of a pitch band.

2.3 Previous Keys and Chords Analysis

Bharucha (1991), in the mid-1980s, proposed the earliest complete system, an artificial neural network (ANN) called MUSACT, to extract tonality and harmonic content from audio signals. Specifically, it extracts chords from tones and keys from chords. Since the majority of systems proposed in recent years and those in the past decade exhibit similar

components and characteristics, we will use Bharucha’s model, to be discussed in Section 2.3.1, as a baseline in reviewing recent work. Section 2.3.2 summarizes important work since the late 1990s. In Section 2.3.3, we concentrate our review on relevant research published after 2008 and draw commonalities and differences based on the Bharucha’s model when pertinent.

2.3.1 Bharucha’s Model

Figure 13 depicts Bharucha’s model where Spectral Representation (component a) is reviewed in Sections 2.1.1 and 2.2, Pitch Height (component b) and Pitch Class (component c) are discussed in Section 2.1.3, and Pitch Class Clusters (component d) and Tonal Centers (component e) are described in Section 2.1.2. The Gating mechanism (component f) takes pitch-class information and tonal center (key) to transform them into a pitch-invariant representation so that the tonic is always “0” in a 12-dimensioned vector representing a musical sequence. The invariant pitch-class representation supports the encoding of sequences into a sequential memory (component h). In other words, all musical sequences are normalized into a common set of invariant pitch categories indexed by a chroma vector $\{0, 1, 2, 3, \dots, 10, 11\}$ where the first index denotes the tonic or key. Figure 13 depicts the network of tones, chords, and keys in his model while Figure 15 describes the gating mechanism.

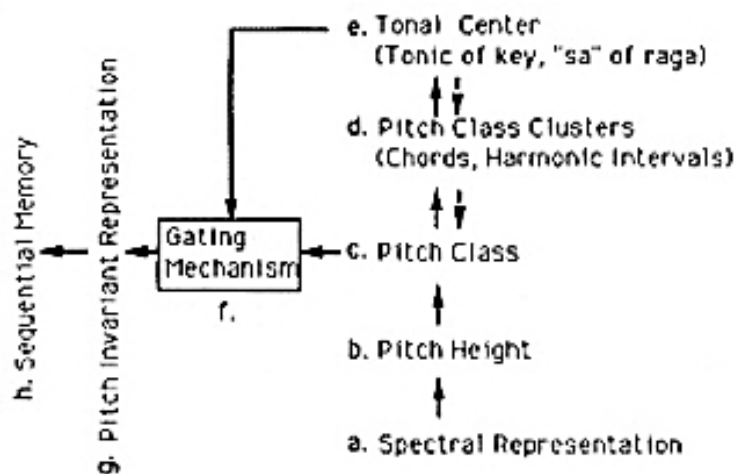


Figure 13: Bharucha's model (1991, p. 93)

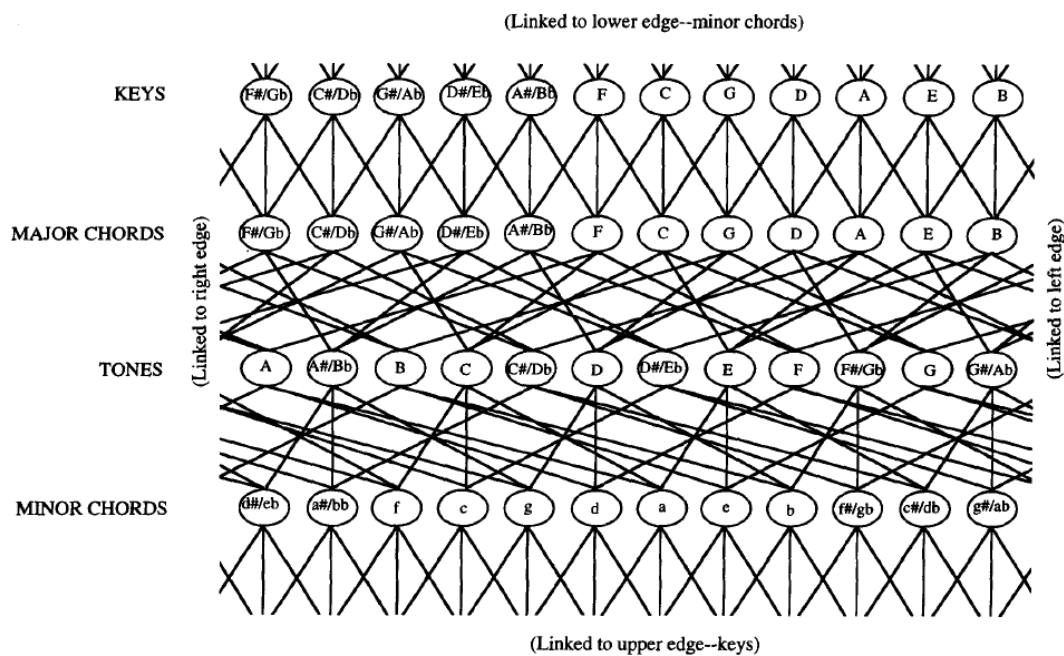


Figure 14: Network of tones, chords, and keys (Bharucha, 1991, p. 97)

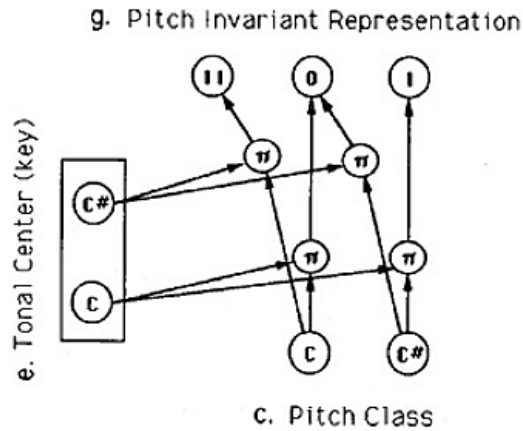


Figure 15: Gating mechanism to derive pitch invariant representation (Bharucha, 1991, p. 97)

According to Bharucha and Todd (1991, p. 128), two forms of tonal expectancy – schematic and veridical – can be modeled by the sequential memory (component h in Figure 13). Schematic expectancies are “culturally based structures which indicate events typically following familiar contexts,” while veridical expectancies are “instance-based structures indicating the particular event that follows a particular known context.” The schematic and veridical expectancies correspond, more or less, to the cultural and sensory aspects of tonal semantics – a system of relations and meanings between tones within a context – as described by Leman (1991, p. 100). The sensory aspect relates to the sounds and acoustical stimulus processed by our auditory system where as the cultural aspect “captures what is added by the cultural character of the music and by learning processes of the listener with respect to this character.” Furthermore, Bharucha and Todd describe the potential conflicts between the two expectancies as the following.

“Schematic and veridical may conflict, since a specific piece of music may contain atypical events that do not match the more common cultural expectations. This conflict, which was attributed to Wittgenstein by Dowling and Harwood (1985), underlies the tension between what one expects and what one hears, and this tension plays a salient role in the aesthetics of music (Meyer 1956). Schematic expectancies are driven by structures that have abstracted regularities from a large number of specific sequences. Veridical expectancies are driven by encodings of specific sequences.”

Transition probabilities for the schematic and veridical expectancies of chord functions are embodied in the sequential memory. Bharucha and Todd further stated that *“the net will learn to match the conditional probability distributions of the sequence set to which it is exposed ... an example of such expectancy is that a tonic context chord generates strong expectation for the dominant and subdominant while supertonic context chord induces resolution to the dominant and submediant progressions.”* Though tonal expectancy, in terms of harmonic progressions, for common-practice music (European art music from 18th to 19th centuries) are generally agreeable among musicologists, the “rule” or “common pattern” of chord progression may not be readily available in pop or rock music which we will discuss in detail in Section 4.5.

2.3.2 Summary of Previous Work

We summarize previous work based on three characteristics: format of music data, supervised vs. unsupervised, and types of output. The approach of using machines to

extract keys and chords are typically categorized based on the format of the music data: raw audio signals or symbolic event-based signals. The former category requires signal processing techniques, which we reviewed in Section 2.2, to extract low-level features such as Pitch Class Profiles (PCP) or chroma vectors from the raw audio signals as a front end. The latter format contains discrete events such as MIDI that can be directly used for key and chord recognition. Since one of the distinguishing characteristics of our approach is the unsupervised machine learning approach, we categorize, rather loosely, previous literature into the two machine learning paradigms – supervised and unsupervised – in terms of their requirements on the use of training data. In other words, we categorize approaches that require training data as supervised methods while those that do not, including knowledge-based systems, as unsupervised. The third characteristic we examine in the proposed methods is whether keys (local vs. global) and chords are estimated simultaneously as well as the chord vocabulary involved in the recognition. Based on the above categorization, we enumerate previous relevant work in Table 5.

Table 5: Previous work of key and chord analysis

Researchers Year	(S)ymbolic or (A)udio	Supervised	Un-supervised	Global Key (GK) Local Keys (LK) Chords (C) with # of chord types in parenthesis
Pre-2005	Fujishima (1999) Wakefield (1999)	A	Two earliest work in proposing transforming audio signals into pitch-chroma representation (chromagram)	
	Raphael and Stoddard (2003)		Use HMM to label segments of MIDI music piece with keys and chords where they are simultaneously estimated; model parameters were trained from unlabeled MIDI files with rhythm and pitch	C(2)
	Sheh and Ellis (2003)	A	HMM-based chord model trained using EM; single 24-dimension Gaussian; Viterbi algorithm for chord labeling	C(2)
	Pauws (2004)	A	Key profile matching & human auditory modeling	GK
2005	Zhu, Kankanhalli, and Gao (2005)	A	Apply tone structures and clustering to estimate diatonic scale root and keys from extracted pitch profile	GK
	Chuan and Chew (2005)	A	Spiral Array model and Center of Effect Generator (CEG)	GK
	Chai and Vercoe (2005)	A	12-state HMM for key 2-state HMM for mode; Relative keys grouped first; detect modes second; Music theory based HMM parameter specification	LK

	Bello and Pickens (2005)	A	HMM-based method; mid-level representation of harmonic and rhythmic information	C(2)
2006	Gómez (2006)	A	Introduced Harmonic PCP (HPCP) which increases resolution in frequency bins with weighted harmonic content; Employed K&K and Temperley key profiles	GK
2007	Izmirli (2007)	A	Extracted chromagram are segmented using non-negative matrix factorization; global and local keys are found using K-S key finding	LK
	Rhodes, Lewis, and Mullensiefen (2007)	S	Bayesian based model selection and Dirichlet distributions for pitch-class proportions in chords	C(5)
2008	Ryynanen and Klapuri (2008)	A	Chord model: 24-state HMM; Note model: 3-state HMM; noise-or-silence model: 3-state HMM; Viterbi algorithm is used to determine note and chord transition; Melody and bass notes are estimated	GK + C(2)
	Weil, Sikora, Durrieu, and Richard (2009)	A	24-state HMM as chord model; employ a beat-synchronous framework; also estimate melody	GK + C(2)
	Cheng, Yang, Lin, Liao, and Chen (2008)		Acoustic modeling: HMM; Language modeling: N-gram; Chord decoding: calculate maximum likelihood against chord templates	
	Lee and Slaney (2008)		Use synthesized symbolic data to train key-dependent HMM;	GK + C(2)

			a global key is estimated; chord sequence is obtained by Viterbi algorithm	
2009	Khadkevich and Omologo (2009)	A	PCP features are used to train 24-state HMM; labeled chord sequence are used to train N-gram language model; beat tracking utilized	C(2)
	Hu and Saul (2009) (Hu, 2012)	S/ A		Latent Dirichlet Allocation (LDA) for both symbolic and audio data; use Mauch' NNLS chroma features; audio data is synthesized from MIDI LK+C(2)
	Weller, Ellis, and Jebara (2009)	A	Replace a generative HMM with a discriminative SVM	C(3)
2010	Mauch and Dixon (2010)	A		Dynamic Bayesian network / GMM for features; all parameters and conditional probability distributions are manually specified GK + C(4)
	Ueda, Uchiyama, Ono, and Sagayam (2010)	A	Use harmonic / percussive sound separation (HPSS) to suppress percussive sound;	LK + C(2)
	Rocher, Robine, Hanna, and Oudre (2010)	A		Harmonic candidates consist of chord/key pairs; use binary chord templates and Temperley key templates; Use Lerdahl's distance and weighted acyclic harmonic graph to select best candidate; Dynamic programming involved LK + C(2)
2011	Cho and Bello (2011)	A	Smooth DCT-based chromagram by time-delay embedding and recurrence plot; GMM and binary chord template are used	C(3)
	Oudre,	A		Template (binary) based C(3)

	Fevotte, and Grenier (2011)		probabilistic framework using EM; used Kullback-Leibler divergence to measure the similarity between chromagram and chord templates	
	Pauwels, Martens, and Peeters (2011)	A	Knowledge based: Local key acoustic model + binary chord template; Lerdahl' tonal distance metric; Dynamic programming search	LK + C(4)
	Lin, Lee, and Peng (2011)	S	Use Artificial Neural Networks (ANN) trained by Particle Swarm Optimization (PSO) and Backpropagation (BP)	C(1):3 maj chord
2012	Itoyama, Ogata, and Okuno (2012)	A	Adopt Markov process for chord sequence, Gaussian mixture for feature distribution, and Pitman-Yor language model for chord transition; Joint posterior probability of chord sequence, key, and bass pitch estimated	C(4)
	Papadopoulos and Peeters (2012)	A	HMM based; key progression is estimated from chord progression and metrical structure; analysis window length is adapted to the target music piece	LK
	de Haas, Magalhaes, and Wiering (de Haas, et al., 2012)		Knowledge-based tonal harmony model; Use Mauch's beat-synchronized NNLS chroma; Use K-S key profiles for key finding and involve dynamic programming	LK + C(3)
	Ni, Mcvicar, Stantos-	A	Beat tracking + Loudness based treble	GK + C(11)

MIREX¹ (Music Information Retrieval Evaluation eXchange) formalized the chord audio detection test in 2008 and many significant work of key and chord recognition have been published through different channels. Since not all proposed systems in the literature participated in MIREX's tasks and many of those who participated submitted multiple versions for competition, it is difficult to determine the exact number of publications. However, to gain a basic understanding of different methods as well as types of keys or chords they aim to estimate, we broadly survey the existing literature after 2008 and categorize them in Table 6. Though we do not claim that the table includes an exhaustive and complete categorization of the existing literature, we do see certain subcategories that are more popular than others. First, the supervised methods are more popular than their unsupervised counterpart. Second, the majority of chord estimation covers only the major and minor chord types. Third, though keys and chords are closely related aspects of tonal harmony, the majority of the proposed methods do not estimate them simultaneously.

¹ http://music-ir.org/mirex/wiki/MIREX_HOME

Table 6: Publication count for key and chord analysis since 2008

Category	Sub category	# of Publication
Machine Learning	Supervised	29
	Unsupervised	21
Signals	Audio	43
	Symbolic	7
Keys	Global	14
	Local	10
Triad Chords	major + minor	21
	major + minor + N	10
	major + minor + augmented + suspended	5
	major + minor + augmented + suspended + N	2
Key + Chords	Global key + chords	8
	Local keys + chords	7

In the above summary of previous work, we purposely concentrate only on comparing and contrasting mechanisms proposed in the literature, not their performance in terms of recognition rates of keys and chords nor the data sets employed in their experiments. This is due to the fact that many experimental results are obtained from datasets that, in many cases, are very different in terms of the number of musical pieces, type of music, as well as the types of keys or chords these proposed systems aim to recognize. Therefore, it is rather meaningless to report recognition rates that cannot be objectively compared. However, for methods that aim to estimate chords for pop music, the majority of them use the same training (for supervised approaches) and testing dataset – a collection of at most 217 popular songs – which is relatively small and highly unlikely representative of popular music. It is also unclear how much of these supervised mechanisms have been overfitted using the said dataset (de Haas, et al., 2012). However, in Section 4.4 Performance Comparison, we will provide details of more recent

experimental results which employ similar test dataset to that of ours; moreover, we will elaborate on the possibility of overfitting in supervised machine learning in Section 4.4.

2.3.3 Recent Work After 2008

Examining Bharucha’s model and previous work in Table 5, we notice that the majority of recently proposed methods highly resemble the Bharucha’s model. First, for proposed methods involving audio data, all have a spectral processing front end using one of the transformations described in Section 2.2. Second, extracted spectral content is transformed into Pitch Class representation and variants of the gating mechanism might be applied to produce invariant representation of pitch classes. Third, for the majority of the supervised learning approach summarized in Table 5, the prevalent HMM component is more or less similar to the Bharucha’s Sequential Memory component where conditional probabilities are obtained through learning.

In the system proposed by Ryyanen and Klapuri (2008), there are two major components – a chord transcription module and a note module. The chord transcription module uses a 24-state HMM for major and minor triads. Trained profiles for major and minor chords are used to compute the observation likelihood given those profiles. Between-chords transition probabilities are estimated from training data and Viterbi decoding is used to find the most likely chord progression. The note module utilizes three HMMs to model the three acoustic aspects – target notes, other notes, and noise-or-silence – of the music data. Melody and bass lines are modeled through the target-notes

module; the noise-or-silence models the ADSR (attack, decay, sustain, release) envelope which we explain in Section 3.4.1; all other sounds are modeled in the other-notes module. Conceptually, these two components are similar to the more simplified system proposed by Cheng et.al (2008) utilizing acoustic and language components. The acoustic component uses a 24-state HMM to model the low-level PCP feature vector to find a chord that best fits the perceived music in a short time interval. The language component employs an N-gram model to determine the best chord progression following the rules of harmony from the commonly-used progression patterns. One distinguishing characteristic of Cheng's system is that the Viterbi algorithm is not used in the chord decoding phase. Instead, the chosen chord and progression are determined by the maximum likelihood principle combining the language and acoustic components. Very similar to Cheng's system, the following year, Khadkevich and Omologo (2009) also proposed a system using HMM and language model (such as N-gram or factored language model, FLM) in which chord sequence is obtained by running a Viterbi decoder on trained HMM while taking the weight of the language model into consideration. Examining the three systems from a high level, the two components in each system appear to correspond quite nicely to Bharucha's schematic and veridical expectancies as described in Section 2.3.1.

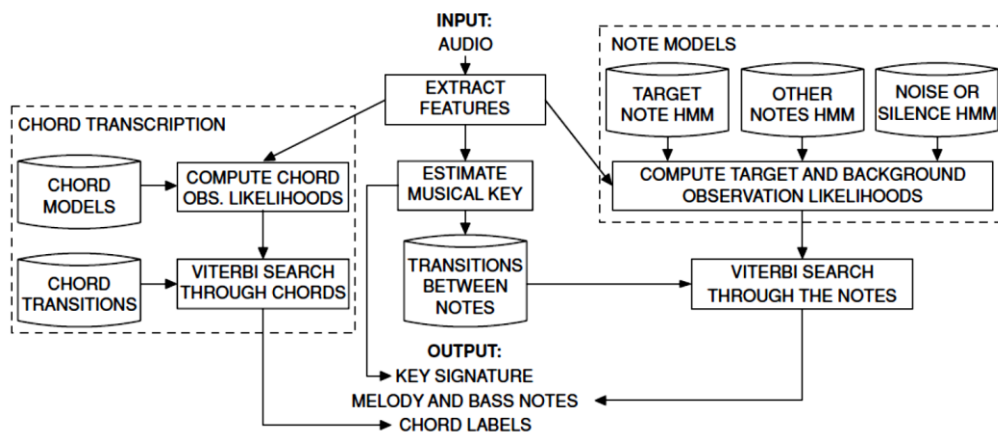


Figure 16: System developed by Ryynanen and Klapuri (2008)

Mauch and Dixon (2010) divide spectral content into bass and treble chromagrams as input to a dynamic Bayesian network (DBN) – a Bayesian network models event of time series – to simultaneously model many aspects of music. The DBN is constructed with six layers where the two observed layers model the bass and treble chroma vectors while the other four hidden source layers jointly model metric position, key, chord, and bass pitch classes. Figure 17 (a) depicts “two slices” of the model. In a typical scenario using the DBN, the conditional probability distribution for each node is estimated from the training data; however, even with simplified scenarios such as 4 metric positions, 12 unique key signatures, 48 chord types, and 12 bass pitch classes, the estimation and specification of the conditional probability distributions (CPD) – through training – for all the nodes in the network quickly becomes infeasible. As stated by Mauch (2010), “... we choose to map expert musical knowledge onto a probabilistic framework, rather than learning parameters from a specific data set. In a complex model such as the one presented in this section, the decisions regarding parameter binding

during learning, and even the choice of the parameters to be learned pose challenging research questions, ...” Due to the infeasibility of training the DBN, all CPDs are manually specified in this method. The other challenging aspect of utilizing such a model is the specification of the model structure which could be learned from adequate amount of training data to understand if there is any causal relationship between, for example, the metrical position and key or other nodes in the DBN. Since the model structure and CPDs are manually specified, we categorize this method as an unsupervised knowledge-based system.

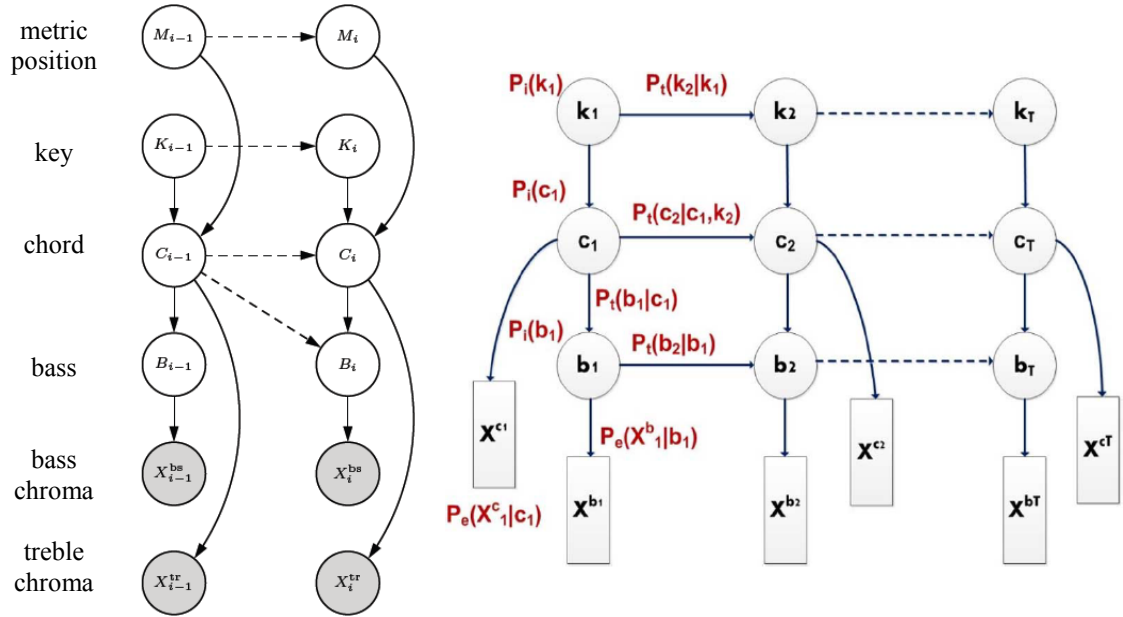


Figure 17: (a) Dynamic Bayesian network developed Mauch & Sandler (2010);
(b) DBN modified by Ni et al. (2012)

Ni et al. (2012) improved Mauch’s work in two ways. First, they extracted treble and bass chromagrams by taking human perception of loudness into account. Second,

instead of using expert knowledge for the specification of model parameters, the probabilities of key chord, bass and conditional probabilities specified in Figure 17 (b) are learned from the training dataset using maximum likelihood. However, in Ni's HMM, the metric position is not modeled. Furthermore, similar to Bharucha's model, they also adopted the technique of using pitch invariant representation, with the assumption that chord transitions are dependent on the tonal center, to increase the effective training data by 12 folds.

de Haas et al. (2012) proposed a system which uses Mauch's NNLS chroma features as input to a complete knowledge-based subsystem for local key finding and chord transcription without using any training data. The Euclidean distance between chroma features and a chord dictionary, consisting of major, minor, and dominant seventh, is calculated for each beat. If the distance between one particular chord candidate and the chroma frame is sufficiently shorter than other candidates, the candidate chord is assigned as the label. Otherwise, a formal model of tonal harmony, a tree-based rule, depicted in Figure 18, is consulted to select the most harmonically sensible sequence among a list of chord candidates.

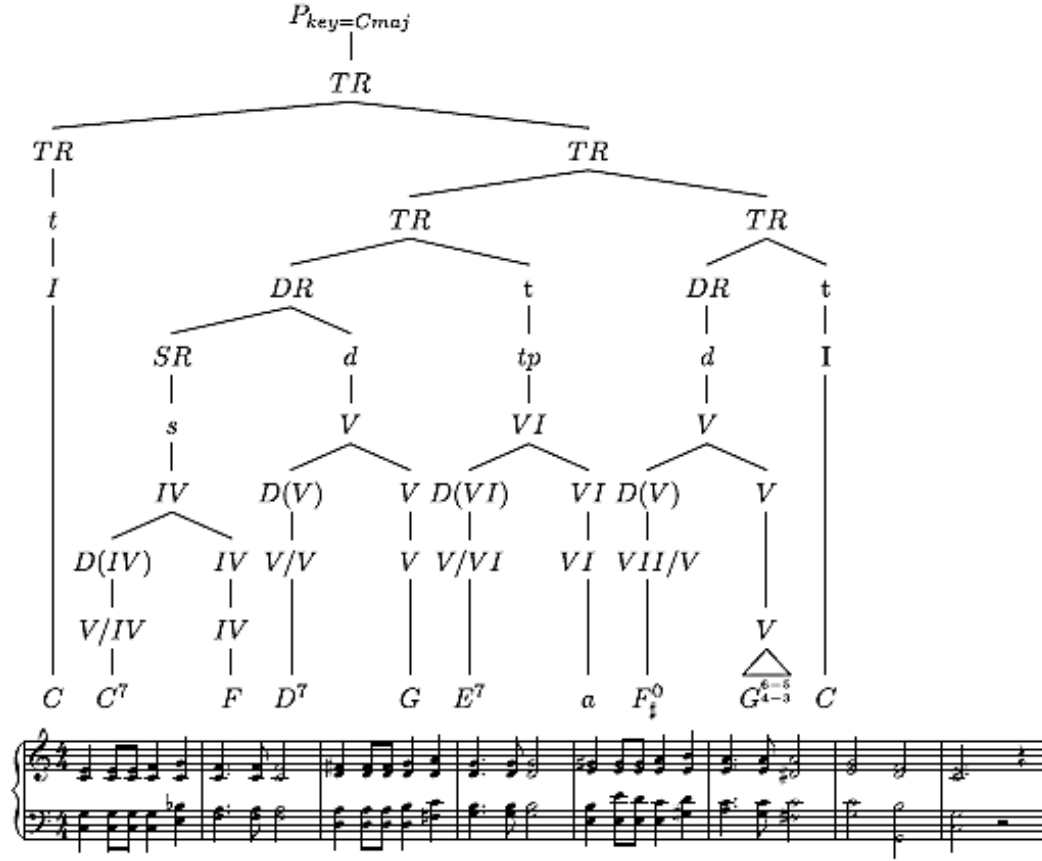


Figure 18: Rule-based tonal harmony by de Haas (de Haas, 2012)

Hu and Saul (2009; Hu, 2012) employed unsupervised learning technique using a Latent Dirichlet Allocation (LDA) probabilistic model to determine keys and chords for symbolic and real audio music. In their application of LDA, musical notes (u) play the role of words and a music song (s) is part of M songs in a corpus $S = \{s_1, s_2, \dots, s_M\}$. Each music document consists of a sequence of N segments (denoted \mathbf{u}) so that $s = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$. Musical keys (z) play the role of hidden topics so that $\mathbf{z} = \{z_1, z_2, \dots, z_N\}$. The graphical model is depicted in Figure 19 where α , β , and θ are parameters that govern the generative process. In their experiment, however, they did not use audio recordings

from the CD albums but used only MIDI-synthesized audio files which can potentially be very different from the original recordings.

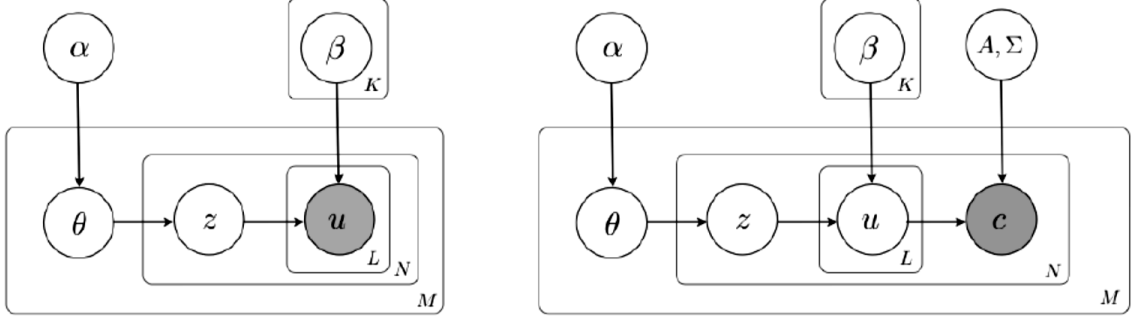


Figure 19: Latent Dirichlet allocation for key and chord recognition (Hu, 2012).
Left model: symbolic music; right model: real audio music

Lin et al. (2011) proposed a system, trained and tested with MIDI symbolic music, using a three-layer feed-forward Artificial Neural Network (ANN) trained by Particle Swarm Optimization (PSO) and Backpropagation (BP). In this work, only successions of single tones in melody are considered for both training and testing. Furthermore, a metrical structure of 4-4 (quarter-note as a beat and 4 beats per measure) is assumed as well as the six types of cadence numbers that are used to cover conclusive and inconclusive phrases in the melody. Only three major chords – C, F, and G – are included in the training and testing datasets.

In the supervised machine learning paradigm of tonality and harmony estimation, most methods summarized in Table 5 use a generative process with the assumption that latent, or hidden, sources are responsible for generating pitches, pitch clusters (chords), or

tonal centers as described in Bharucha's model. Weller et al. (2009), on the other hand, employed a discriminative Support Vector Machine (SVM) which avoided density modeling in a generative setting commonly found in HMMs. Specifically, the existing 2008 LabROSA Supervised Chord Recognition System is modified by replacing the HMM with a large margin structured prediction approach (SVMstruct) using an enlarged feature space which improved the performance significantly.

MIDI synthesized audio have the potential to be used as training data for supervised learning methods in key and chord recognition as proposed by Lee and Slaney (2008). The lack of manually expert-transcribed pop music as training data for the two tasks is widely documented for the past decade which we review in Section 4.4. In their approach, they use the Melisma Music Analyzer developed by Sleator and Temperley (2001) to obtain chord labels along with other information such as meter and key from the MIDI files. With chord labels and their timing boundaries, these MIDI files are converted to the WAV format using a variety of computer instruments as training data. A 24-state and 36-state HMMs, are constructed for the Beatles and classical music, respectively; each state represents a chord using a single multivariate Gaussian component. Furthermore, Lee and Slaney developed 24-state key-dependent HMMs so that a specific HMM is chosen for chord recognition based on the most probable global key identified. Using the Viterbi decoder, the chord sequence is obtained from the optimal state path of the corresponding key model. Their model is described in Figure 20.

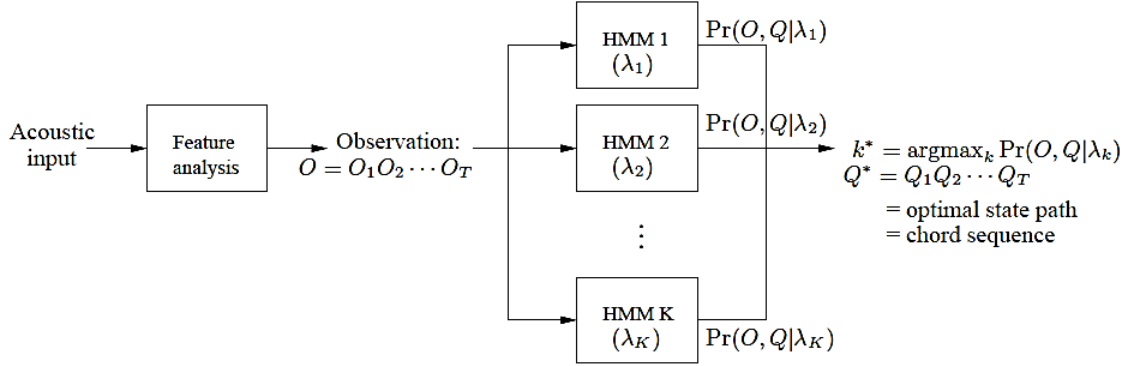


Figure 20: Chord recognition model developed by Lee and Slaney (2008)

Rocher et al (2010) proposed an unsupervised concurrent estimation of chords and keys from audio which involve four steps. First, chroma vectors are extracted from audio signals. Second, a set of key-chord candidate pairs are established for each frame. Third, a weighted acyclic graph is constructed using candidate pairs as vertices and Lerdahl's distance (Lerdahl, 2001) as edges. Fourth, the best key-chord candidate sequence is computed using dynamic programming technique that minimizes the total cost along the edges of the graph. Pauwels et al. (2011) also developed a very similar system which largely follows the four steps described earlier.

Another notable unsupervised approach is by Odure et al. (2011) which only takes chroma features and a user-defined dictionary of chord templates to estimate chords of a music piece in a probabilistic framework without using other music information such as key, rhythm, or chord transition. Candidate chords for each frame are treated as

probabilistic events and the fitness of each chord candidate is measured by the Kullback-Leibler divergence between the chroma feature and candidate chord templates.

2.4 Mixture Models

In our work, we use an infinite Gaussian mixture model (IGMM) (Rasmussen, 2000; Wood & Black, 2008), a specific instantiation of Dirichlet Process Mixture model (DPMM), as a probabilistic framework to model the uncertainties for key and chord analysis. In this section, we review the fundamentals and specifications of a generic DPMM to facilitate the discussion of IGMM in Section 3.2.

To use a traditional mixture model, as a prerequisite, the number of mixture component needs to be specified prior to the modeling effort; however, such information is usually not available. Therefore, the use of a finite mixture model is not suitable in our application. A DPMM was first proposed by Ferguson (1973) and Antoniak (1974) which eliminated this need by treating the number of mixture component as part of the unknown parameters to be estimated.

Figure 21 depicts the simplest form of a DPMM which we call a basic DPMM to differentiate it from other forms of DPMM.

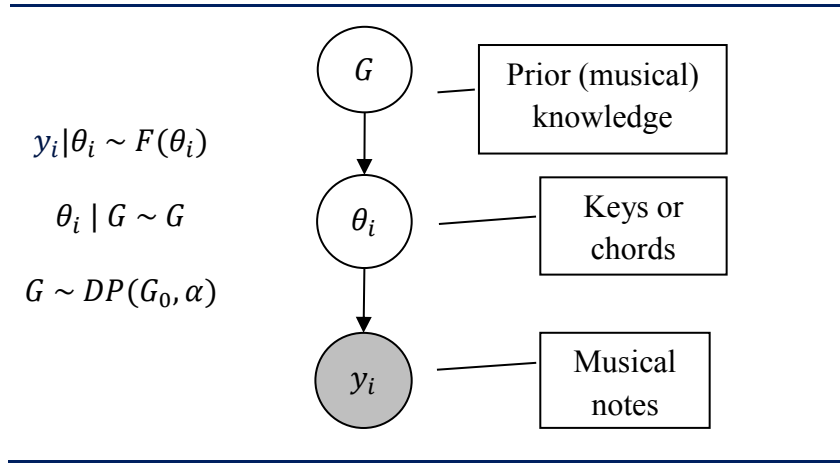


Figure 21: A basic Dirichlet Process Mixture Model

Parameters in Figure 21 are defined below:

- $Y = \{y_1, y_2, \dots, y_n\}$ denotes the n observed data points.
- G is drawn from a Dirichlet Process (DP) with a base (arbitrary) distribution G_0 and a concentration parameter α . We denote $G \sim DP(G_0, \alpha)$. θ_i 's are random samples generated from G . We denote $\theta_i | G \sim G$ and $\Theta = \{\theta_1, \theta_2, \dots, \theta_j\}$. θ_i may repeat due to discreteness. Distinct values of θ_i 's are represented by $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_k\}$.
- y_i is generated by a mixture of distribution $F(\theta)$. We denote $y_i | \theta_i \sim F(\theta_i)$. Each F_i has a density $f_i(\cdot)$.
- Define $\Theta^{(i)} = \{\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_k\}$.

We will elaborate the use of parameter G in the context of a Dirichlet distribution and a Dirichlet Process. A Dirichlet distribution, often denoted $Dir(\alpha)$, is the multivariate generalization of the beta distribution. A beta distribution can be used to model events bounded by a pair of minimum and maximum values while a Dirichlet distribution typically models a set of categorical-valued observations where the size of the vector α determines the number of categories and the values of α represent the concentration of each category. A Dirichlet Process, denoted as $DP(G_0, \alpha)$, is a stochastic process which generates an infinite stream of parameter values drawn from the base distribution G_0 and the concentration vector parameter α ; i.e., a draw from a DP produces a random distribution. Based on the above specifications, we can immediately write down the posterior distribution in Equation 9 which is the product of likelihood and prior:

Equation 9: Posterior distribution of Gaussian parameter

$$p(\varphi_j | Y) \propto [\prod f_i(y_i | \varphi_i)] p(G | G_0, \alpha) \text{ for } j = 1 \dots k$$

Integrating out G , from Blackwell and MacQueen (1973), we have the following distribution of θ_i given $\theta^{(i)}$:

Equation 10: Sampling function 1

$$\theta_i | \theta^{(i)} \sim \frac{\alpha G_0}{(\alpha + n - 1)} + \frac{1}{(\alpha + n - 1)} \sum_{j \neq i} \delta(\theta_j)$$

Equation 11: Sampling function 2

$$\theta_i | \Theta^{(i)} \sim \frac{\alpha G_0}{(\alpha + n - 1)} + \frac{1}{(\alpha + n - 1)} \sum_{j=1 \sim k} n_j \delta(\varphi_j)$$

where $\delta(x)$ is a Dirac delta function. Equation 10 and Equation 11 state the most important results of a DPMM which characterizes the fact that given all previously obtained θ 's, the next θ will be based on the following:

- A new θ_i , i.e., the value of θ_i that was not seen before, will be generated with a probability proportional to α .
- A repeated θ_i , i.e., the value of θ_i seen before, will be generated with a probability proportional to how many times it was generated before in relation to other θ 's.

Equation 10 and Equation 11 give the theoretical footing for the sampling process to generate localized candidate keys and chords in a music piece. This sampling process has the same form as that of a Chinese Restaurant Process (CRP) which enables us to generate infinite number of samples. Imagine there is a Chinese restaurant with an infinite number of tables and each table also has the potential to seat unlimited number of customers. The owner of the restaurant uses Equation 10 and Equation 11 as the seating rule to seat his customers as below:

- The first customer may pick any table of his liking.
- The following customer may pick an empty table with the probability proportion

to α or one of the occupied tables with a probability proportion to the number of customers already occupied at that table.

- To make sure empty tables are picked so that tables with large number of customers do not get over crowded, the owner uses another sampling process to determine α probabilistically.

The sampling process described in Equation 10 and Equation 11 are intuitively simple but inefficient as suggested by Neal (2000); therefore a different form of the Dirichlet process mixture model is in order which is specified in Figure 22.

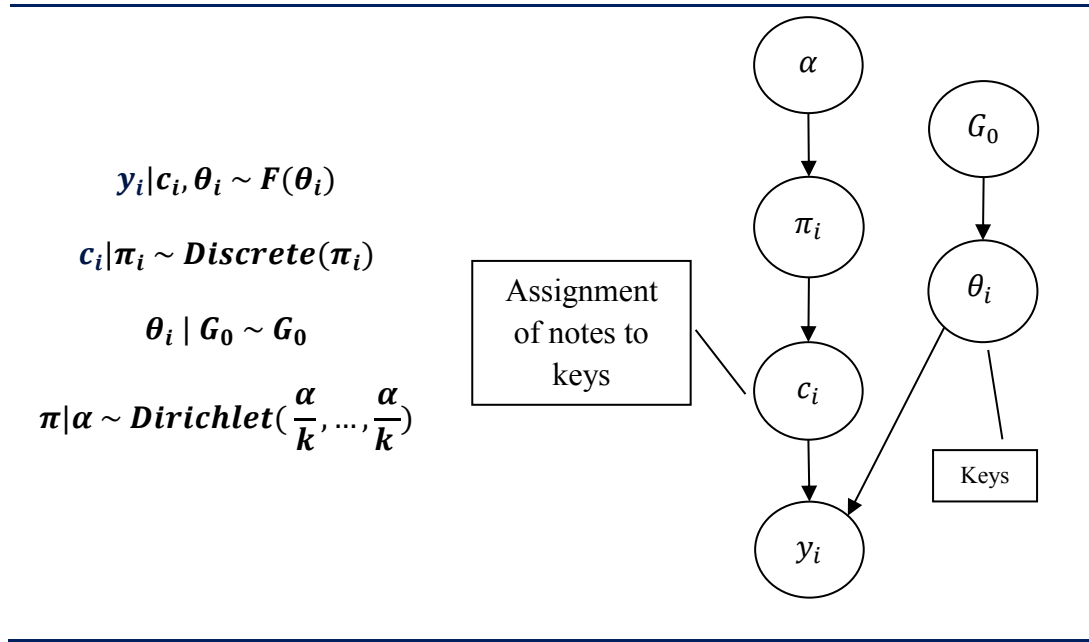


Figure 22: A standard DPMM for key and chord modeling

The first question that comes to mind regarding the standard DPMM is the disappearance of the DP from the basic DPMM. Instead, the components of the DP are decoupled into two places – the base measure G is being used solely to generate Θ while the concentration parameter (α) is used in the Dirichlet distribution as a prior for a discrete distribution of the mixture proportions (π). Notice the difference between a Dirichlet distribution and a DP is that the Dirichlet distribution has a fixed dimension while the DP is infinite in terms of its measure space. Therefore, it would be apparent that, in the current model, when we take k to infinity, we would immediately have a DP. Now we formally define the new parameters used in the standard DPMM:

- Parameter α is the prior for a discrete distribution for mixture proportions π_i where $i = 1, \dots, k$.
- The class indicator $C = \{c_1, c_2, \dots, c_n\}$ establishes a mapping between Y and Φ . Therefore, $c_i = j$ if $\theta_i = \varphi_j$.
- Define $c^{(i)} = \{c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n\}$.

C and Θ are the two model parameters that we need to use as the vehicle to recognize keys and chords. From Equation 10 and Equation 11, we immediately deduce that C has the same prior predictive distribution as that of Θ since $(\theta_i | \Theta^{(i)}, Y, C, k)$ is proportional only to either the counts of observations generated by Φ for a repeated value which was seen before (an occupied table) or α for a new value (an empty table).

Therefore, the predictive (or prior) distribution of c_i given all other variables $p(c_i = j | \Theta^{(l)}, Y, c^l, k)$ can be expressed below:

Equation 12: Sampling function for an existing index variable

$$p(c_i = \text{existing_}j | \pi, \alpha) = p(c_i = \text{existing_}j | \pi) \propto n_j$$

Equation 13: Sampling function for a new index variable

$$p(c_i = \text{new} | \pi, \alpha) = p(c_i = \text{new} | \alpha) \propto \alpha$$

From Figure 22 and Equation 13, we see that hyperparameter α serves as a prior to the mixture proportions as well as a probabilistic event to introduce a new θ into the mixture of local keys. To sample hyperparameter α from the generative process depicted in Figure 22, we follow the sampling process proposed by (West, et al., 1994) as described in Equation 14. The idea is to draw a new value for α at the end of each iteration (after processing all n data points) based on the most recent values of α and k (number of Gaussian components) using Gamma(1, 1) as the prior for α .

Equation 14: Sampling function for alpha

$$p(\alpha | k, \pi, Y) = p(\alpha | k) \propto p(\alpha) p(k | \alpha)$$

Chapter 3 Methodology

Two principles guide our development of the methodology. The first is Einstein’s “Make everything as simple as possible, but not simpler.” The second principle attributes to Butler Lampson’s quote and David Wheeler’s corollary “All problems in computer science can be solved by another level of indirection, except for the problem of too many layers of indirection.”

Due to the scarcity of manually transcribed training data, we choose to directly estimate local keys and chords from the target music data without using any training data; therefore, our overarching approach is unsupervised machine learning in contrast to the more popular supervised learning methods we reviewed in Chapter 2. In Section 3.1, we provide an overview of the methodology and how each component contributes to the extraction and recognition of keys and chords for music in symbolic and audio formats. Since the infinite Gaussian mixture model (IGMM) plays an important role in extracting a bag of local keys (BOK), a common thread in our approach for both symbolic and audio formats, in Section 3.2, we review the general specification of an IGMM and how it is constructed as a generative process to extract a BOK. In Sections 3.3 and 3.4, we provide treatment that is specific to the symbolic and audio domains, respectively. In the last section, we discuss performance metric that we employ in evaluating our proposed method.

3.1 Overview of the Methodology

The core components of the methodology are depicted in Figure 23 in which the horizontal dimension covers modules used in the symbolic and audio domains while the vertical dimension depicts the processing flow from signal processing, key and chord recognition, and validation. Since the data format of symbolic and audio signals are drastically different, as described in Sections 1.2 and 2.2, the signal processing mechanisms used in extracting keys and chords for each data format is expected to be different. For symbolic music (MIDI), features such as musical notes and their duration of play can be easily extracted. However, for real audio signals, the task of extracting clean pitch information remains a difficult research problem since the 1970s (Lerch, 2012, p. 94). Therefore, in the signal processing step, we propose to employ an undecimated wavelet transform on the raw audio signals to produce cleaner and smoother signals by reducing transient noise and filtering out higher harmonics.

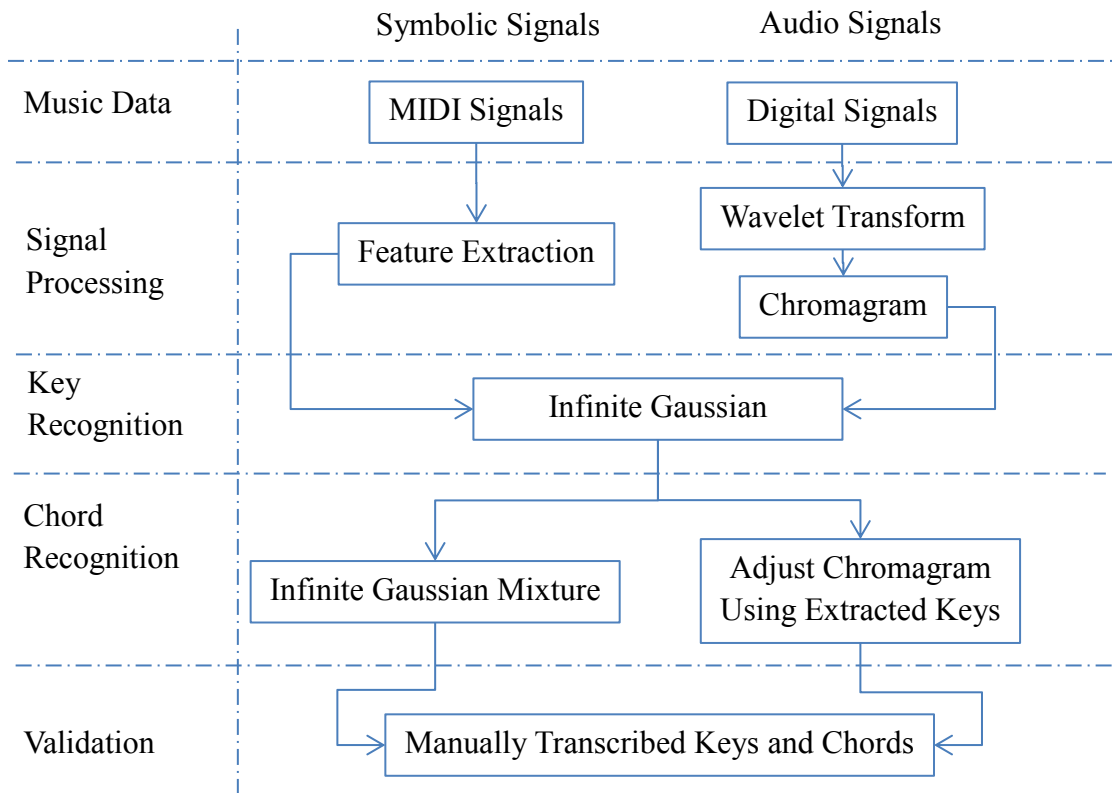


Figure 23: Methodology overview

The main approach that we adopt in key and chord recognition is to extract a bag of local keys first and then use the extracted key information to improve the recognition of chords. A bag of local keys is extracted from an Infinite Gaussian Mixture model (IGMM) without training data. Since an unsupervised machine learning approach typically is employed to perform clustering without training data, our method uses IGMM to find “clusters” as tonal centers, directly from the musical piece. The IGMM is a generative process which we depict in Figure 24.

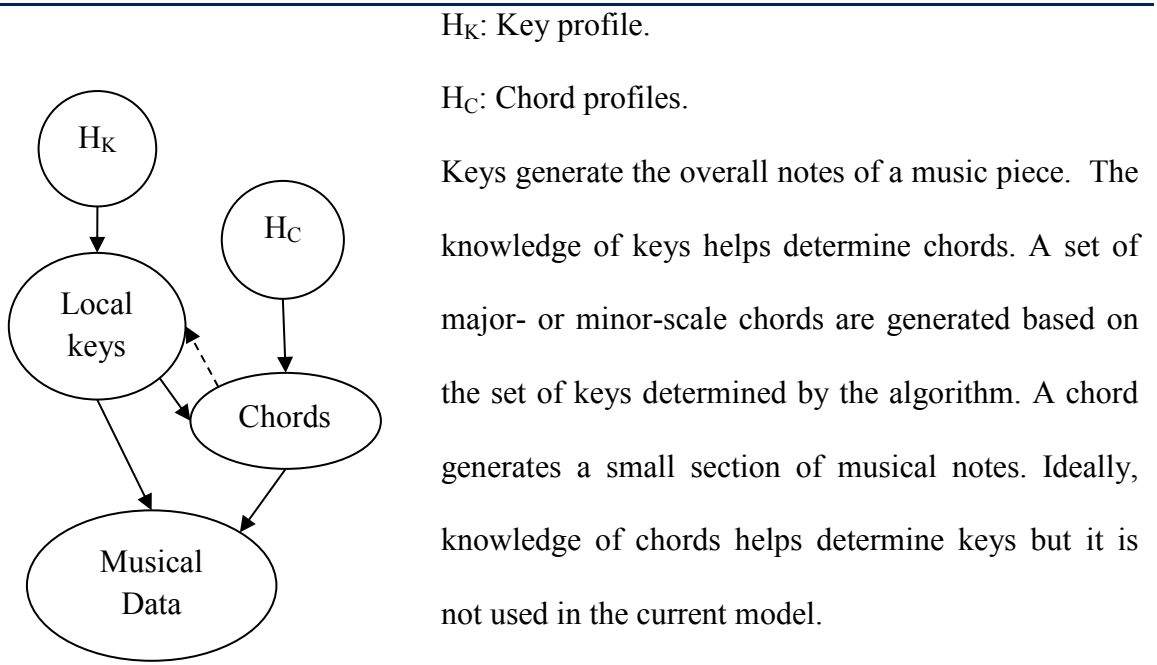


Figure 24: A conceptual generative process for keys and chords.

There are three main distinguishing ideas in our methodology. First, we use one generative model to determine what keys (or chords, for symbolic music) generated the overall and localized set of musical notes. Second, since the judgment of keys and chords are subjective as described in the Introduction section, our technique models extracted keys and chords as probability distributions. Third, our technique directly estimates keys and chords without using any training data. We discuss the detail specifications of the generative model – IGMM, a specific instantiation of a Dirichlet Process Mixture Model – in the next section.

3.2 Infinite Gaussian Mixture Model

We use the Infinite Gaussian Mixture Model (IGMM) to model the generative process of musical data as well as the musical knowledge related to keys and chords. When presented with a music piece, without any prior knowledge of the piece, we do not know if there are any key modulations or the number of chord types involved. Without such precise knowledge, it is not ideal to use a mixture model pre-specified with a fixed number of components such as GMM and Bayesian GMM. Following Wood’s depiction (Wood & Black, 2008), Figure 25 provides a hierarchical specification – a plate notation – of traditional GMM, Bayesian GMM, and IGMM. In the plate notation, we note the difference between a traditional and Bayesian mixture is the addition of prior knowledge (G_0, α) in the Bayesian mixture while the infinite Bayesian mixture employs potentially infinite number of model parameters (θ, π) in which the exact number of components are fully determined by the observed data.

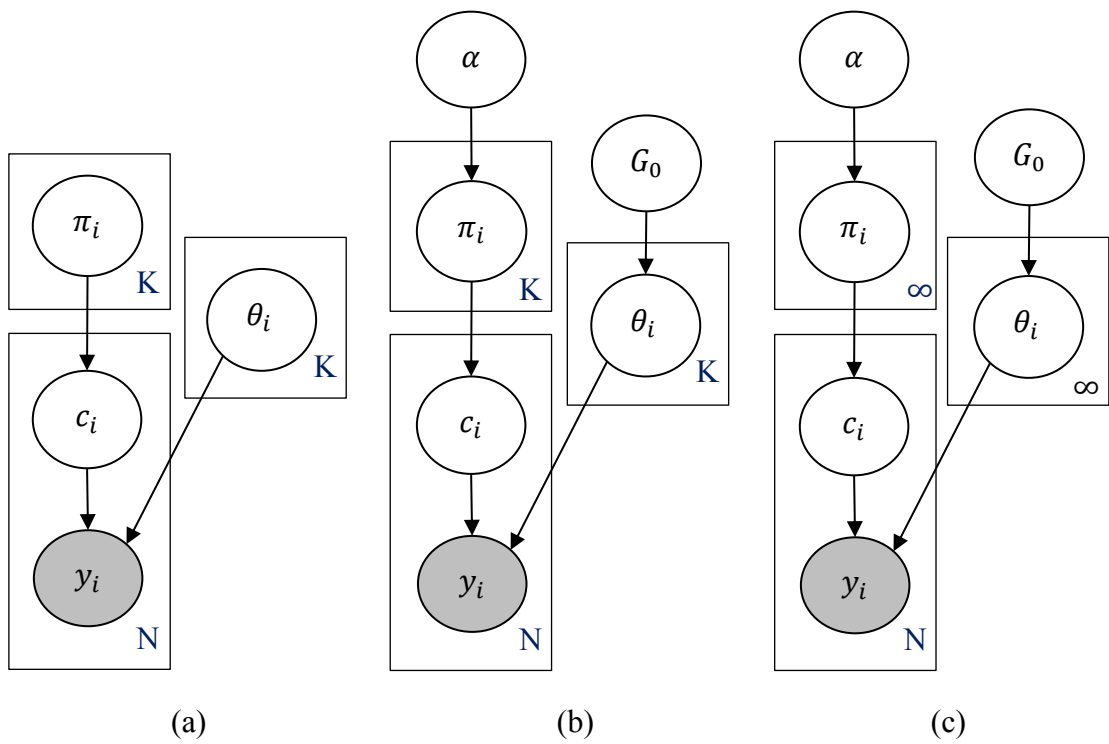


Figure 25: Types of mixture models (Wood & Black, 2008). (a) Traditional mixture, (b) Bayesian mixture, and (c) Infinite Bayesian mixture. The numbers at the bottom right corner represent the number of repetitions of the sub-graph in the plate.

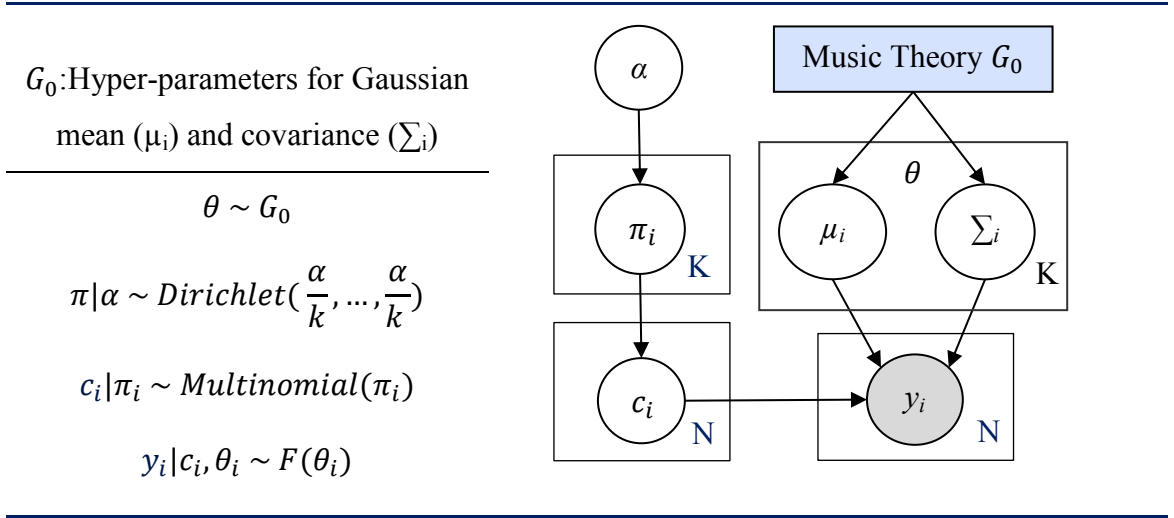


Figure 26: Specification of Infinite Gaussian Mixture Model

Since IGMM is a specific instantiation of DPMM, its model parameters are similar to DPMM. For easier reference and discussion in the context of using IGMM for extracting a bag of local keys, we repeat some of the definitions described in Section 2.4 and provide a complete definition of the IGMM parameter below:

- $Y = \{y_1, y_2, \dots, y_n\}$ denotes the n groups of musical data.
- Music theory G_0 governs the generation process of $\theta_k = \{\mu_k, \Sigma_k\}$.
- θ_i 's (keys or chords) are random samples generated from G . We denote $\theta_i | G \sim G$ and $\Theta = \{\theta_1, \theta_2, \dots, \theta_j\}$. θ_i may repeat due to discreteness. Distinct values of θ_i 's are represented by $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_k\}$.
- y_i is generated by a mixture of distribution $F(\theta)$. We denote $y_i | c_i, \theta_i \sim F(\theta_i)$. Each F_i has a density $f_i(\cdot)$.

- Define $\Theta^{(i)} = \{\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_k\}$.
- α is the prior for a discrete distribution for mixture proportions π_i where $i = 1 \dots k$.
- The class indicator $C = \{c_1, c_2, \dots, c_n\}$ establishes a mapping between the observed music Y and the generating keys or chords Φ . Therefore, $c_i = j$ if $\theta_i = \varphi_j$.
- Define $c^{(i)} = \{c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_k\}$.

Musical notes Y are generated by a mixture of multivariate Gaussian components with Gaussian parameters Θ . Y is represented as an $n \times 12$ matrix. The prior knowledge over class (keys or chords) assignments specify how likely a set of musical notes would belong to (or be generated by) a key or chord. The mixing proportions (π) are modeled as a Dirichlet distribution which serves as a prior for multinomial component indicators (c_i). Since Dirichlet distribution is a conjugate prior to the multinomial distribution, the posterior of c_i is also Dirichlet. They are represented as the following:

Equation 15: Distribution for the proportional variable

$$\pi | \alpha \sim \text{Dirichlet}\left(\frac{\alpha}{k}, \dots, \frac{\alpha}{k}\right)$$

Equation 16: Distribution for the indexing variable

$$c_i | \pi_i \sim \text{Multinomial}(\pi_i)$$

We note that the structure of IGMM is identical to a standard DPMM and therefore the sampling process of a new θ_i (keys or chords) in an IGMM can be expressed in Equation 12 and Equation 13 as described in Section 2.4. Furthermore, in the context of key finding using a Chinese Restaurant Process with a mixture model, we can think of each table as a key or chord and each customer as a group of simultaneously played notes. As each musical note (or a group of notes) arrives, we probabilistically assign it to a key that most likely generated it, given the knowledge that we have obtained up to the arrival of that data point; we repeat this sampling process until the assignment of all musical notes converges. The same can be imagined for the context of chord recognition. For both tasks, we generate possible keys and chords based on a CRP to best fit the entire (or segment of) music piece, without setting the number of such tables a priori. If we repeat such sampling process $N_{\text{warm-up}} + N$ iterations and discard the samples generated by the first $N_{\text{warm-up}}$ iterations, we have collected N samples or keys or chords. Such samples represent our belief of what keys or chords generated each localized segment (of various lengths) of the music piece.

Given the observed music $Y = \{y_1, y_2, \dots, y_n\}$, the joint posterior distribution of the model parameters is described in Equation 17. Since the indicator variable \mathbf{c} associates each chroma vector to key θ , together they completely determine what local key generated each chroma vector. Therefore, as described in Section 2.4, our goal is to use an iterative sampling process to obtain c (Equation 12 and Equation 13) and θ (Equation 10 and Equation 11).

Equation 17: IGMM joint distribution

$$p(c, \theta, \pi, \alpha | Y) \propto p(Y | c, \theta) p(\theta | G) \prod_{i=1}^n p(c_i | \pi) p(\pi | \alpha) p(\alpha)$$

Music theory (G_0) consists of a set of hyperparameters to form distributions that govern the generation of candidate keys and chords for the given music piece $Y = \{y_1, y_2, \dots, y_n\}$. Specifically for IGMM, the relationship between G_0 and $\theta_k = \{\mu_k, \Sigma_k\}$ can be described below.

Equation 18: Prior for Gaussian covariance

$$\Sigma_k \sim \text{Inverse} - \text{Wishart}(\Lambda_0^{-1}, d_0)$$

Equation 19: Prior for Gaussian mean

$$\mu_k \sim \text{Gaussian}(\mu_0, \Sigma_k)$$

where the Inverse-Wishart distribution is the conjugate prior for the covariance matrix of the multivariate Gaussian.

In Figure 26, θ_i is a Gaussian component with mean (μ_i) and covariance (Σ_i). $C = \{c_1, c_2, \dots, c_n\}$ is an indicator variable establishing a mapping between each chroma vector in Y and Θ . Hyperparameter α is the prior for a discrete distribution for mixture proportions (π_i) where $i = 1 \dots k$. A GMM would have a set value of k , but in the case

of an IGMM, k is completely determined by the generative process which allows it to go into infinity. The mixing proportions (π_i) are modeled as a Dirichlet distribution which serves as a conjugate prior for multinomial component indicators (C). A similar infinite mixture called Infinite Latent Harmonic Allocation has been recently proposed by (Yoshii & Goto, 2012) as a multipitch analyzer which estimates multiple fundamental frequencies (F0) from audio signals.

Table 7: Gaussian coding examples for IGMM

Examples	[C, C#/Db, ..., A#/Bb, B]
C Major Key Profile	[5 0 3 0 3 3 0 3 0 3 0 3]
C (harmonic) Minor Key Profile	[5 0 3 3 0 3 0 3 3 0 0 3]
C Major Key Covariance Matrix	A 12x12 matrix where 1 is assigned to notes with > 0 values in the C Major key profile
C Major Chord Profile	[5 0 3 0 3 0 0 3 0 0 0 0]
C Major Key Covariance Matrix	A 12x12 identity matrix

Table 7 describes how we encode musical notes (y_i), means of Gaussian key and chords (μ_i), and covariances of Gaussian keys and chords (Σ_i). For y_i and μ_i , we follow closely with the encoding profile proposed by Lerdahl (2001). We implement Σ_i as an identity matrix. These encodings are the constraints used to guide the unsupervised

learning of IGMM to efficiently recognize latent keys and chords which generated the observed musical notes through the CRP.

3.3 Symbolic Domain

Symbolic music such as MIDI (Musical Instrument Digital Interface) contains all the data necessary for computers to play the music prescribed in the MIDI file. Extracting useful features from MIDI for the two tasks (key and chord recognition) are straightforward as described in the first subsection. The second subsection discusses the details of how to use an IGMM to model a music piece which leads to effective recognition of keys and chords.

3.3.1 Feature Extraction

A MIDI file stores musical performance information to be played by a MIDI device or a computer that connects to a MIDI interface. A MIDI file does not contain any recording of music performed by musicians but instead instructions to a MIDI-equipped device on how to play it. A sound synthesizer is one example of such device that is capable of imitating timbres of different musical instruments. Similar to a composer of classical music putting musical notes on a score for different instruments of an orchestra, a MIDI composer uses a host of software and hardware to produce music that closely mimics the performance of an orchestra in a concert hall. Similar to the staff notation of music score

consumed by musicians, the MIDI specification defines the format of MIDI music, stored in a MIDI file, to be read and played by a MIDI device.

A MIDI file contains a sequence of commands, also known as events, regarding the timbre as well as note pitches and their starting and ending times. A MIDI device turns these sequences into signals consumed by the sound cards to produce the intended music. We use Toivainen and Eerola's MIDI Toolbox (2004) to read a MIDI file into a matrix where features and sequence of events are represented by columns and rows, respectively. The toolbox extracts seven features for each event: onset (in beats), duration (in beats), MIDI channel, MIDI pitch, velocity, onset (in seconds), and duration (in seconds). The onset and duration indicates the starting time or beat of the MIDI pitch specified in the event and the length of such event. A MIDI channel can be thought of as the timbre generated by different instruments while the velocity indicates how forceful a note should be played. MIDI channels can be used to filter out sounds produced by percussion instruments since such sounds do not directly contribute to the recognition of keys and chords. Though the information of how fast or forceful a note is played in a piece can be useful in aiding the two tasks that we have at hand, we discard this information to simplify the modeling effort. Figure 27 depicts these seven features in the MIDI representation for the Beatles' song "Let it be."

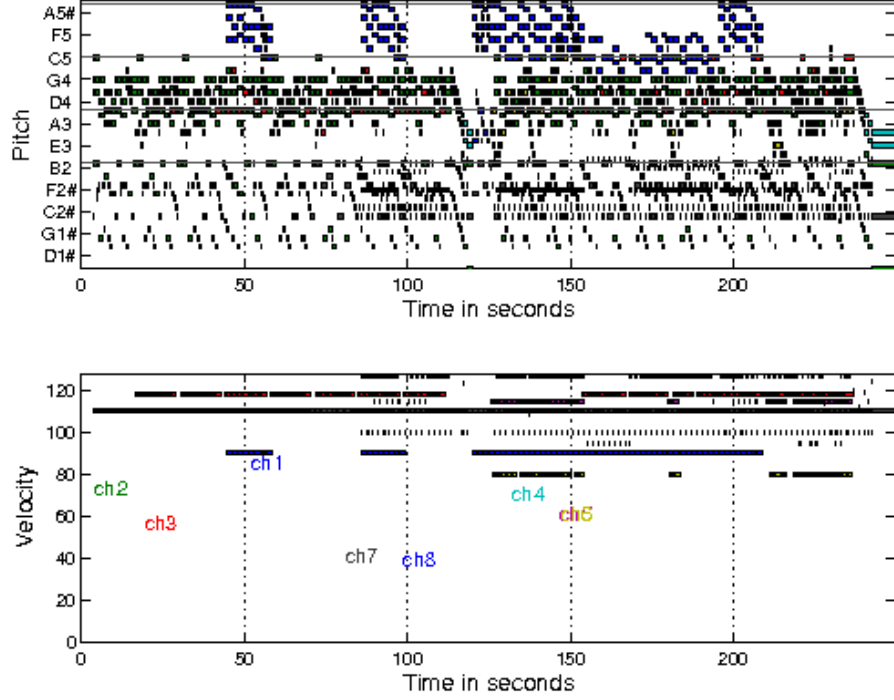


Figure 27: MIDI representation of "Let It Be"

Four features – onset time, duration, MIDI channel, and MIDI pitch – are first extracted to obtain groups of simultaneously-played musical notes. After the extraction, we convert the extracted MIDI pitches to pitch classes as a sequence of data points. We denote them as $Y = \{y_1, y_2, \dots, y_n\}$ where y_i represents the i th group of pitch classes that are played together. As described earlier, percussion sound is treated as noise and filtered out through the proper MIDI channel. Note that, however, unlike most of the profile-based key-finding algorithms, we do not use the time duration of each data point to recognize keys and chords. In other words, we hypothesize that the duration of each set

of notes played in the music piece has minimal impact on the key and chord finding activities.

3.3.2 Keys and Chords Recognition

The first data point is denoted as y_1 and the last group is denoted as y_n . We feed $Y = \{y_1, y_2, \dots, y_n\}$ into the IGMM to iteratively generate key and chord samples that most likely produced Y . We arbitrarily generate the first key sample, Key_{sample}^1 , which is in turn used to help generate the first sample of chords, $Chord_{sample}^1$. After some burn-in iterations, these samples start to converge to the estimated keys and chords. Note that a sample generated from an iteration, say Key_{sample}^i or $Chord_{sample}^{i+1}$, contains all possible keys (due to modulations) or chords used in the entire music piece. In other words, a key sample is a time series of keys and a chord sample is a time series of chords for the entire target music piece. We iterate $2s$ times until we have generated s samples of keys and chords. In our implementation, we model 24 types of keys (12 tonic x 2 modes) and 13 types of chords (power, major, minor, diminished, augmented, suspended, 7th, major 7th, minor 7th, diminished 7th, major 6th, first inversions of major and minor triads) for each key. Table 8 depicts the algorithm for key and chord recognition using IGMM.

Table 8: Sampling algorithm using IGMM for symbolic key and chord recognition

Preprocess the MIDI file to extract a four-dimensioned feature {onset time, duration, MIDI channel, and MIDI pitch} and store them as input data $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$

Initialize G; Initialize c_1 and θ_1 to random values.
For $i = 1$: 2s samples do
For $j = 1$: n sets of musical notes do
Sample a new c based on Equation 12 and Equation 13
If a new θ_i is required
Sample a new θ based on Equation 18 and Equation 19
Update α based on θ_i 's distribution from iteration $i-1$
End
End
Regroup \mathbf{Y} based on all sampled Θ ; For each cluster generated by θ_i , find the closest key/chord profile as the output label
End

Given Y , we use a generative process to determine what local keys (latent variable Θ) generated Y without any training data. Our emphasis is on finding the most likely local keys that are present in the target music piece but ignore their sequence and precise modulation points. Each θ_i in Θ is modeled as a Gaussian component, specified by its mean and covariance. To bypass the requirement of specifying the number of local keys in a Gaussian mixture, we use an infinite Gaussian mixture model (IGMM) depicted in Figure 26.

3.4 Audio Domain

In the acoustic audio domain, we perform key and chord recognition on music recordings, such as albums on compact disks (CDs), of sound waves produced by instruments or human singing. We extract music directly from CDs and convert it to the .WAV file format. Different from a midi file containing commands to instruct midi devices how to play the music, a wav file contains encoded acoustic sound waves to be decoded by computers when played. Due to the drastic differences between midi and wav files, we approach the two tasks in this section differently but still aim to use the same probabilistic framework as described in the previous section. Table 9 describes the four stages of our system for the audio domain which corresponds to the audio track in Figure 23.

Table 9: Four stages of extracting keys and chords from audio

Stage I	Undecimated wavelet transform on WAV audio
Stage II	Extract chroma features from wavelet approximation
Stage III	Extract a bag of local keys from chromagram using infinite Gaussian mixture
Stage IV	Adjust chromagram using KK tonal profiles based on extracted local keys to determine chords

Stage I denoises the audio file using undecimated wavelet transform. The denoised wavelet approximations are fed into Stage II - a MATLAB Chroma Toolbox - developed by Müller and Ewert (2011) to extract frame-based chromagrams. Using a simple peak-picking algorithm, the chromagram is converted into an integer-based 12-bin

representation to extract a bag of local keys using a generative process in Stage III. Using the extracted keys, we further transform the wavelet-based chromagram to recognize chords in Stage IV. The following sections describe each stage in detail.

3.4.1 Wavelet Transformation

Audio CD recordings are typically consumed by CD players, not computers. To process audio files on a computer, especially Windows platform or MATLAB program, we extract audio tracks from CDs to convert to WAV form in mono channel with uncompressed PCM (Pulse Code Modulation) at 11,025 Hertz sampling rate and 8 bits per sample. PCM is a common method of storing and transmitting uncompressed digital audio. A typical audio CD has two channels (stereo) with 16-bit PCM encoding at a 44.1k Hz sampling rate per channel. The WAV file format is commonly used for digital audio files on Microsoft Windows platform. Unlike MIDI music whose percussive sound can be easily filtered out by MIDI channels, a WAV audio is a direct representation of sounds from all participating instruments and vocals and the sound produced by percussion instruments is much harder to separate from the rest of the sound in the mix.

In this wavelet preprocessing step, we aim to reduce two types of sound – attack transients and high harmonics – that negatively impact the tasks of key and chord recognition. An attack transient is short-duration high-amplitude sound at the beginning of a sound wave which are part of an ADSR (attack, decay, sustain, release) envelope in real audio music signals (Cavaliere & Piccialli, 1997). Examples of such transient noises

are the excitement when a string is bowed or plucked, the air leakage of blowing a trumpet's mouthpiece, or when a piano key is struck. Decay transients, such as the diminishing sound of a plucked string, are very important in many instruments, particularly those that are struck or plucked. Though transients are considered "noises" for our tasks, the overall characteristics of ADSR envelope, depicted in Figure 28 are great features for instrument recognition. In rock or popular music, one of the most prominent instruments is the guitar and each tone played on such plucked instrument generates an initial transient "noise" within about the first 50 ms (Bader, 2013, p. 164) when the string is struck. Therefore, when the music is played by different instruments, the noise generated by transients can be significant, especially in popular or rock music. Similar to timbre enabling us to differentiate instruments playing two notes with the same frequency and loudness, the ADSR envelope can be used to classify different music instruments from audio signals (Li, et al., 2011). In audio recording and production application, the "attack" characteristics can be edited so that a piano can be made to sound like an organ, a French horn to sound similar to a saxophone, or an oboe to sound like a trumpet (Alten, 2011, p. 16). In other words, removing the initial transient from a musical sound significantly strips the characteristics of a musical instrument. Similar to percussion sounds, attack transients are not periodic waves; therefore they need to be minimized so that we can perform key and chord recognition more effectively.

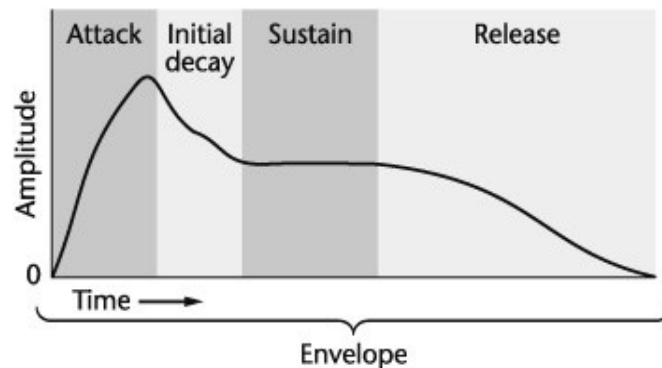


Figure 28: ADSR envelop (Alten, 2011, p. 16)

Higher harmonics are the second type of “noise” that we aim to decrease. In Section 2.1, we briefly review the unique tonal mix of fundamental and harmonic frequencies that distinguishes an instrument from others, even if the sounds have the same pitch, loudness, and duration. Since no real music contains only pure tones (sine waves) and the fundamental frequencies are the greatest contributor to extract tonality and harmony content, it is reasonable to seek ways to remove higher harmonics that negatively impact the two tasks.

Figure 29 illustrates the fundamental frequencies and their high harmonics of notes produced by a piano, violin, and flute. In the figure, though the piano and violin both play the same C4 note, we see that the violin has many more significant upper harmonics than that of the piano. In other words, if we can successfully remove all higher harmonics but keep only the fundamental frequency – in our case, C4 – the tasks of key and chord recognition would be much simpler. In the same figure, we also see many distinct higher harmonics produced by the flute as well as non-periodic white noise.

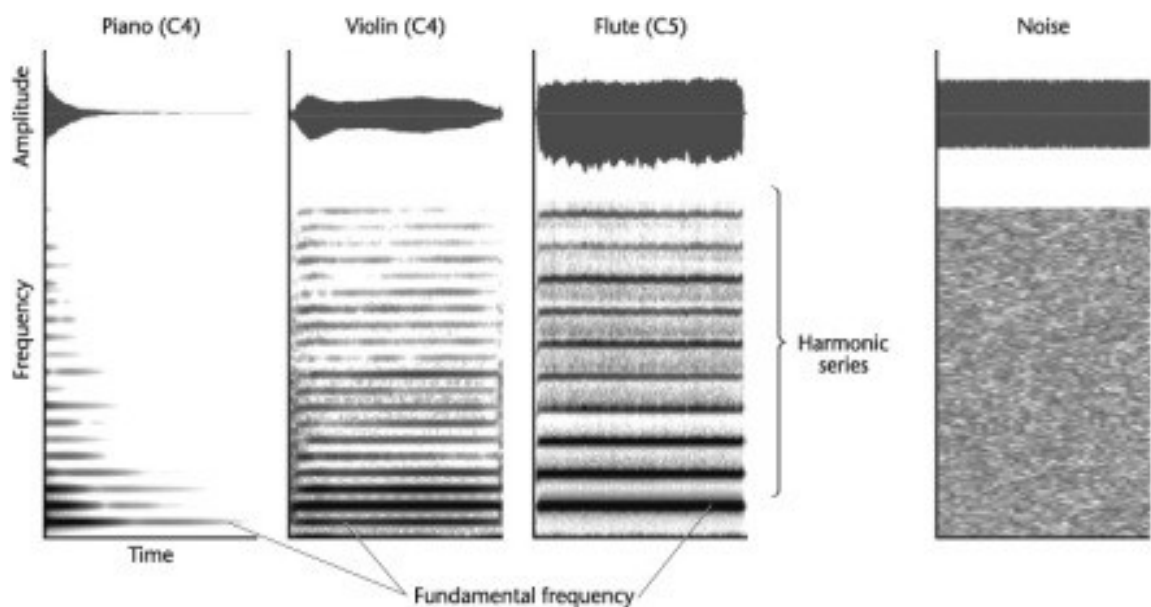


Figure 29: Fundamental frequency and harmonics of piano, violin, and flute (Alten, 2011, p. 15).

Since we aim to reduce the two types of “noise” – attack transients and high harmonics – for key and chord recognition, we have a dilemma at hand in selecting a tool that can reduce both of them – one aperiodic, the other one periodic – simultaneously. Fortunately, these two seemingly contradicting “noises” can be approached by “period regularization” using wavelet transformation. As suggested by (Cavaliere & Piccialli, 1997), one can build a two-channel system so that the output of the first channel represents period-regularized version of the input while the other channel outputs period-to-period fluctuations, transients, and noises as discussed earlier. In our case, the period-regularized output from the first channel can be used to reduce higher harmonics while the attack transient can be located in the second channel. A good candidate to perform such two-channel transformation is a wavelet transform where variable analysis window

sizes are employed in analyzing different frequency components within a signal as supposed to the fixed window size of a STFT discussed in Section 2.2. The basic idea of a wavelet transform is to apply scaling (dilation and contraction) and shift (time transition) on a base wavelet $\psi(t)$ to find similarities between the target signals and $\psi(t)$. Figure 30 depicts such transformation.

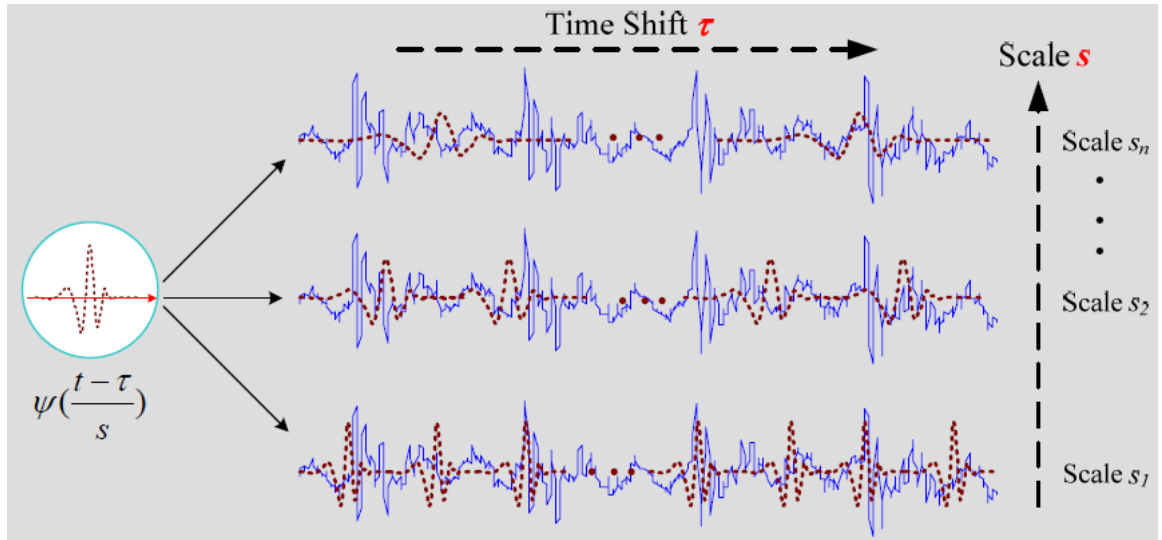


Figure 30: Wavelet transform with scaling and shift (Yan, 2007, p. 28)

Since our target music contains discrete digital signals, we will concentrate our discussion on the discrete version of the wavelet transform where the scaling and shifting can be realized using a pair of low-pass and high-pass wavelet filters. A discrete wavelet transform decomposes the input signal into two parts using a highpass and a lowpass filter – so that the lowpass filter outputs a smoother approximation of the original signals

while the high pass filter produces the residual noises. Figure 31 depicts the operations of the widely known discrete wavelet transform (DWT) and Figure 32 describes the less known undecimated discrete wavelet transform (UWT). For easier comparison, both transformation decompose the signal S at three levels; H and L represent high-pass and low-pass filters, respectively, while 2 with an arrow pointing down (in a circle) denotes “down sampling by 2 .” To reconstruct the signals from the coefficients from decomposition, we reverse transform the coefficients by upsampling $cA3$ and $cD3$, passing through L' (low-pass reconstruction filter) and H' (high-pass reconstruction filter) respectively, and combining them to form $cA2'$. In both figures, the difference between the conventional DWT and UWT is the lack of down sampling processes in the UWT and hence the term “undecimated.” Figure 33 depicts a four-level wavelet transform (only the decomposition part). Furthermore, since our signal preprocessing step involves only using the approximated signals from the wavelet transform, we will concentrate our discussion on the decomposition part of the discrete wavelet transform.

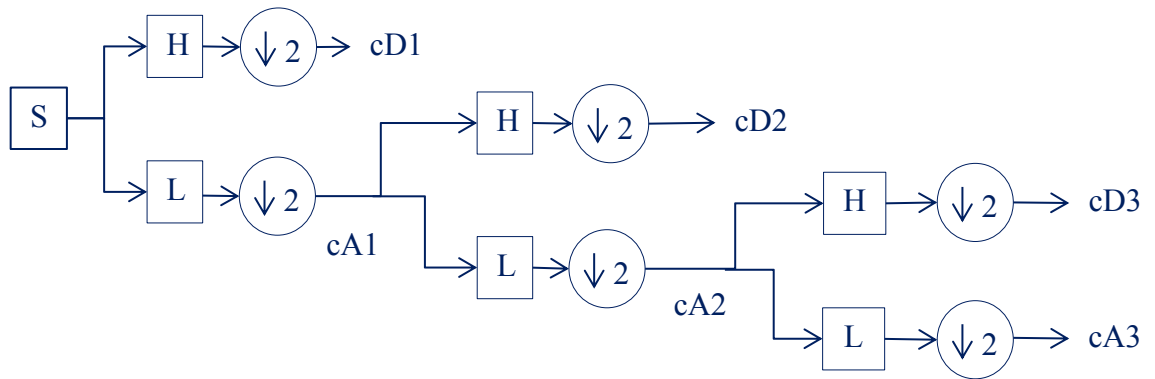


Figure 31: Discrete Wavelet Transform (DWT)

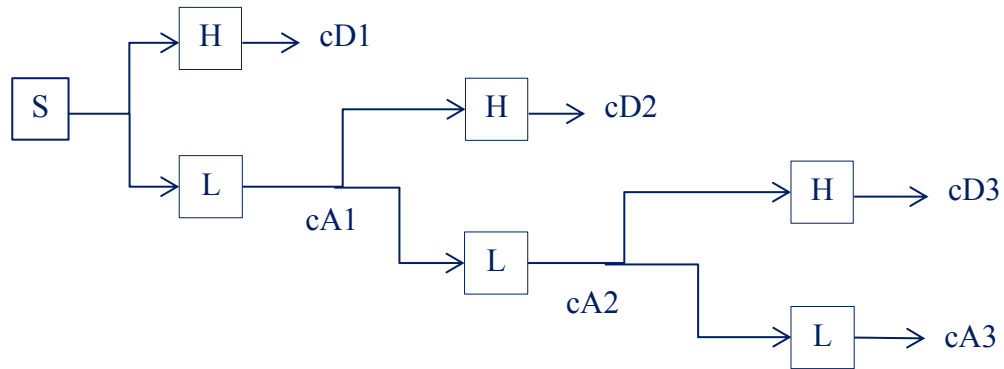


Figure 32: Undecimated Discrete Wavelet Transform (UWT)

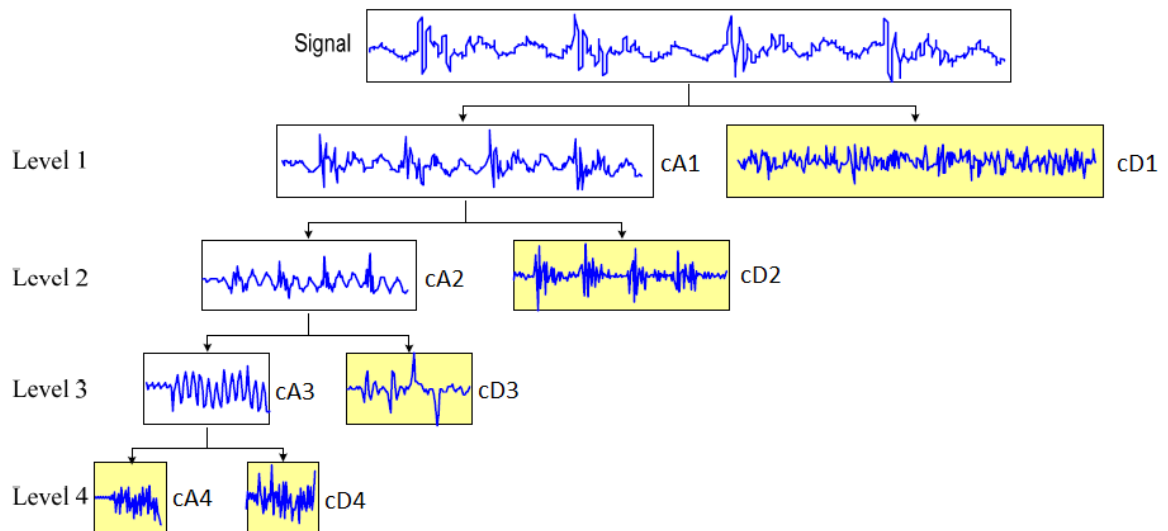


Figure 33: Four-level discrete wavelet transform (Yan, 2007, p. 36)

Regardless of how the raw audio signals – using DWT or UWT – are regularized by the wavelet transform, the first step of such transform is to select appropriate

“families” of wavelets by stretching and shifting the selected wavelet to match the target signals to discover its frequency and location in time. Therefore, the rule of thumb for selecting a proper wavelet family for transformation is to choose wavelets that match the general shape of the raw audio signals. Since the continuous versions of wavelet representation can be more easily examined in terms of their shapes than the discrete counterpart which is characterized by a high-pass wavelet filter (mother wavelet) and a low-pass scaling function (father wavelet), we inspect the shape of some well-known continuous wavelets. Figure 34 and Figure 35 illustrate order-4 and order-8 Daubachies (db) and Symlet (sym) wavelets, respectively. We see that the wave shape of the db and sym wavelets generally match that of raw audio signals within a short time span. Furthermore, as the order of the wavelet increases, the wavelet becomes smoother.

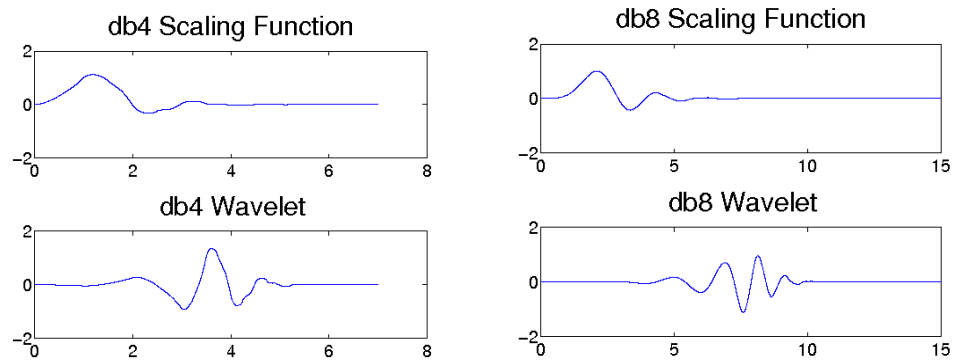


Figure 34: Daubachies scaling functions

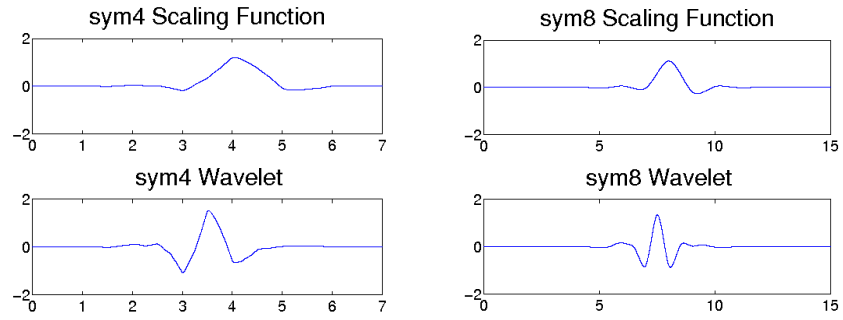


Figure 35: Symlet scaling functions

On the discrete side, a decomposing wavelet, is characterized by a pair of low-pass and high-pass filters as discussed earlier. Figure 36 depicts two pairs of decomposition filters for db8 and sym8 wavelets.

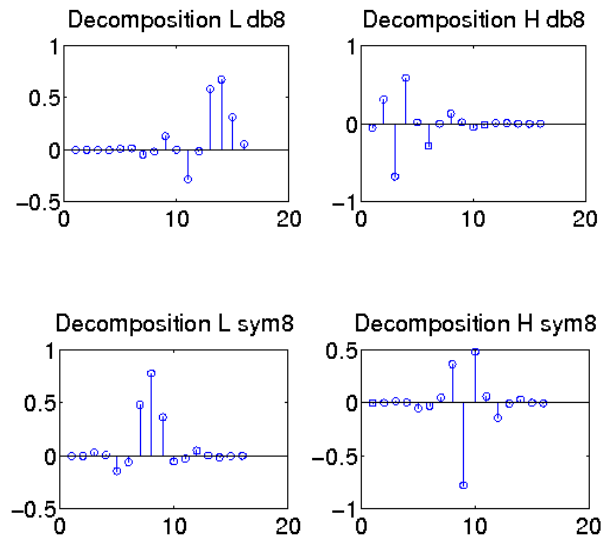


Figure 36: Decomposition wavelets. Top two: Low-pass and high-pass filters for db8; Bottom two: Low-pass and high-pass filters for sym8.

Once a wavelet and its order, such as db4, is chosen and the level of decomposition is determined, a typical denoising process using a DWT or UWT is to manipulate the decomposed signals (such as the approximation coefficients $cA1 \sim cA3$ or, especially for the purpose of denoising, detailed coefficients $cD1 \sim cD3$, as described in Figure 33) within a certain time window for certain frequency ranges before the reconstruction stage. Figure 37 illustrates the general relationship between the coefficients and frequency allocation for three levels of signal decomposition.

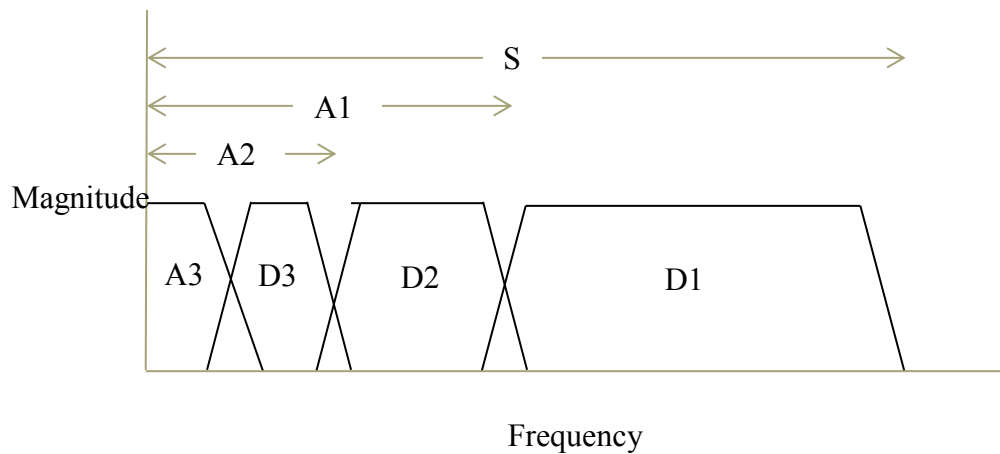


Figure 37: Frequency allocation of wavelet transform.

Figure 38 and Figure 39 depict the decomposition of the signals in waveform and spectrogram, respectively, using 1.5 seconds of the Beatles' song "Let It Be" (starts from 13.5 seconds and ends at 15 seconds; sampling rate 22050Hz) to demonstrate the four-level UWT using db4.

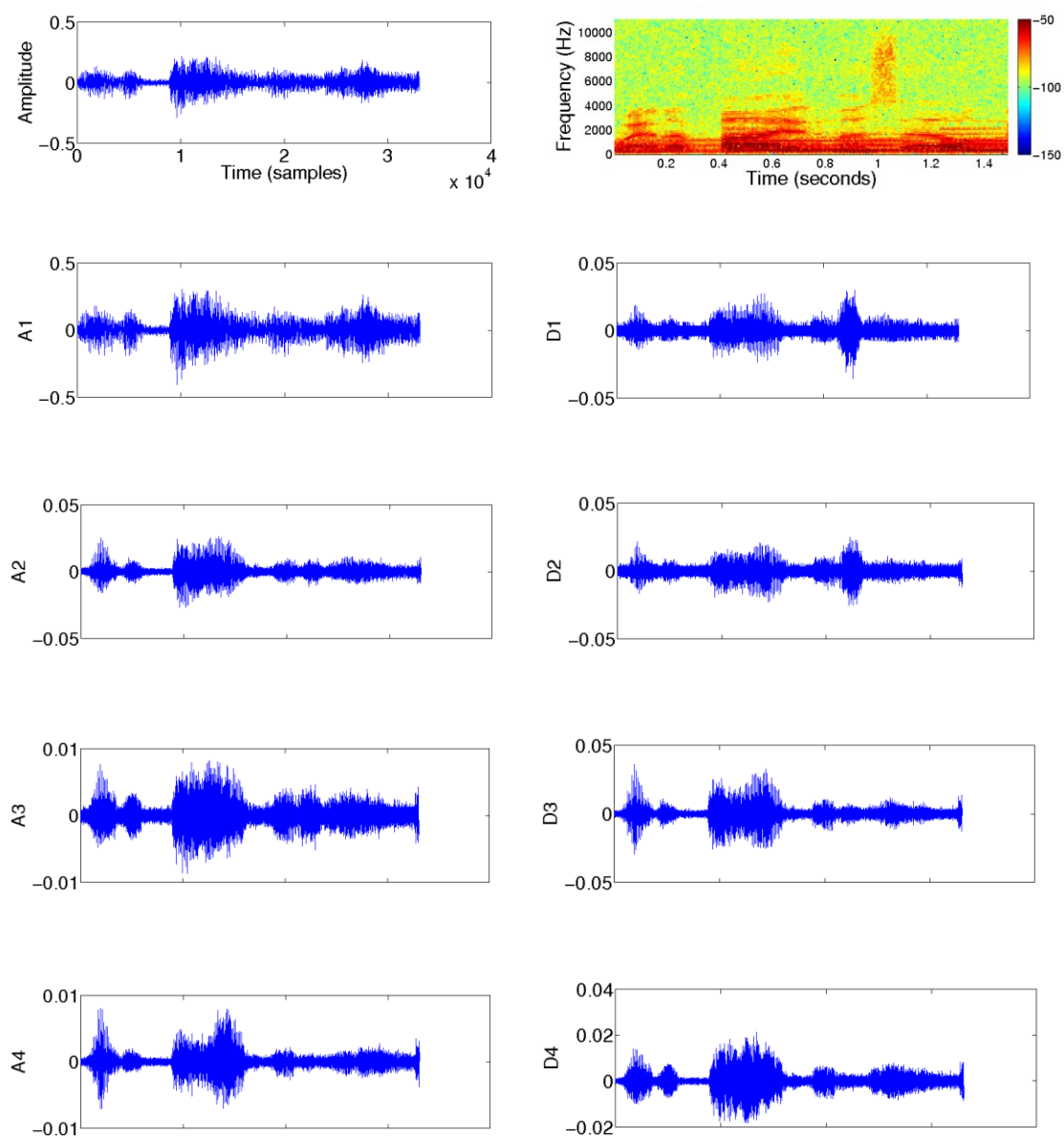


Figure 38: Amplitude and time representation of 1.5 seconds of “Let it be.” Top row represents the original signal.

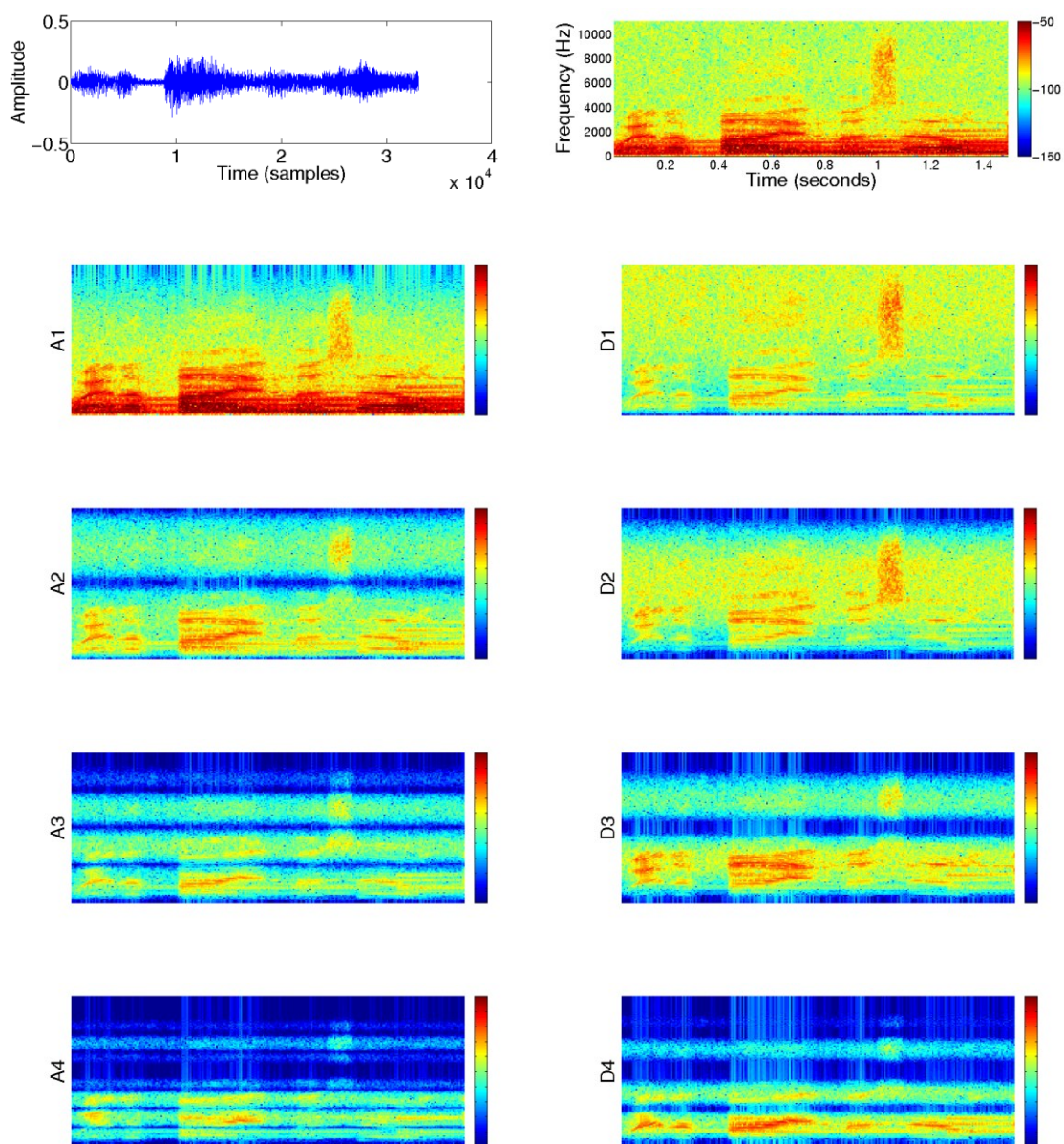


Figure 39: Frequency and time representation of 1.5 seconds of “Let it be.” Top row represents the original signal.

From Figure 38 (waveform), we notice that the general waveform of A1 is similar to the raw signals but the amplitude appears to be slightly higher. However, as the level

of decomposition increases, the similarity in shape between approximation and raw signals as well as the amplitude, for both the approximation and detail components, drastically decreases. From the perspective of a spectrogram depicted in Figure 39, we note that high frequency components are filtered out in the approximation coefficients as the level of decomposition increases which coincides nicely with the frequency allocation scheme depicted in Figure 38. Since human vocal frequency has a ceiling of approximately 1500 Hz while high-pitched musical instruments, such as a piccolo or violin, whose fundamental frequencies of high notes are in the range of 2000 Hz to 4000 Hz, we hypothesize that using certain level of approximation coefficients to represent the raw audio signals would improve the tasks of key and chord recognition.

To perform a wavelet transformation, we first choose an appropriate base wavelet which matches the shape of the target audio signals. This is usually done by visual comparison and thus subjective in nature. Therefore, among families of wavelet, such as Daubechies, Symlet, Haar, Coiflet, and Biorthogonal, we choose Daubechies (db) and Symlet (sym) as our candidates for UWT. Both wavelet families have an order range from 2 to 20 which are denoted as Db2 ~ Db20 and Sym2 ~ Sym20. Once a family of base wavelets is selected, we need to determine the level of wavelet decomposition. A higher order base wavelet is generally smoother than a lower order one while wavelet decomposition at a higher level also gives a smoother representation of the raw audio signals. Due to the large number of combinations from nineteen orders of db and sym wavelet families as well as different levels of approximations, we randomly picked one song from each of the 12 Beatles' albums to test what combinations work well so we can

narrow down the number of order and approximation levels. From the preliminary experiment, we determined that orders 4 ~ 8 of Db and Sym with decomposition levels 3 ~ 4 had potential to produce good results for the two tasks. Therefore, we have a total of 2 (base wavelets) x 5 (orders) x 2 (levels) wavelet configurations for the UWT. A selection criterion is in order so that the best set of approximation coefficients is used to represent the raw signals.

Many wavelet selection criteria, such as maximum-energy and minimum Shannon entropy based criteria as well as correlation and information-theoretic based criteria, have been proposed by Yan (2007). Recall that our goal of this wavelet preprocessing step is to obtain smoother approximations of the raw signals by removing non-periodic components such as transients or percussion sounds as well as high order harmonics that do not positively contribute to the recognition of keys and chords. From this perspective, it suggests that selecting wavelet approximation with minimum Shannon entropy would be a good selection criterion. The Shannon entropy of the approximation coefficients is defined as Equation 20:

Equation 20: Shannon entropy

$$E_{entropy}(S) = - \sum_{i=1}^n p_i \cdot \log_2 p_i$$

where S is the signal and p is the energy probability distribution of n wavelet approximation coefficients.

However, from the insight that we gain from Figure 38 and Figure 39, we notice that as the level of approximation increases, higher frequency components are discarded which result in the overall waveform to deviate severely from the raw signals. In other words, employing entropy-based criterion alone tends to produce unwanted or overly smoothed results since such criterion is solely based on the content of the coefficients. Therefore, a similarity-based criterion should also be employed so that our search for the best approximation also takes the raw audio signals into consideration. Equation 21 depicts how similarity is measured between a wavelet approximation and raw signals using a correlation coefficient.

Equation 21: Wavelet similarity measure

$$C(S,A) = \frac{C_{SA}}{\sigma_S \sigma_A}$$

where S is the signal and A is wavelet approximation. C_{SA} denotes their covariance. σ_S and σ_A are the standard deviation of S and A, respectively.

Since the length of the raw audio signals must be a multiple of 2^N for UWT, we satisfy this requirement by removing the last 2^N sampled raw data points, i.e., we remove at most $(2^N - 1)$ samples for the N-level UWT from the raw signals. Removal of up to 7 or 15 trailing samples has virtually no impact on chroma representation since the wavelet transformation maintains the original sampling rate of 22050 Hz. Therefore, the removed trailing samples represent a duration of at most 7×10^{-4} seconds. In other words, the

dimensions of the denoised signals will remain the same for each song regardless of the values of N ($=3\sim 4$) under UWT.

3.4.2 Chroma Extraction and Variants

As discussed in Section 3.4.1, to reduce transients and higher harmonics, we apply a novel approach by employing undecimated wavelet transform on the raw audio signals and use the wavelet approximation to extract the chroma feature. This is in contrast to most of the methods proposed in the literature which apply low-pass or median filters on the pitch (Fujishima, 1999; Peeters, 2006; Varewyck, et al., 2008) or chroma representations (Oudre, et al., 2011), or both representations (Bello & Pickens, 2005; Mauch & Dixon, 2010) as a smoothing technique for noise and transient reduction. In other words, the low-pass or median filters operate on the magnitude spectrum, under the assumption that peaks of frequency magnitude concentrate on a handful of frequency bins to filter out noises and transients. Therefore, this is in contrast to our wavelet-based transform operating on wave signals in the time domain. The second novelty of our approach is the employment of the two wavelet selection criteria to reduce attack transients and higher harmonics, as described in Equation 20 and Equation 21 by dynamically selecting the best wavelet approximation. In the literature, many of the proposed methods simply cut off frequencies above certain arbitrary levels. For instance, Khadkevich and Omologo (2009) extract chroma vectors between 100 Hz and 2k Hz for chroma vectors while Pauws (2004) cuts off frequencies above 5 kHz.

Since the wavelet transform is undecimated, the UWT approximation coefficients represent the signal with the same sampling rate as the original WAV signal. The wavelet-transformed signals are used for chroma feature extraction. We input these wavelet coefficients as denoised signals into the Chroma Toolbox (Müller & Ewert, 2011) where a Constant Q Transform (CQT), which we reviewed in Section 2.2.1, with a multi-rate filterbank is used. Table 10 displays the sampling rates for ranges of pitches and hop size in terms of fractions of analysis frame length while Table 11 shows a partial list of frequencies, bandwidths, and quality factor Q.

Table 10: Sampling rate for CQT

MIDI Pitch	Piano Note	Sampling Rate (f_s)	Hop Size
21 – 59	C0 – B3	882	1/2
60 – 95	C4 – B6	4410	1/2
96 – 108	C7 – E8	22050	1/2

Table 11: Specification of frequency, bandwidth, and Q

Note	MIDI #	Frequency	Bandwidth (Hz)	Sampling Rate (Hz)	Bandwidth / Sampling rate	Q Factor
A3	57	220.00	8.80	882	.0100	25
A#3	58	233.08	9.32	882	.0106	25
B3	59	246.94	9.88	882	.0112	25
C4	60	261.63	10.47	4410	.0024	25
C#4	61	277.18	11.09	4410	.0025	25
D4	62	293.66	11.75	4410	.0027	25

To understand the effects of using wavelet denoised audio signals in the performance key and chord recognition, we also employed three variants of chroma features – CLP, CENS, and CRP – for performance comparison. These chromagrams are extracted using the Constant Q transform using the parameter specification described in Table 10 and Table 11. Therefore, their differences all lie in the selection and further transformation of the spectral content determined from the CQT.

CLP, Chroma Log Pitch, is a chroma feature with logarithmic compression. The energy e in each frequency bin is first transformed with $\log(\eta \cdot e + 1)$ where η is a suitable positive constant and then normalized using Equation 8. CENS, Chroma Energy Normalized Statistics, considers short-time statistics over energy distribution within the chroma bands using a quantization function which assigns discrete values (0 ~ 4) based on the energy level of each pitch class. Subsequently, the quantized values are convolved with a Hann window which results in a weighted statistics of energy distribution. CRP, Chroma DCT-Reduced log Pitch, is obtained from the CQT by applying a logarithmic compression similar to that of CLP followed by a discrete cosine transform (DCT). Finally, our undecimated wavelet transformed with N-level approximation, CUWT-N, is fully described in Section 3.3.1. Table 12 summarizes all variant chromagrams that we use in our experiments.

Table 12: Variants of chroma features used in experiments

Name	Feature Description
CLP	Chroma Log Pitch
CENS	Chroma Energy Normalized Stats (no log)
CRP	Chroma DCT-Reduced log Pitch
CUWT-N	UWT on raw signals to produce CLP

In the following discussion, we use these specific names to address different variants of chroma features for performance comparison. However, for a general discussion of chroma features without the need to address a specific variant, we use CF_i to denote the chroma feature of the i th frame.

3.4.3 Local Keys Recognition

To achieve higher performance of chord recognition, we first extract a bag of local keys (BOK) of a music piece for two reasons. First, since a key typically covers wider segments of the music piece than a chord, we assume that extracting local keys from a chromagram is less impacted by noises (such as percussion) due to their wider coverage than that of chords in a music piece. Second, given local keys of a music piece, we can predict prominent pitches that reside within the key; therefore we have a higher chance of extracting the correct chords from a noisy chromagram.

Our estimation of BOK uses bag of frames (BOF) as the data source. The BOF approach has been used as a global musical descriptor for several audio classification problems involving timbre, instrument recognition, mood detection, and genre classification (Pachet & Roy, 2008). In BOF, each acoustic frame obtained from the signal processing methods, like the ones we discussed in Section 2.2, is considered a word using Latent Dirichlet Allocation (LDA) for document classification. An application of LDA in chord and key extraction, by Hu and Saul (Hu & Saul, 2009), is briefly described in Section 2.3. In the BOF approach, as the name “bag” suggests, acoustic frames are not treated as time series but are often aggregated together to be analyzed using various statistical methods for computing statistics such as means or variance across all frames. Also reported by Pachet and Roy (2008), BOF serves as a data source for Gaussian Mixture Models (GMM) for more complex modeling in supervised classification context to train a classifier. In our application, we feed BOF into the IGMM to produce BOK (bag of local keys). For the remainder of the section, we discuss how the IGMM, discussed in Section 3.2, is used to generate a bag of local keys.

Equation 12 and Equation 13 govern how to sample a new (or existing) configuration c_i for data point y_i . The idea is that for each y_i in $Y = \{y_1, y_2, \dots, y_n\}$ that we process iteratively, we first use Equation 12 and Equation 13 to probabilistically determine whether it was generated by a local key that was not seen before or by one of the existing local keys; based on the determination, we generate a new θ as the new unseen local key for y_i or associate y_i to an existing local key. Therefore, if c_i is obtained by Equation 12, we simply associate y_i with an existing θ_j . If c_i is obtained through

Equation 13, we sample a new θ from G as described in Figure 26 using Equation 18 and Equation 19. Mean (μ_i) and covariance (Σ_i) of Gaussian key, using a mix of harmonic and natural minor scales, are encoded the same as that of the symbolic domain which is described in Table 7. We implement Σ_i as a diagonal matrix and assign a value of 1 for notes present in the key.

We input Y into the IGMM to iteratively generate local key samples that most likely produced Y . We arbitrarily generate the first key sample and after four burn-in iterations, these samples start to converge to the estimated local keys very quickly, usually in less than 12 iterations. Note that a sample generated from an iteration contains all possible local keys used in the entire music piece. We iterate s times to obtain s samples of local keys and discard those that cover less than 10% of the chromagram. Table 13 summarizes the algorithm.

Table 13: Key sampling algorithm using IGMM (audio)

Obtain peak pitches Y (triad peak-picking)
Initialize G ; Initialize c_1 and θ_1 to random values.
For $i = 1: s$ samples
For $j=1: n$ ($n = \text{size of } Y$)
Sample a new c_i based on Equation 12 and Equation 13
If a new θ_i is required
Sample a new θ_i
Update α based from iteration (i-1) using Equation 14
Regroup Y based on all sampled θ_i ;
Discard θ_i 's that cover less than 10% of the chromagram; output Θ as a bag of local keys

Each frame of the chromagram represents the energy level of 12 pitch classes and we want to use prominent pitches to quickly estimate keys within the whole music piece. Since triads (major and minor) are the most prevalent chords in pop music, we apply a simple peak-picking algorithm on each frame to choose the most likely major or minor triad to represent the frame for key recognition. The most likely preliminary triad is the one, among 24 triads, that possesses the highest energy. We denote y_i as the triad representing frame i and denote $Y = \{y_1, y_2, \dots, y_n\}$ for n frames of a music piece. Note that Y is a series of preliminary triads that we use to estimate local keys and therefore not the results of chord recognition.

Based on Equation 12, Equation 13 and the sampling process described in Table 13, we see that data points in Y are assumed to be exchangeable which is a prerequisite of

a Dirichlet mixture model. In our case, it means that for every finite subset of Y , the joint distribution of them is invariant under any permutation of the C indicator variable. Obviously, exchangeability does not exist in music since musical notes contained therein are products of careful orchestration by composers and performers and random exchange of them within the piece render them unrecognizable to listeners. However, for tonal music, its tonal centers (keys) dominate the use of specific pitch hierarchy of the tonic, so the random exchange, in terms of their placement in the music piece, of pitches would have minimal effect in our estimation of BOKs. In other words, since our goal is not to extract local keys on a frame-by-frame basis but to quickly estimate what local keys are present in the target music piece, we can uphold the presumption of exchangeability in the IGMM.

3.4.4 Chord Recognition

The goal of this component is to recognize six chord types (maj, min, aug, dim, sus, and none) by taking advantage of the key information obtained in Section 3.3 to transform the chromagrams extracted from Section 3.4.3 to mimic human perception of keys and chords. The idea is that once we have the keys extracted, we consider only pitch energy of diatonic tones and further adjust chroma energy using the K-K profiles described in Section 2.1.3.

We use binary templates TKey to represent the keys that we have determined in Section 3.4.3. Specifically, for C major key, $\text{TKey}_{\text{maj}} = [1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1]$; for C

minor key, we use a mix of harmonic minor and natural minor scales so that $TKey_{\min} = [1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1]$. Similarly, binary templates are used for chord classes. Therefore, a C major chord has a template $TChord_{\text{maj}} = [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$.

Given the key information, the two K-K profiles can be adopted to promote prominent while suppress less prominent pitches in a CF_i extracted from Section 3.4.1. The K-K profile for the C major key has the format of $KK_{\text{maj}} = [6.35 \ 2.23 \ 3.48 \ 2.33 \ 4.38 \ 4.09 \ 2.52 \ 5.19 \ 2.39 \ 3.66 \ 2.29 \ 2.88]$; for the C minor key, $KK_{\min} = [6.33 \ 2.68 \ 3.52 \ 5.38 \ 2.6 \ 3.53 \ 2.54 \ 4.75 \ 3.98 \ 2.69 \ 3.34 \ 3.17]$. We denote $KK_{\text{determined}}$ as the key profile for the key(s) determined from Stage III, as described in Table 9, by circular shifting either KK_{maj} or KK_{\min} .

Each time we circular shift $TChord_c$, we compute the following dot product to obtain the adjusted chroma energy for frame i :

Equation 22: Adjusted chroma energy

$$CF_{i_adjusted} = CF_i \cdot TKey \cdot KK_{\text{determined}} \cdot TChord_c$$

where $TChord_c$ template corresponds to the highest energy sum, $CF_{i_adjusted}$, of the above dot product is the recognized chord for frame i .

After each frame is assigned a chord label as described above, we perform one smoothing step to erase sporadic chord labels due to the unavoidable noise in a chromagram. A sporadic chord label, in our case, is defined as a chord assignment that lasts only one frame among its neighboring frames while a stable chord label spans at

least two frames. Assuming we have a segment of chord labels ‘PQR’ where ‘Q’ is a sporadic chord label while ‘P’ and ‘R’ are stable ones before and after Q, respectively, we adopt the following rules to correct a sporadic ‘Q.’

- For $P = R$, we change Q to P.
- For $P \neq R$, we adjust Q to either P or R by examining the duration of chords P and R in the entire music piece. We denote the number of occurrences for P, Q, and R as p , q , and r , respectively. The principal idea is that a chord label with lower occurrences in the whole music piece tends to move to chords with more popular chords but not the other way around. Table 14 depicts the rule.

Table 14: Correction rule for sporadic chord labels

Given $P \neq R$ and $(p,q,r) \rightarrow$ Adjust PQR to		
$p > q > r$	$q > p > r$	$r > p > q$
\rightarrow PPR	\rightarrow PPR	\rightarrow PPR
$p > r > q$	$q > r > p$	$r > q > p$
\rightarrow PRR	\rightarrow PPR	\rightarrow PRR

3.5 Evaluation Metrics

For local key recognition, we use precision, recall, and F-measure. These metrics are based on conditional probabilities and widely used in information retrieval tasks. We

follow the definition provided by Roelleke (2013). For document retrieval tasks, precision and recall are described as the following. Given a set of retrieved documents and a set of relevant documents,

- Precision: the portion of retrieved documents that are relevant
- Recall: the portion of relevant documents that are retrieved

We give a formal definition of precision and recall based on conditional probabilities, in the context of local key recognition with query q .

Equation 23: Precision

$$\begin{aligned} \text{precision}(q) &:= P(\text{relevant} | \text{retrieved}, q) \\ &= \frac{P(\text{retrieved}, \text{relevant} | q)}{P(\text{retrieved} | q)} \end{aligned}$$

Equation 24: Recall

$$\begin{aligned} \text{recall}(q) &:= P(\text{retrieved} | \text{relevant}, q) \\ &= \frac{P(\text{retrieved}, \text{relevant} | q)}{P(\text{relevant} | q)} \end{aligned}$$

The F-measure is the harmonic mean of precision and recall. It is defined as the following.

Equation 25: F-measure

$$F = 2 \times \frac{Precision \times Recall}{(Precision + Recall)}$$

For chord recognition, there are many different terms used such as average overlap score, proposed by Oudre (2011), relative correct overlap, described by Mauch and Dixon (2010), and Harte’s chord symbol recall (Harte & Sandler, 2005), which are essentially recall measure defined in Equation 24.

Since we use Harte’s chord transcription as the ground truth (GT), we follow his definition of chord symbol recall which is defined as the summed duration of time periods where the correct chord has been identified, normalized by the total duration of the evaluation data. The CSR is formally defined below:

Equation 26: Chord symbol recall

$$\text{Chord Symbol Recall (CSR)} = \frac{| \text{estimated segments} \cap \text{annotated segments} |}{| \text{annotated segments} |}$$

where $| \cdot |$ represents the duration of a set of chord segments.

Chapter 4 Experimental Results

In this chapter, we discuss experimental results of applying the method of recognizing keys and chords from two musical data formats – symbolic (MIDI) or real audio (WAV) – of songs from the Beatles. The Beatles’ 12 albums (thirteen CDs) were converted into the WAV format for audio key and chord recognition. Among the 180 songs from the CD albums, we are able to find 159 in the MIDI format from the Internet which we use as the symbolic dataset for the two tasks. Section 4.1 describes the characteristics of the Beatles’ albums. Experimental results from the symbolic and acoustic audio domains are discussed in Sections 4.2 and 4.3, respectively. In Section 4.4, we provide a detailed comparison, taking different experimental setting proposed in the literature, of our experimental results with that of reported state-of-the-art methods. In the last section, as a concluding remark, we provide a high-level pro-and-con analysis of supervised, unsupervised, and knowledge-based systems that we discussed in Chapters 2, 3, and 4.

4.1 The Beatles Albums

We exclusively use the Beatles’ as our dataset in this experiment for three reasons. First, their music is widely regarded as the era’s most influential force which, as described by Schinder (2008, p. 159), “revolutionized the sound, style and attitude of popular music

and opened rock and roll's doors to a tidal wave of British rock acts." Schinder further stated that, "The band's increasingly sophisticated experimentation encompassed a variety of genres, including folk-rock, country, psychedelia, and baroque pop, without sacrificing the effortless mass appeal of their early work." They produced 12 albums with a total of 180 songs over three decades and many MIDI composers have made MIDI versions of the Beatles' collection available over the internet. Second, due to their popularity, full score of their songs are in print (Lowry, 1988) as well as detailed analyses of each song are on the internet (Pollack, n.d.) which can readily serve as the ground truth (GT) to understand the performance of a computerized key and chord recognizer. Third and most importantly, Harte's transcription project (Harte, et al., 2005) annotated all 180 songs with precise time information (start and end time) for chords. Table 15 and Figure 40 provide the basic timing information and chord type distribution (Harte, 2010), respectively, for the Beatles' 12 albums (13 CDs).

Table 15: 12 albums of the Beatles

Album Name	# of Songs	Time (mins:secs)
Please Please Me	14	32:45
With the Beatles	14	33:24
A Hard Day's Night	13	30:30
Beatles for Sale	14	34:13
Help!	14	34:21
Rubber Soul	14	35:48
Revolver	14	34:59
Sgt. Pepper's Lonely Hearts Club Band	13	39:50
Magical Mystery Tour	11	36:49
The Beatles (the white album; CD1 / CD2)	17 / 13	46:21 / 47:14
Abby Road	17	47:24
Let It Be	12	35:10
Total	180	8 h: 8 min: 48 secs

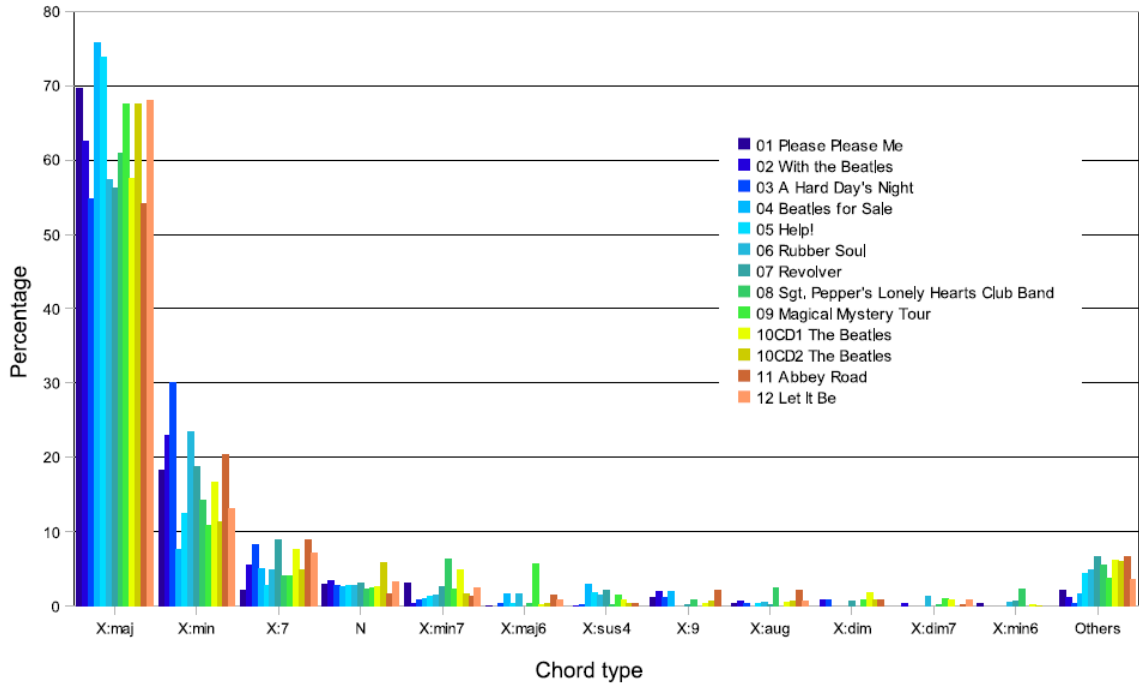


Figure 40: Chord type distribution for the Beatles' 12 albums (Harte, 2010)

4.2 Symbolic Domain

One hundred fifty nine MIDI-based songs mimicking the Beatles' collections were downloaded from the internet for the two tasks.

4.2.1 Keys Recognition

For key recognition, we use Pollack's notes (Pollack, n.d.) as the ground truth to judge the effectiveness of the IGMM key-finding algorithm since his notes have detailed information regarding each song's home key as well as modulations. However, since his

notes do not have the complete sequence for key modulations, we simply gather the home key and all modulations described in his notes and compare them with the results obtained from the IGMM. In other words, we treat keys obtained from IGMM and Pollack’s notes as a bag of local keys and compare them as such.

One interesting and challenging aspect of using MIDI files for model validation (both keys and chords) is the need to detect if a target MIDI file has been transposed to a different key since the detected key of a transposed piece is, by definition, different from the original key and the certainty (or lack) of transposition help us determine whether the algorithm has correctly detected the key. Musicians very often transpose songs to be sung by different vocalists with different vocal ranges or the original chords are difficult to perform by their instruments. It is obvious that the key samples obtained from the IGMM iterations or any key-finding algorithms alone cannot detect and confirm the presence of key transposition. However, since we determine keys and chords in an iterative fashion in the IGMM, we can transpose a $Chord_{sample}$ (a sequence of chords for the entire target piece) based on the chromatic scale and see if a transposed $Chord_{sample}$ is closer to the GT chords. Specifically, we circular shift each $Chord_{sample}$ to find a best match between the chord samples and Harte’s GT. Such shifts are only performed when there is a disagreement among the key samples generated by IGMM, K-S key-finding algorithm, and all published GT. For example, for the song “Hold Me Tight,” the IGMM determines it as C Major which is the same as the K-S key-finding algorithm, but Pollack’s notes ascertain it as F Major. Since the keys disagree, we circular shift the Gaussian chords 1 ~ 11 positions which results in the fifth position producing a drastic shorter Euclidean

distance between $Chord_{sample}$ and Harte’s annotation. Therefore, we determine that the MIDI file is transposed from the key of F Major to C Major and confirm that the key determined by IGMM is correct.

To get a baseline understanding of how the IGMM performs in key finding, we first compare the performance of the IGMM with that of the K-S algorithm (implemented in Toivainen and Eerola’s MIDI Toolbox (2004)) in finding “home” keys. In the K-S algorithm, a home key is the key profile that produces the highest correlation with the given MIDI. In IGMM, similarly, we designate the key that has the highest percentage of notes assigned to it as the home key. Note that the K-S algorithm is not designed to detect songs with key modulations and there are 26 songs (out of 159) with multiple keys. We further categorize songs into single and multiple keys to better understand the performance of the two methods. For a fair comparison, if Pollack’s GT does not specify a “home” key for a song with multiple keys, we award one point to algorithms that produced a key with the highest correlation (for K-S) or percentage (for IGMM) which is part of the GT multiple keys. The results are depicted in Table 16. We note that IGMM outperforms the K-S algorithm for both categories of songs.

A more reasonable performance measure for the key information retrieval task is to use precision and recall. Since the K-S key-finding algorithm is not designed to recognize keys for songs with modulations but the IGMM is capable of doing so, it is impossible to apply such measure on the two algorithms for fair comparison. Therefore,

we only report such measure for the IGMM key-finding task which is described in Table 17.

Table 16: Experimental results of key finding using K-S and IGMM

Ground Truth (Pollack’s notes)	# of songs	K-S key finding		IGMM key-finding	
		# of	% of	# of	% of
		songs correct	songs correct	songs correct	songs correct
Single key	133	92	69.2%	101	75.9%
Multiple keys (2 ~ 4 key modulation)	26	16	61.5%	20	76.9%
Overall	159	108	67.9%	121	76.1%

Table 17: Precision, recall, and F-measure for the IGMM key-finding task

	# of songs	Precision	Recall	F-Measure
Single key	133	.752	.865	.804
Multiple keys (2 ~ 4 key modulation)	26	.718	.587	.646
Overall	159	.742	.814	.776

We notice that the precision for songs with modulations is just slightly lower than songs with single keys. The low recall for songs with multiple keys (58.7%) indicates that IGMM tends to retrieve fewer relevant keys than that of the GT. This phenomenon can be explained by the crowded-tables-get-more-crowded property of the CRP sampling process in IGMM.

4.2.2 Chords Recognition

In contrast to the lack of timing information for keys, Harte’s annotations contain a sequence of chords with exact start and end times for each song. However, since MIDI music are not an exact replica of the original in terms of length and timing, it would be impossible to perform comparisons based on the timings of chords between the MIDIs and the originals. Therefore, we employ the technique of dynamic time warping (DTW) to compare IGMM’s annotation with Harte’s GT. DTW uses a similarity matrix (SM) to determine the similarity between two given sequences. Since we use a 12 dimension Gaussian to represent a chord in IGMM, we convert Harte’s chord annotations into the same 12-dimensioned Gaussian format and inject a Euclidean distance into each cell in the SM as the basis for finding the similarity between the two chordal sequences. We follow Paiement (2005) to employ the Euclidean distance as a way to represent the psychoacoustic dissimilarity between the two sequences. Table 18 depicts a sample Euclidean distance between two sets of chords based on the encoding profiles described in Table 7. We denote the Euclidean distance for $Chord_{sample}^j$ as $DistChord_{sample}^j$.

Table 18: Sample Euclidean distance of chords

	N	G	D:7	C:7	D	B: min	A	A:7	E: min
G	6.6	0	8.1	8.1	7.5	6.2	9.3	8.1	6.2
C:maj7	6	5.6	8.4	4.7	8.9	7	7.8	7.2	3.6
D:7	7.2	8.1	0	8.6	3	6.9	8.1	8.6	9.7
C:7	7.2	8.1	8.6	0	9.7	9.7	8.8	8.2	6.9
C	6.6	7.5	8.1	3	9.3	9.3	8.2	7.7	6.2
E:min	6.6	6.2	9.7	6.9	9.3	7.5	7.5	6.9	0
B:min	6.6	6.2	6.9	9.7	6.2	0	9.3	9.7	7.5
A:7	7.2	8.1	8.6	8.2	8.1	9.7	3	0	6.9
D	6.6	7.5	3	9.7	0	6.2	7.5	8.1	9.3

Apparently the Euclidean distances such as those described in Table 18 are entirely dependent on the encoding profiles depicted in Table 7. An identical match between a chord generated by IGMM and the GT has an Euclidean distance of zero. The second shortest Euclidean distance has a value of 3 if the two chords are one note apart such as the C major chord and the C7 chord. Using the chord sequences produced by IGMM and the GT, we can construct an SM based on their Euclidean distances. Figure 41 shows a set of 12 grayscale images where each image represents one SM for the song titled “Hold Me Tight.” Zero Euclidean distance is represented by a white color cell while the largest distance is represented as black cell. Recall that we generate an SM for each $Chord_{sample}$ and we circular shift $Chord_{sample}$ 11 times to check for the presence of key so there are a total of 12 images in the plot where the top left image, which we will call the original MIDI chords, represents the SM between the IGMM chord sequences and Harte’s GT. The first upward shift of one interval is to the immediate right of the original chords and the fourth shift is the one immediately below the original. The

starting point of the two sequences is on the top left corner of each image. The GT sequence is from left to right for a total of 85 chords while the IGMM sequence is from top to bottom for a total of 537 chords ($n=537$ as the size of Y). The red line indicates the best matched path between the two sequences so that a diagonal straight red line indicates a good match between the determined chords produced by IGMM and the GT. The sum of the Euclidean distance along the red line is displayed on top of each image. In this example, we see that the original MIDI and the GT has a Euclidean distance of 1555.2. However, a 5-interval shift produced a Euclidean distance of 734.5 which is a sharp drop from the original MIDI. Therefore, we conclude that the MIDI is transposed downward 5 intervals from the original recording (from F major to C major). In this case, the K-S key-finding algorithm also determines that the MIDI file has a key of C major which corroborates our finding.

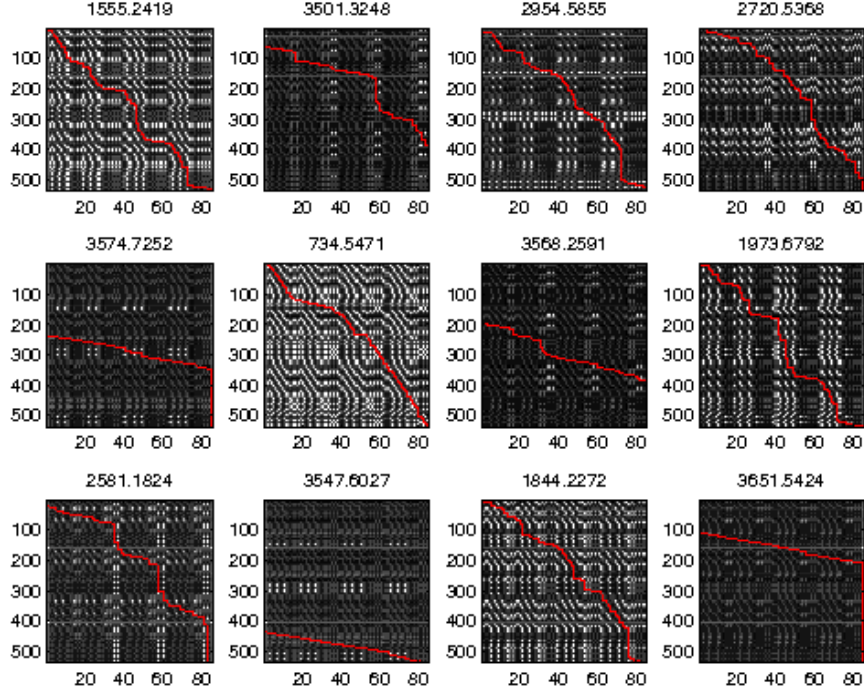


Figure 41: Similarity matrix for the song titled “Hold Me Tight”

Figure 42 shows the Euclidean distances, in 10 bins, between IGMM chords and the GT. We define the shortest Euclidean distance among the 11 circular shifts as $\text{DistChord}_{\text{sample}}^{j_{\min}}$ and the length of such best path for $\text{Chord}_{\text{sample}}^i$ as $\text{length}(\text{DistChord}_{\text{sample}}^{j_{\min}})$. Therefore, the Euclidean distance is calculated using $[\sum_j \text{DistChord}_{\text{sample}}^{j_{\min}} / \sum_j \text{length}(\text{DistChord}_{\text{sample}}^{j_{\min}})]$. Recall that since the IGMM generates s $\text{Chord}_{\text{sample}}$ and each $\text{Chord}_{\text{sample}}$ represents a sequence of chords with a length very close to the length of Y . The similarity measure has $\sum_j \text{length}(\text{DistChord}_{\text{sample}}^{j_{\min}})$ in the denominator, which, in most cases, is very close to $\text{length}(Y) \times s$. We see that 115 songs have a Euclidean distance less than three and the overall average distance is 2.43. The results are encouraging since the shortest Euclidean

distance is 3 (such as $\text{dist}(\text{Gmajor}, \text{G7})$) for any chord mismatch using the profiles illustrated in Table 7.

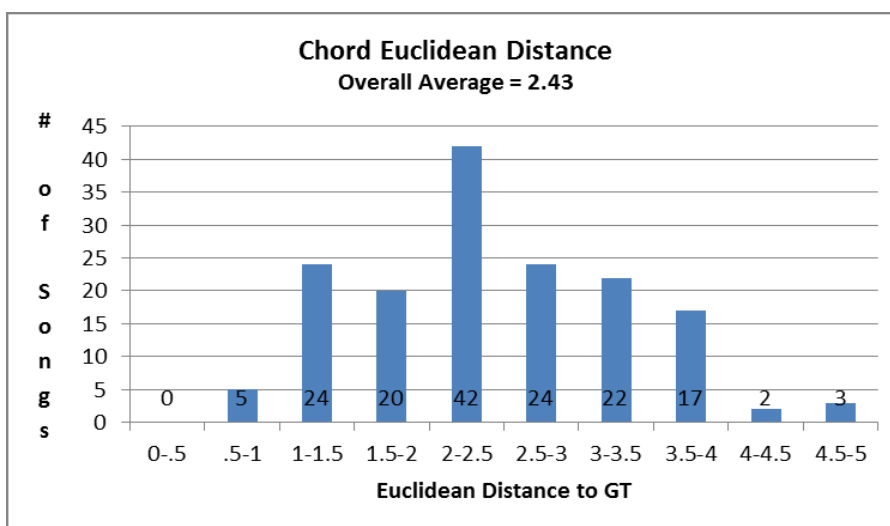


Figure 42: Euclidean distance of IGMM chords to ground truth

Recall the iterative characteristics of our IGMM implementation – a $\text{Key}_{\text{sample}}$ is served as prior knowledge for generating a new $\text{Chord}_{\text{sample}}$. We hypothesized that a positive recognition of key leads to a more accurate extraction of chords and vice versa. Figure 43 displays the box-and-whisker plot of the Euclidean distance between the IGMM chords and the GT categorized by whether their keys are correctly identified. We see that the average Euclidean distances are 2.48 for songs with their keys correctly identified and 2.91 for the other case. This result is encouraging since the shortest misclassification has a distance of 3 given the large number of chord vocabulary used in both the IGMM and the GT. It is also interesting to see that the chord Euclidean distance

difference between the two categories presented in Figure 43 is not large. Specifically, the majority of misclassified keys are closely related to the GT keys such as Major G in GT but minor E from the IGMM, which allows the IGMM to label notes with near-correct chord labels.

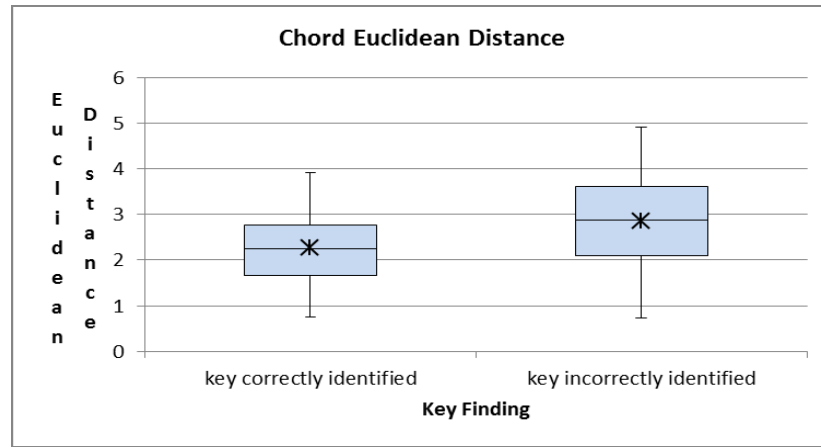


Figure 43: Average chord Euclidean distances between IGMM and GT.

4.3 Audio Domain

We tested the performance of this new approach using 175 songs² from the Beatles' 12 albums.

² We exclude 5 songs out of 180 due to ambiguous tunings. They are: Revolution 9, Love You Too, Wild Honey Pie, Don't Pass Me By, and The Continuing Story of Bungalow Bill.

4.3.1 Key Recognition

Similar to symbolic key recognition, we use musicologist Allan Pollack’s complete annotation of all Beatles’ recordings as the ground truth for the 175 songs in our experiment. Different from the key recognition task using the symbolic MIDI music which differs from the Beatles’ original recordings, Pollack’s annotation faithfully coincides with the 12 albums of recordings in our experiments. Since his notes do not have the complete sequences for key modulations and their timings, we simply collect all keys described in his notes to compare with recognized keys as described in Section 4.2.1. Figure 44 depicts the overall distribution of local keys for the 175 songs. Figure 45 and Figure 46 show the key distribution for songs without key modulations and those with multiple local keys, respectively.

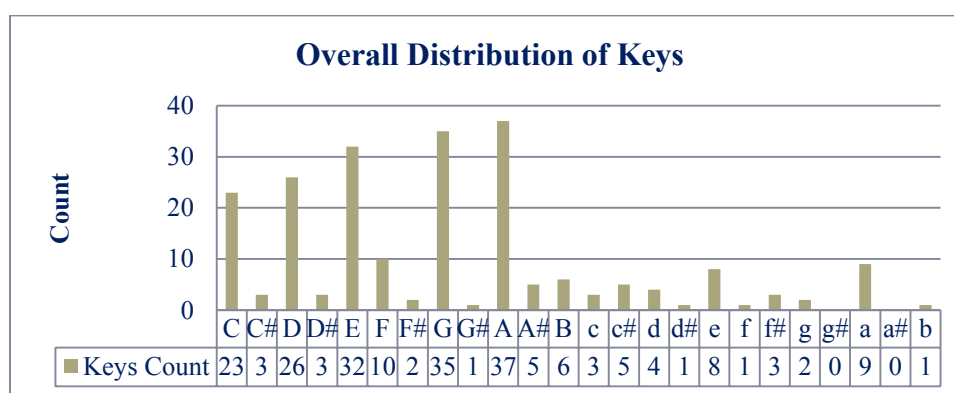


Figure 44: Overall keys distribution

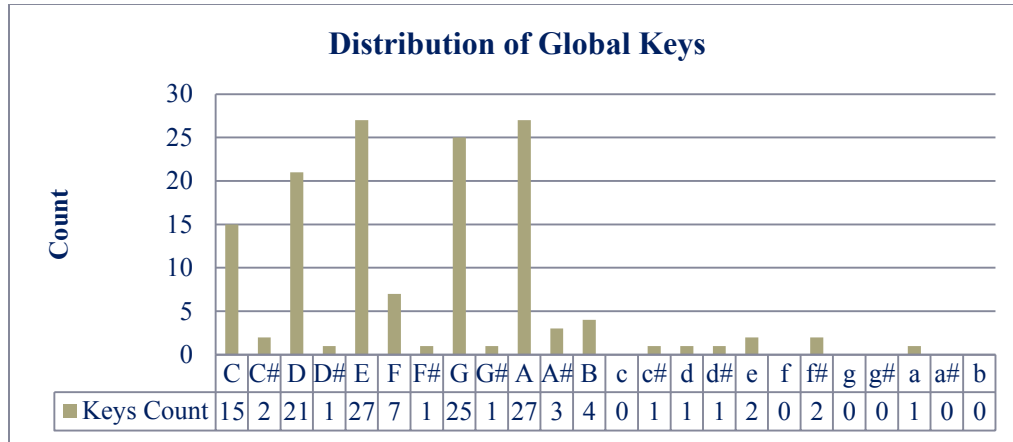


Figure 45: Distribution of global keys

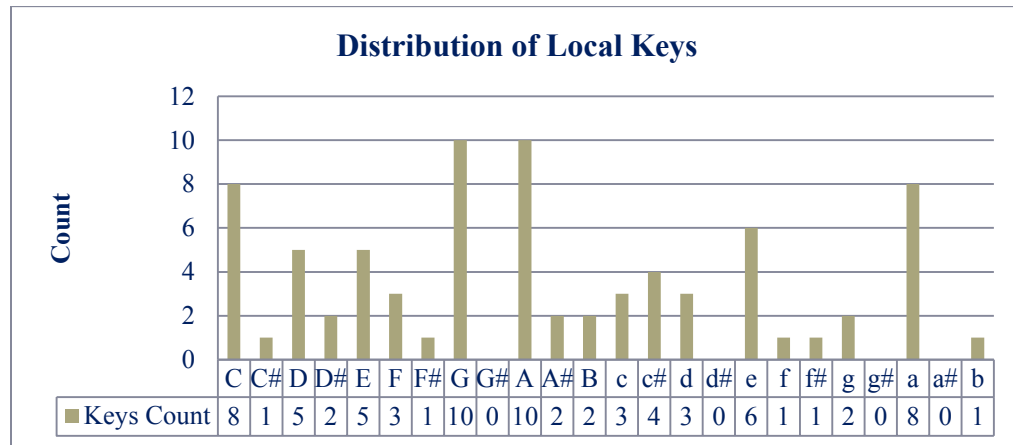


Figure 46: Distribution of local keys

As described in Section 3.4.2 and Table 12, four types of chroma features – CLP, CENS, CRP, and CUWT-n – are extracted from audio signals. Recognized keys that cover less than 10% of the total frames are discarded. Moreover, we strictly compare our results with Pollack’s notes – i.e., related keys (parallel, fifth, relative major/minor) are not counted as correct recognition. We categorize songs into single and multiple keys and

compute their precision, recall, and F-measure values. Figure 47 depicts the overall recognition rates for the five chromagrams used in the experiment. To understand each chromagram's performance due to the presence of multiple local keys, we use Figure 48 and Figure 49 to show the key finding results for songs with one global key and those with multiple local keys, respectively.

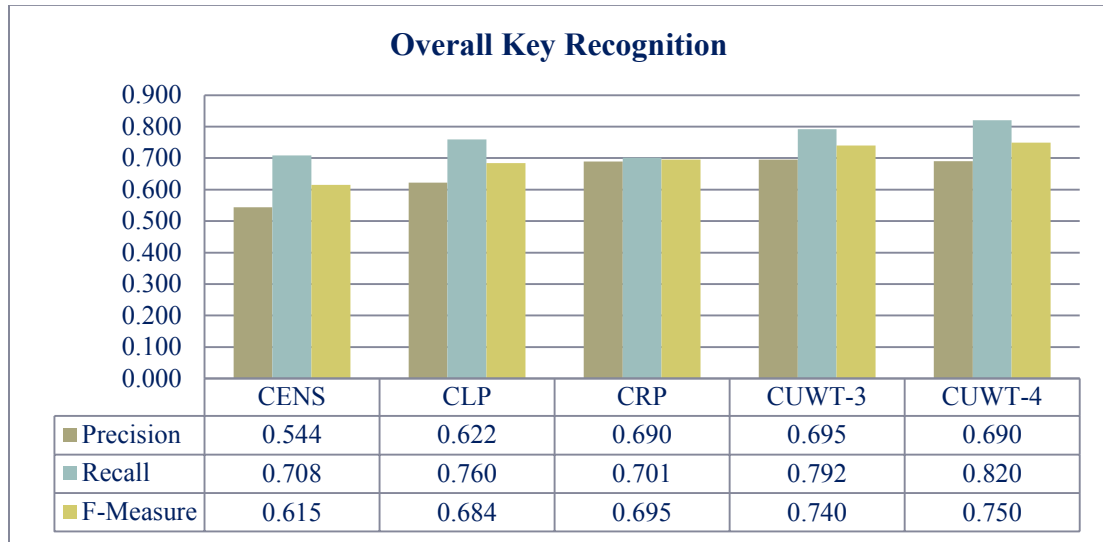


Figure 47: Overall key finding

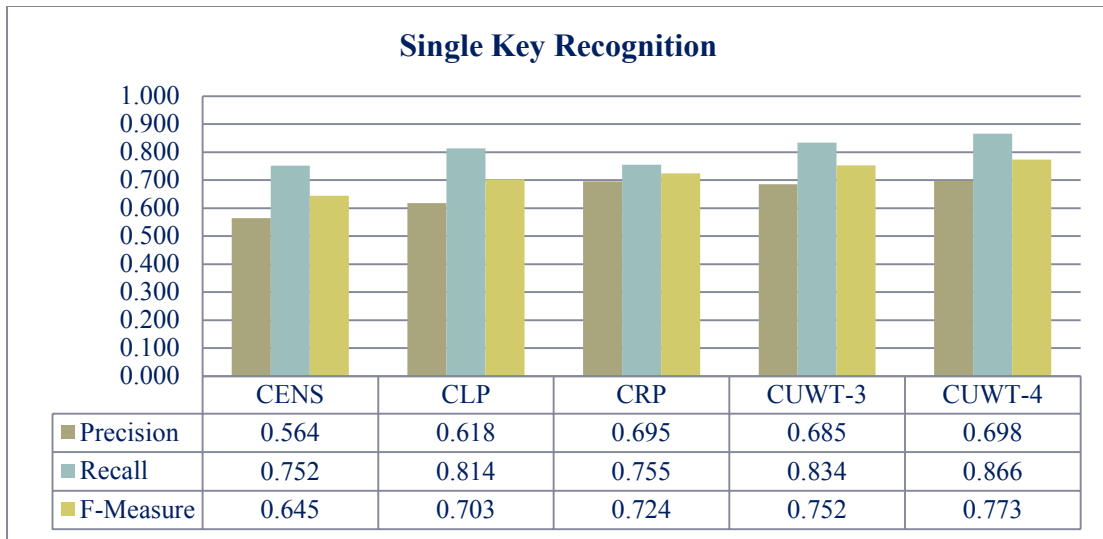


Figure 48: Single key finding

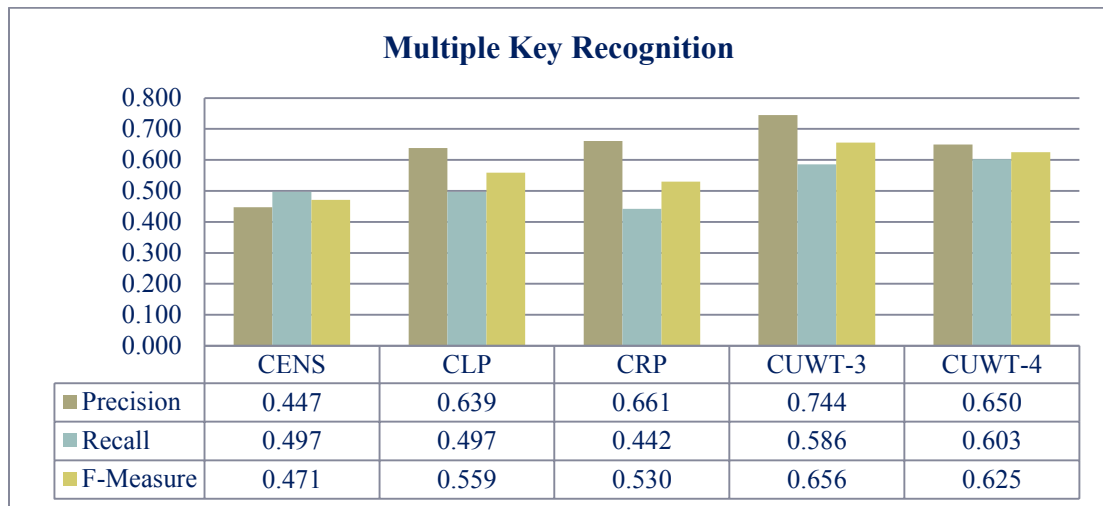


Figure 49: Multiple key finding

Two observations can be made from the above three figures.

- For songs with only global keys, the recall statistic is about 15% ~ 20% higher than the precision statistic. This is in line with our expectation since multiple keys

extracted for songs with only one global key can still have a recall value of one if the global key from the ground truth is part of the set of extracted local keys.

- For songs with multiple local keys, on the other hand, the precision statistic appears to have higher values than that of the recall counterpart, with the exception of the CENS feature.

From the above three figures, we see that the CUWT-4 chromagram – audio signals preprocessed with wavelet transform whose level-4 approximation is used to produce a CLP – consistently yields the highest precision, recall and F-Measure among the five types of chromagrams across musical pieces with global keys and multiple local keys. We can attribute such performance improvement solely to the undecimated wavelet transform of the raw signals before the chroma features are extracted. Therefore, using the CUWT-4 chromagram as the benchmark, Figure 50, Figure 51, and Figure 52 depict the performance improvement of CUWT-4 over the other four types of chromagrams in terms of precision, recall, and F-measure, respectively.

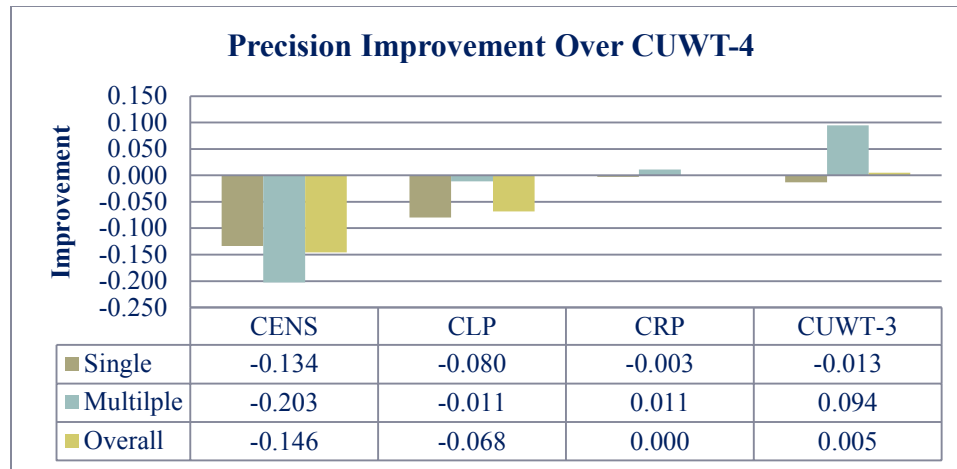


Figure 50: Precision improvement over CUWT-4

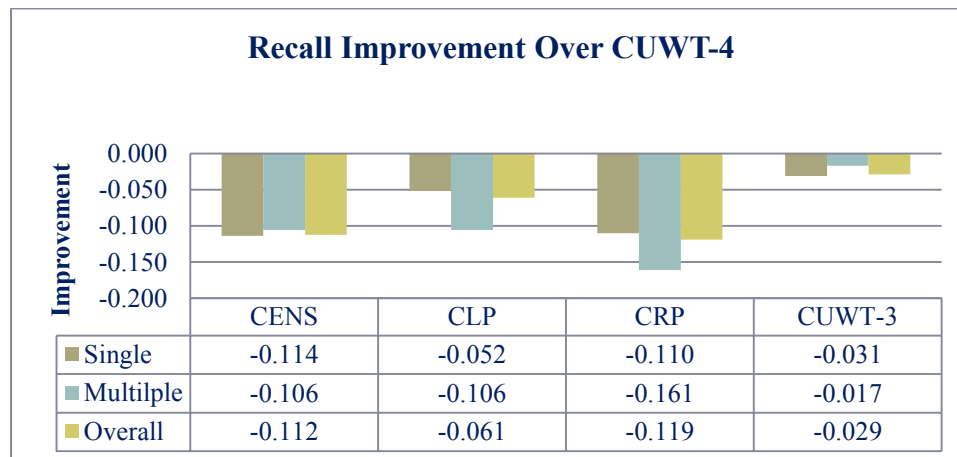


Figure 51: Recall improvement over CUWT-4

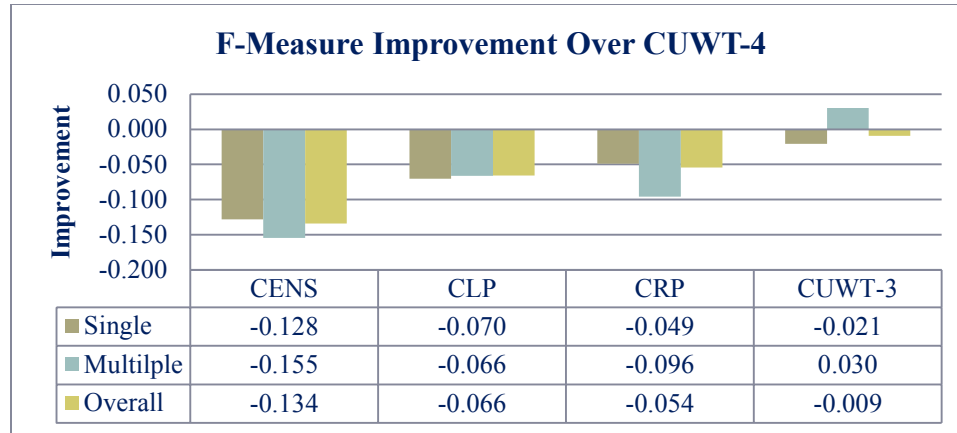


Figure 52: F-measure improvement over CUWT-4

Three observations can be made from the above three figures.

- CUWT-3 and CUWT-4 are superior chromagrams than CENS, CLP, and CRP for the audio key finding task. CUWT-3 and CUWT-4 are very comparable in their performance.
- The highest performance gain of using CUWT-4 is in extracting multiple keys. Specifically, it improves the recall rate by at least 10%.
- Chroma features play a critical role in key finding. A simple wavelet smoothing and approximation of the raw audio signals can yield a significant improvement.

The results are encouraging and clearly indicate that a chromagram using level-4 approximation of UWT in conjunction with an IGMM generative process can be used to recognize single (global) as well as multiple keys (modulations) in a music piece. Since the overall recall is 13% higher than precision, we conclude that the algorithm generates

a high number of false positives. Moreover, the algorithm performs approximately 15% better in recognizing single key than its multiple-key counterpart.

4.3.2 Chord Recognition

In contrast to the lack of timing information for keys, Harte’s annotations (Harte & Sandler, 2005) contain a sequence of chords’ start and end times for each song. Recognition rate is defined as the number of frames that correctly identifies the chord over the total number of frames (Chord Symbol Recall, CSR) for the whole duration of the 175 songs. Since all chords specified in Harte’s annotation can be mapped to the six chord types (five chord type and a “no chord”), summarized in Table 19, all frames are evaluated against the ground truth and no frames are discarded.

Table 19: Six types of chords

Chord Class	Chord Type
Major	maj, maj7, 7, maj6, 9, maj9
Minor	min, min7, minmaj7, min6,min9
Diminished	dim, dim7, hdim7
Augmented	Aug
Suspended	sus2, sus4
N	No chord

Since the average time difference, in terms of song lengths, between Harte's annotation and our chroma features is 262 ms, which is more than two frames (200ms), we suspect that there is a slight misalignment in our WAV files after they are ripped from the audio albums. Therefore, we also report a recognition rate with one frame tolerance on each side of the annotated chord. Figure 53 and Figure 54 depict the CSR (overlap rate) and its spread using box whisker, respectively.

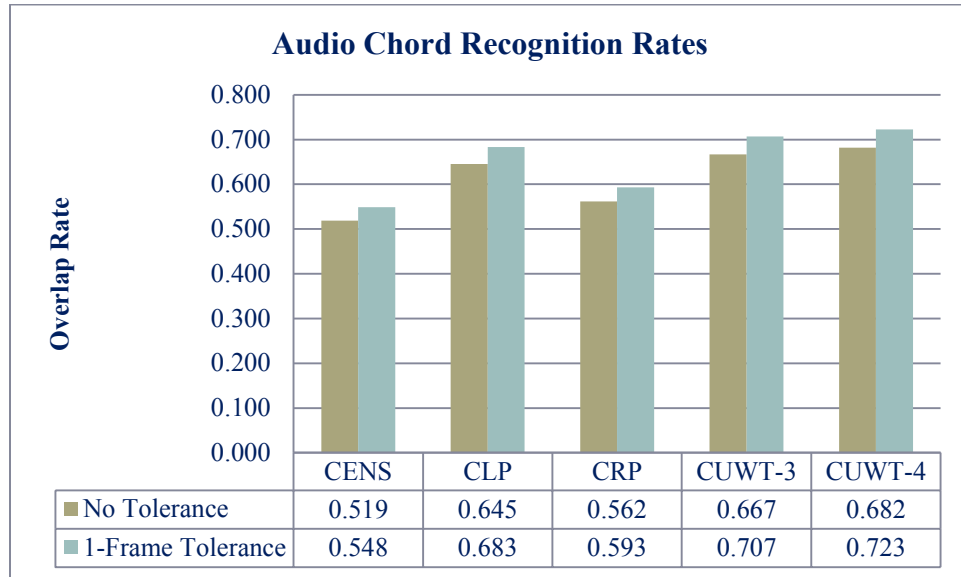


Figure 53: Chord recognition rates

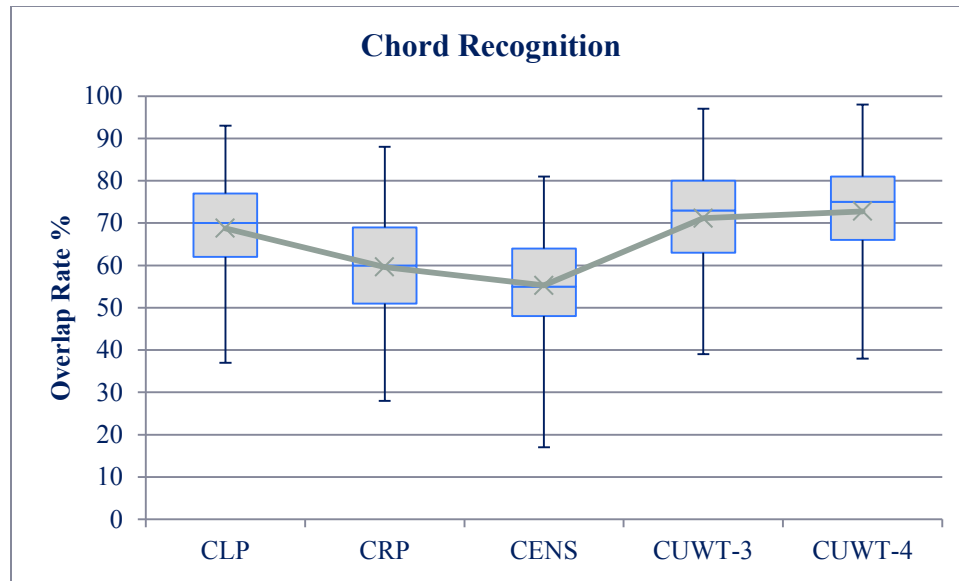


Figure 54: Chord recognition overlap rate (box and whisker)

Two observations can be made from Figure 53 and Figure 54.

- Proper alignment between the audio piece and ground truth is critical. Approximately four percent of performance gain is observed when tolerance of one frame (0.1 second) is given.
- Similar to what we observed for the key finding task, CUWT-3 and CUWT-4 remain to be superior chromagrams than the other three for the audio chord finding task.
- We see that the three chroma features (CLP, CENS, and CRP) produced drastically different results which are consistent with the experimental results described in (Müller & Ewert, 2011) with the exception that the CLP outperforms CRP significantly in our experiment.

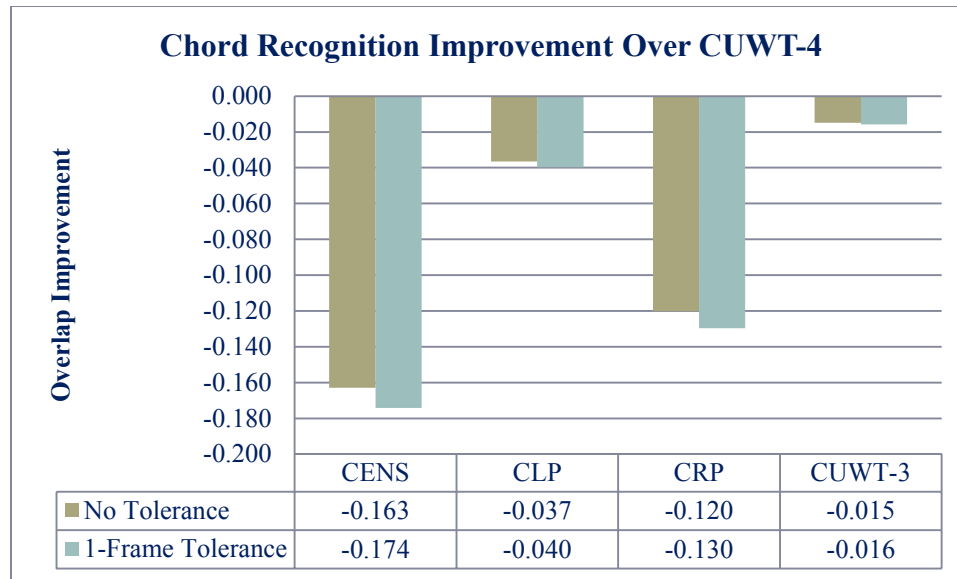


Figure 55: Chord recognition improvement over CUWT-4

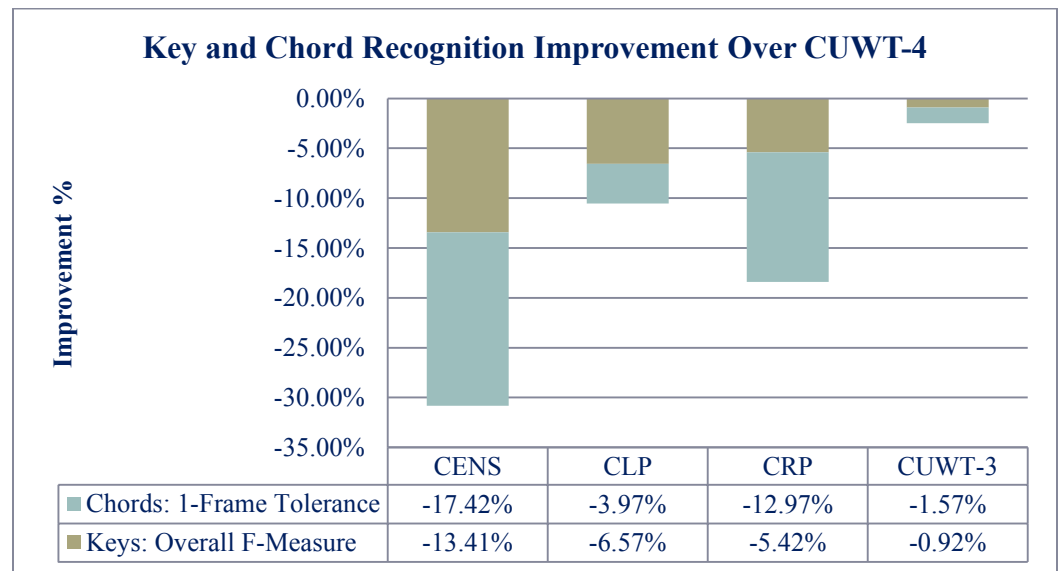


Figure 56: Combined improvement over CUWT-4

Using the CUWT-4 chroma feature as the benchmark, Figure 55 depicts the performance gain (or loss) of the other four chromagrams for the chord finding task. Using the same benchmark, Figure 56 shows the combination of performance gains (or losses) of key and chord recognition. From the two figures, we state the following observations.

- The CUWT-4 chromagram outperforms the most commonly used CLP by about 4% in chord recognition. A simple CLP outperforms the CRP and CENS by about 9% to 13%. Therefore, the selection of a chromagram has a high impact on the chord recognition rate.
- If we combine the F-Measure from the key finding task and the chord recognition rate (CSR or AOS), we notice a gap of more than 30% between the top and bottom performers.

To see the effect of using key knowledge for chord recognition, we also show our chord recognition rates without the use of extracted keys in Figure 57 where the simple peak-picking algorithm is applied as described in Table 13 but extending the templates to cover aug, dim and sus chord types. We see that using extracted keys improve the performance of a simple template-based chord recognizer at least 20% on all chromagrams. The step to correct sporadic chord assignments, described at the end of Section 3.4.4, accounts for roughly a 1% improvement across all features listed in Figure 57 which implies that the overall framework produced reasonable chord segmentation.

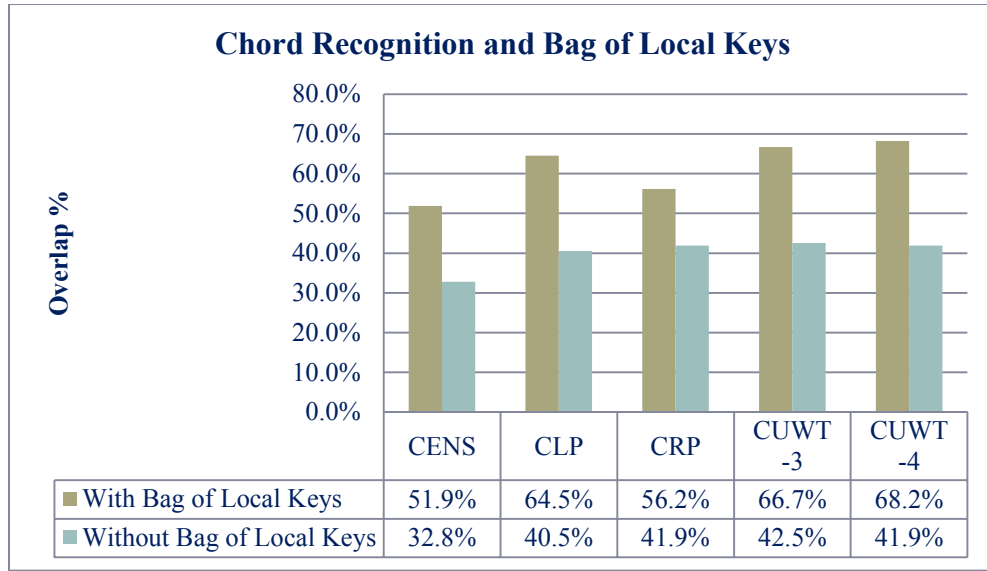


Figure 57: Effect of bag of local keys on chord recognition

4.4 Performance Comparison

While reviewing existing methods in Sections 2.3.2 (Summary of Previous Work) and 2.3.3 (Recent Work After 2008), we deliberately omitted reporting chord recognition rates in terms of Chord Symbol Recall (or Average Overlap Score) because of many differences in experimental settings. We group these differences into three broad categories – data set used for testing, performance scoring mechanism, and training data employed for supervised methods – and document them to provide a basis for performance comparison. Since we use 175 Beatles’ songs as our test data, we select methods in the literature with similar test dataset and experimental settings as ours so that a fair and objective comparison can be achieved. Furthermore, since the manual transcription of the complete Beatles’ collection was released by Harte (2010) in late

2007, all methods that we select for comparison were proposed after 2008 which is consistent with our detailed review in Section 2.3.3. The second criteria that we consider is the type of methodology – supervised, unsupervised, and knowledge based – so that a well-balanced representation among different technical approaches can be achieved. Moreover, as the last criteria, we select top performers from each type of methodology. Table 20 summarizes the comparison.

Table 20: Performance comparison of similar work published after 2008

Methodology	Training data (Y or N)	Test data	Recognition (chord type)	Detect “no” chord	Discard audio segment	Chord symbol recall
Rocher, Robine, Hanna, and Oudre	N	174 Beatles	maj, min	N	> 0	74.9%
Oudre, Fevotte, and Grenier	N	180 Beatles	maj, min	Y	0	72.4%
Pauwels, Martens, and	N	174 Beatles +	maj, min,	N	22.6%	78.4%

Peeters	binary chord template		18 Queen + 18 Zweieck	dim, aug			
Ni, Mcvigar, Stantos-Rodriguez, and De Bie	Beat tracking + Loudness based treble and bass chroma + HMM	Y	179 Beatles + 20 Queen + 18 Zweieck	maj, min, maj/3, maj/5, maj6, maj7, min7, 7, Dim, aug	Y	?	83.0%
Hu & Saul (2012)	Latent Dirichlet Allocation (LDA) for both symbolic and audio data; use Mauch' NNLS chroma features; audio data is synthesized from MIDI	N	136 Beatles	maj, min	N	?	49.1%
de Haas, Magalhaes, and Wiering	Knowledge-based tonal harmony model; Use Mauch's beat-synchronized NNLS chroma; Use K-S key profiles for key finding and involve dynamic programming	N	179 Beatles + 20 Queen + 18 Zweieck	maj, min, 7	N	> 0	74.1%
Wang & Wechsler	Wavelet based chromagram + bag of local keys + template-based chord matching	N	175 Beatles	maj, min, aug, dim, sus	Y	0	72.3%

It is clear that the supervised HMM-based machine learning method proposed by Ni et al. outperforms all other unsupervised or knowledge-based systems in the literature. Their method not only produces the highest chord recognition rate by a relatively wide margin, but the chord vocabulary – the number of chord types – they aim to recognize far exceeds all other methods. However, as emphasized by de Hass et al. (2012), in the 2011 edition of MIREX's chord estimation task, the recognition rates among participants are

between 12.6% and 82.9% while a deliberately overfitted result yields a CSR of 97.6%. Due to the scarcity of labeled training data, the majority of the supervised approaches are trained from the available 217 musical pieces (leave-one-out cross validation by Ni et al.), it is unclear how much of the trained systems and other supervised approaches have been overfitted by the said data set. The scarcity of training data is also reported in (Chai, 2005; Rhodes, et al., 2007; Pauwels, et al., 2011).

On the other hand, most of the unsupervised and knowledge-based systems appear to perform at about the same level, with the exception of the LDA-based method. The knowledge-based approach proposed by Pauwels et al. seems to be leading the pack; however, they discarded 22.6% of the audio segments that do not fit into the chord vocabulary and it is unclear how much such experimental setting affects the recognition rates. Furthermore, as indicated by de Hass (2012) as well as our own discovery discussed in Section 4.3.2, many different “re-mastered” versions of the Beatles are circulating on the market which might be slightly different from the version used in Harte’s annotated ground truth. These factors, along with different target chord vocabulary and test dataset employed in the proposed methods, we believe that our approach performs at least at, or outperforms, the other unsupervised or knowledge-based systems proposed in the literature.

4.5 Tonal Harmony and Machines

Thus far, we have discussed, mostly from the technical aspect, existing literature and our proposed methods in using machines to understand music in terms of its tonal and harmonic content. Given the experimental results that we surveyed, we believe that there is no clear winner in extracting the three fundamental elements – pitch, chords, and keys – using the three computing paradigms – supervised, unsupervised, and knowledge-based – that we commonly find from the literature. This is a perfect juncture for us to examine the merit of each paradigm in the context of music, and specifically, the extraction of tonal harmony using machines.

Supervised learning mimics the way, in certain aspects, how students analyze tones and harmony with feedback from teachers with the “correct” analyses in their learning process. However, as indicated by Bharucha (1991, p. 85), “one can demonstrate in a psychological laboratory that people without formal musical training in harmony analysis, are capable of making judgments about chords and their relationships.” He further stated that “this implicit or tacit knowledge of chords must have been obtained through passive perceptual exposure without feedback.” Passive exposure, in this case, means that there is no explicit training involved nor guidance from a supervisor for correct labeling of tonal and harmonic analysis. Clarke (2005, p. 29) further stated that “suitably enculturated listeners can make systematic judgments about tonal structure in music (expressed, for instance, in terms of the perceived completeness or stability of a

sequence) without any experience of ‘supervised learning’ or formal music instruction (Krumhansl 1990).”

The supervised approach for analysis of tonal harmony typically includes an HMM or N-gram as we discussed in Sections 2.3.2 and 2.3.3. However, components such as the Markov process or N-gram are only capable of capturing the local structure of music but not the large-scale structure at a higher level (Lewis, 1991). He further indicated that “the probability distribution function is not an economical representation – the probability distribution function must represent all possible structures, including those which are not desired ...” Such difficulties are evident in the dynamic Bayesian network (DBN) system we described in Section 2.3.3.

Rule-based approaches address the issue of “short-sightedness,” described earlier, in a sense that they can describe both large- and small-scale structure of a target music piece; however, it is difficult for such an approach to reconcile multiple parallel contexts such as meter, rhythm, and tonal harmony simultaneously (Todd & Loy, 1991, p. 29). Lewis (1991) further stated that “rules must be weakened or modified to handle ambiguity and “fuzzy” structure, properties which are characteristic of most forms of music.”

The availability of rules that adequately formalizes harmony progression of rock music might be another area of concern for rule-based approaches. Many music styles, including popular and rock, have certain patterns of motion which occur more often than others. However, these patterns of progression might not coincide with the common-

practice music (European art music from 18th to 19th centuries) whose basic principles of harmony have been studied extensively for the past hundred years. Though such well-studied harmonic successions are found in rock, as described by Stephenson (2002), these successions are “in the statistical minority” and “when they do occur, their rhythmic deployment within the phrase structure is usually not the same as that associated with common-practice music.” The harmonic patterns of a corpus of 100 representative rock songs, chosen from Rolling Stone magazine’s 20 top-ranked songs for each decade from the 1950s through the 1990s, are analyzed by de Clercq and Temperley (2011). They reported that “strong asymmetries of root motion found in common-practice music are notably absent in rock,” and “perhaps rock harmony is guided by strong and restrictive principles that have not yet been observed.” They further suggested that “a more ‘data-driven’ approach to rock harmony may be desirable, an approach in which the music is allowed to speak for itself.”

The unsupervised approach typically involves using a Bayesian-based probabilistic framework which certainly has its fair share of criticism regarding the violation of tonal expectation. Since tonal expectation, as described by (Todd & Loy, 1991, p. 40), generally explains “judgments about what pitch or chord should follow after the presentation of context pitches or chords,” is not addressed in this approach, tonal and harmonic labels are purely determined based on probability maximization. Therefore, a pure unsupervised approach has a higher potential for producing unrealistic labels than those of rule-based and supervised methods.

Without a doubt, the humans' auditory system and brain combined is the most powerful signal processing tool to analyze tonal harmony; the two systems working together allow us to enjoy music. By the same token, we believe that injecting a rule-based module of tonal expectancy into an unsupervised framework has the highest potential of allowing machines to extract and recognize tonality and harmony from raw audio signals.

Chapter 5 Applications and Extensions

Music segmentation is the process of partitioning the target music signals into multiple sections so that each section is homogeneous within its boundary but distinct from its neighboring sections; in musicology, we call it form analysis. It usually serves as an intermediate step to solve a larger problem such as content-based information retrieval. In computer vision, an extracted image segment can be used as a query to retrieve the content of similar nature. For popular music, a short “catchy” melody or text, which typically resides in a verse or chorus section, can be used as a query to retrieve the popular song. However, there are a few notable exceptions due to the inherent differences in the format of audio and image data and what they represent. First, music signals are one dimensional time series so the boundaries of a segment can completely be represented by two time points. Second, for western popular music, some segments are expected to repeat with certain order. Third, music is created to be pleasant to our ears so it follows certain “rules” to meet our expectations formed by previous listening experience. Methods employed for music segmentation can be categorized into repetition-, novelty-, and homogeneity-based, as described by (Paulus, et al., 2010). A theme that connects these methods is a self-similarity matrix (SSM) which was first proposed by (Foote, 1999) for music visualization and subsequently used by many researchers for segmentation (Jensen, 2007). In (Jensen, 2007), timbre, chroma, and

rhythm were used to produce SSMs in which a shortest path algorithm was employed to find the segmentation points; similarly in (Paulus, et al., 2010), using the three features, a probabilistic fitness function was introduced. Most recently in (Chen & Li, 2011), chroma and Mel-frequency cepstral coefficients (MFCCs) were used as features for clustering and the results from the two-level clustering were combined to produce better segmentation results.

In traditional musical form analysis on common period music, cadence patterns and key schemes are often employed as cues, but their usages are not strictly followed in popular music. Therefore, four other cues are used in rock music to signal the beginning of a new segment: text, instrumentation, rhythm, and harmony, as proposed in (Swain, 2002). An example of text cue could be the arrival of the title line; the instrumentation cue could be the addition of the guitar or background vocals. These two cues are not in the scope of this chapter. In our work, we propose to use keys and harmony (chords) to produce a multi-dimensional harmonic rhythm as the segmentation cue. Harmonic rhythm is delineated by (Swain, 2002) covering six dimensions: texture, phenomenal, bass pitch, root, density, and function. For our case, except bass pitch, the other five dimensions can be completely created from local keys and chords that correspond to the three rock cues (keys, rhythm, and harmony) described earlier. Our approach for segmentation is novel since we extract and separate the harmonic content into five dimensions of harmonic rhythm as the segmentation cue while most existing work use the whole chromagram for music structure analysis.

Based on the above overview, we see that there are three types of information to be extracted from the audio signals: local keys, chords, and segments. We have successfully extracted two elements – using undecimated wavelet transform on the audio signals, an infinite Gaussian mixture to extract a bag of local keys, and template-based chord recognition mechanism – from the Beatles’ 12 albums of 175 songs. We are currently combining the local keys and chords to create harmonic rhythm on a frame-by-frame basis to be used by the third component for music segmentation.

Figure 58 depicts the high-level components and flow of our system. After performing a wavelet transform on the audio signals to extract a chromagram, we extract a bag of local keys and subsequently a time series of chords. The extracted chords are then used to transform the bag of keys into a time series. Given the two time series, a multi-dimensional harmonic rhythm is formed to facilitate segmentation which is casted as a change detection problem. The last step is to use the segmentation information to refine chords. We describe each component in detail in the following subsections.

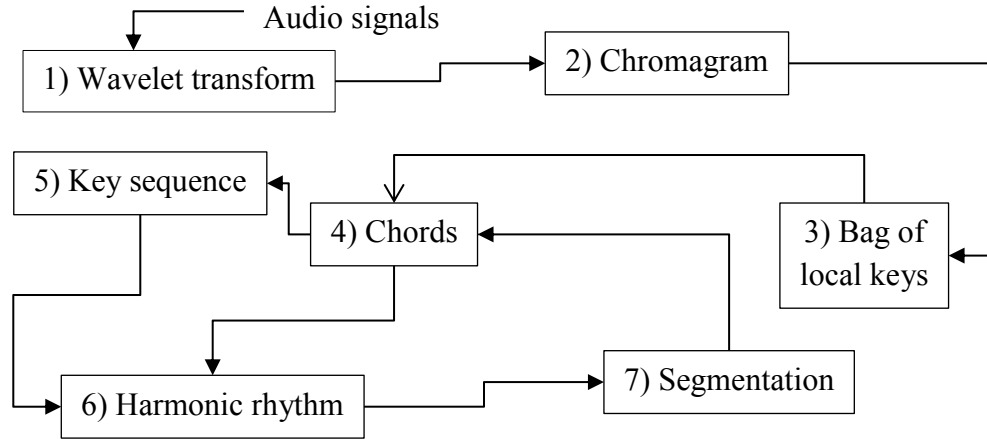


Figure 58: Music segmentation through harmonic rhythm

As described in the Introduction section, we propose to build a multi-dimensioned harmonic rhythm as the segmentation cue. Table 21 associates three elements – wavelet-based chromagram, extracted local keys, and chords – to the five dimensions of the harmonic rhythm. Suggested by (Stephenson, 2002), a key modulation (change of key) is the most obvious signal of a new segment and its cue, represented by the function dimension of the harmonic rhythm, is extracted by the local key estimator described in Section 3.2. The phenomenal, root, and function dimensions correspond to the harmonic aspect of a music piece; specifically, the chord, chord progression, and the degree of the chord in the diatonic scale, respectively. The original definition of the texture dimension is the fastest rhythm played (such as violin or piano) in a music piece within a measure, but we use the chromagram extracted from wavelet approximations as the texture dimension. Since extracted keys and chords can never be 100% accurate, we use the density dimension to express the percentage of texture information represented by the

phenomenal dimension that we extracted. Therefore, if the density is high, we are more confident about the cue within its time window.

Table 21: Segmentation cues

Dimension	Segmentation Cue
Texture	Wavelet-based chromagram
Phenomenal	Chords
Root/quality	Root progression
Density	% of energy in texture that articulates the phenomenal dimension
Function	The triad's position (roman numeral) in a key

In rock form analysis, (Stephenson, 2002) states that there is no need to wait for the complete unfolding of a harmonic pattern to see if it differs significantly from what has come before. It coincides with our listening experience of popular music, i.e., without formal music training, most listeners are capable of sensing a new “segment” coming up for a song that they listen to for the first time. This is the main idea of our proposed segmentation process using machines, i.e., to mimic the humans’ perception of change based on the five cues from harmonic rhythm. Other than using the cue of local key changes, other dimensions will be inspected from the perspectives of speed and independency (Swain, 2002); both are related to tension and resolution. Speed is one of

the fundamental ways to create tension: the faster the motion, the greater the tension. As the tension builds up, listeners expect to hear a resolution which signifies a change; though such a change alone does not necessarily warranty the beginning of a new segment. Specifically for the speed perspective, we will examine the speed of change on phenomenal and root dimensions of the harmonic rhythm to detect change. Independency among dimensions of harmonic rhythm also creates tension: the more divergent they are, the more tension they build; the resolution of such tension is the arrival of convergence. Since speed is the best indicator of (in)dependence among salient dimensions, we will detect the change points of divergence and convergence by examining the root, phenomenal, and density dimensions. Therefore, the task of music segmentation can be approached by detecting three changes – key, speed, and independence – from the harmonic rhythm.

The sequential probability ratio test (SPRT) and cumulative sum (CUSUM), originally developed for quality control purposes in manufacturing, are the first two approaches for change detection on sequential data; many methods were derived from them (Basseville & Nikiforov, 1993). These methods are statistically parametric and require estimation of likelihood. However, it is impossible to assume any underlying distribution in harmonic rhythm, so a nonparametric method is in order. Recently, a non-parametric, martingale based change detection method was proposed in (Ho & Wechsler, 2010) by examining the strangeness of a newly arrived data point to see if the assumption of exchangeability is violated, which signals a change in the data stream. For data points in the harmonic rhythm that fall inside a segment, we can safely assume that they are

generated by the same latent variable and therefore exchangeable. Different from the online streaming data, we have the complete harmonic rhythm to help determine the appropriate strangeness measure for the speed and dependency cues of the target music. Based on the strangeness of the cues in sequence of the harmonic rhythm, we can detect the segmentation boundaries.

We have discussed all components and steps in Figure 58 except the processes from components 7 to 4, a part of an estimation refinement loop consisting of components 4, 5, 6, and 7. This last step uses the segmentation information to fine tune the time series of chords estimated in step 4.

Chapter 6 Conclusions and Future Work

In this chapter, we summarize the work that we have performed and highlight the contributions to the field of Music Information Retrieval. Potential future work is also discussed.

6.1 Summary

With the end goal of devising a simple, but not simpler, mechanism to extract tonality and harmony from real audio music of WAV format, we started our journey from a much simpler and clearer format of the symbolic MIDI music. Since MIDI is designed to instruct computers to communicate and play music, musical notes can be easily extracted and the two tasks (tonality and harmony recognition) are completed by modeling the target music using an infinite Gaussian mixture. Since there is no ground truth available for MIDI music, manually transcribed key and chord from the real audio WAV recordings is used as a validity check for the symbolic domain. We obtain reasonable good results for both the key and chord recognition tasks. Using a bag-of-notes modeling experience from the symbolic domain, we proceed to analyze the WAV track for real audio CD albums.

WAV audio, unlike MIDI, requires a Fourier-like transform to convert the signals from the time domain to the frequency domain for the two tasks. Due to inherent “noisy” characteristics of the audio recording – such as the attack transients and higher harmonics that do not contribute positively to the recognition of tonality and harmony content of the target music piece, we use a wavelet transform on the raw WAV data to obtain a smoother and period-regularized approximation to the original signals. A best candidate approximation is chosen based on entropy-based and similarity-based criteria. The chosen approximation, still in time representation, is transformed into frequency representation using a Constant-Q transform where a series of 12-dimensioned Pitch Class Profiles, or chromagram, is generated for extracting local keys.

The processing paths of symbolic and real audio data, i.e., MIDI and WAV, cross at the adoption of infinite Gaussian mixture for extracting a bag of local keys from a bag of frames. Using the Beatles’ 175 songs in our experiments, we observe that the wavelet approximated signals provided at least a 5% improvement on the F-measure over other chromagrams on extraction of local keys. Using the obtained local keys, the energy levels in each chromagram is adjusted by applying the Krumhansl & Kessler profiles to promote diatonic pitches to find the most suitable chords for harmony extraction. Again, using the 175 Beatles’ songs, with no frames discarded, we achieved a 72.3% recall (correct overlap) rate on extracting six types of chords – major, minor, augmented, diminished, suspended, and N (none) – which rivals results from the state-of-the-art. We also observed that our wavelet transformed chromagram outperforms others by at least 4% in terms of chord recognition.

6.2 Contributions

The main argument for applying unsupervised machine learning paradigms for harmony analysis on audio signals follows the principle of Einstein’s – “As simple as possible, but not simpler” – and David Wheeler’s corollary to Butler Lampson’s quote – “..., except for the problem of too many layers of indirection.” From experimental results, we show that our approach – a much simpler one compared to most of the existing methods – performs just as well or outperforms many of the much more complex models for harmony analysis without using any training data. We make four contributions to the music signal processing and music information processing communities:

1. We have shown that using undecimated wavelet transform on the raw audio signals improves the quality of the pitch class profiles.
2. We have demonstrated that an infinite Gaussian mixture can be used to efficiently generate a bag of local keys for a music piece.
3. We have ascertained that the combination of well-known tonal profiles and a bag of local keys can be used to adjust the pitch class profiles for harmony analysis.
4. We have shown that an unsupervised chord recognition system – without any training data or other musical elements – can perform as well, if not exceed, many of the supervised counterparts.

6.3 Future Work

We see that there are three lines of future work. First, we can adjust the framework to replace the one-way interaction (a bag of local keys first, then frame-by-frame chords) with two-way estimation so that chord information can be used to transform the bag of local keys into a time series of local keys which can in turn improve the chord recognition task iteratively. Second, build the harmonic rhythm as a five-dimensioned segmentation cues for structural analysis. For the first two lines of work, we have elaborated on them in detail in Section 5. Third, extend the use of the infinite Gaussian mixture to develop a new global descriptor using a bag of spectral frames as input as briefly described in Section 3.4.

Bibliography

Alten, S. R., 2011. *Recording and Producing Audio for Media*. 1st ed. Boston: Cengage Learning PTR.

Anglade, A., Ramirez, R. & Dixon, S., 2009. *Genre classification using harmony rules from automatic chord transcriptions*. Kobe, Japan, 10th International Society for Music Information Retrieval Conference.

Antoniak, C., 1974. Mixtures of Dirichlet processes with applications to nonparametric problems. *Annals of Statistics*, Volume 2, pp. 1152-1174.

Bader, R., 2013. *Nonlinearities and Synchronization in Musical Acoustics and Music Psychology*. Berlin; New York: Springer.

Basseville, M. & Nikiforov, I., 1993. *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River: Prentice Hall.

Bello, J. & Pickens, J., 2005. *A robust mid-level representation for harmonic content in music signals*. London, U.K., 6th International Conference on Music Information Retrieval.

Benson, D., 2007. Music: A Mathematical Offering. In: New York: Cambridge University Press, pp. 162-165.

Bharucha, J. J., 1991. Pitch, Harmony, and Neural Nets: A psychological Perspective. In: P. Todd & D. Loy, eds. *Music and Connectionism*. Cambridge: Massachusetts Institute of Technology, pp. 84-99.

Bharucha, J. & Todd, P., 1991. Modeling the Perception of Tonal Structure with Neural Nets. In: P. Todd & D. Loy, eds. *Music and Connectionism*. Cambridge: Massachusetts Institute of Technology, pp. 128-137.

Blackwell, D. & MacQueen, J., 1973. Ferguson distributions via Polya urn schemes. *Annals of Statistics*, Volume 1, pp. 353-355.

- Brown, J. C., 1991. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1).
- Cavaliere, S. & Piccialli, A., 1997. Granular synthesis of musical signals. In: C. Roads, S. Pope, A. Piccialli & G. de Poli, eds. *Musical Signal Processing*. Lisse: Swets & Zeitlinger B. V., pp. 155-186.
- Chai, W., 2005. *Automated Analysis of Musical Structure*. Doctoral dissertation in Media Arts and Sciences, School of Architecture and Planning: Massachusetts Institute of Technology.
- Chai, W. & Vercoe, B., 2005. *Detection of key change in classical piano music*. London, U.K., 6th International Conference on Music Information Retrieval.
- Cheng, H.-T. et al., 2008. *Automatic chord recognition for music classification and retrieval*. Hannover, Germany, 2008 IEEE International Conference on Multimedia and Expo.
- Chen, R. & Li, M., 2011. *Music structural segmentation by combining harmonic and timbral information*. Miami, Florida, 12th International Society for Music Information Retrieval Conference.
- Cho, T. & Bello, J., 2011. *A feature smoothing method for chord recognition using recurrence plots*. Miami, Florida, 12th International Society for Music Information Retrieval Conference.
- Chuan, C.-H. & Chew, E., 2005. *Polyphonic audio key finding using the spiral array CEG algorithm*. Amsterdam, The Netherlands, IEEE International Conference on Multimedia & Expo.
- Clarke, E. F., 2005. *Ways of Listening: An Ecological Approach to the Perception of Musical Meaning*. New York: Oxford University Press.
- de Clercq, T. & Temperley, D., 2011. A corpus analysis of rock harmony. *Popular Music*, Volume 30/1, pp. 47-70.
- de Haas, W. B., 2012. *Music Information Retrieval Based on Tonal Harmony*. Doctoral dissertation in Information and Computing Sciences: Universiteit Utrecht, The Netherlands.

- de Haas, W. B., Magalhães, J. P. & Wiering, F., 2012. *Improving audio chord transcription by exploiting harmonic and metric knowledge*. Porto, Portugal, 13th International Society for Music Information Retrieval Conference.
- Ferguson, T., 1973. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, Volume 1/2, pp. 209-230.
- Foote, J., 1999. *Visualizing music and audio using self-similarity*. Orlando, Florida, ACM Multimedia.
- Fujishima, T., 1999. *Real time chord recognition of musical sound: a system using Common Lisp Music*. Beijing, China, International Computer Music Association .
- Gómez, E., 2006. Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, 18(3), pp. 294-304.
- Harte, C., 2010. *Towards automatic extraction of harmony information from music signals*. Doctoral dissertation in Electronic Engineering: University of London, Queen Mary.
- Harte, C. & Sandler, M., 2005. *Automatic chord identification using a quantized chromagram*. Barcelona, Spain, 118th Audio Engineering Society Convention.
- Harte, C., Sandler, M., Abdallah, S. & Gómez, E., 2005. *Symbolic representation of musical chords: A proposed syntax for text annotations*. London, U.K., 6th International Conference on Music Information Retrieval.
- Hewitt, M., 2010. *Harmony for Computer Musicians*. 1st ed. Boston: Cengage Learning PTR.
- Ho, S.-S. & Wechsler, H., 2010. A martingale framework for detecting changes in data streams by testing exchangeability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), pp. 2113-2127.
- Hu, D. J., 2012. *Probabilistic Topic Models for Automatic Harmonic Analysis of Music*. Doctoral dissertation in Computer Science: University of California, San Diego.
- Hu, D. & Saul, L., 2009. *A Probabilistic Topic Model for Unsupervised Learning of Musical Key-Profiles*. Kobe, Japan, 10th International Society for Music Information Retrieval Conference.
- Itoyama, K., Ogata, T. & H.G., O., 2012. *Automatic chord recognition based on probabilistic integration of acoustic features, bass sounds, and chord transition*. Dalian,

China, 25th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems.

Izmirli, O., 2007. *Localized key-finding from audio using non-negative matrix factorization for segmentation*. Vienna, Austria, 8th International Conference on Music Information Retrieval.

Jensen, K., 2007. Multiple Scale Music Segmentation Using Rhythm, Timbre, and Harmony. *EURASIP Journal on Advances in Signal Processing*, 2007(Article ID 73205), pp. 1-11.

Khadkevich, M. & Omologo, M., 2009. *Use of hidden Markov models and factored language models for automatic chord recognition*. Kobe, Japan, 10th International Society for Music Information Retrieval Conference.

Koelsch, S. & Siebel, W., 2005. Towards a neural basis of music perception. *TRENDS in Cognitive Sciences*, 9(12), pp. 578-584.

Krumhansl, C., 1990. *Cognitive Foundations of Musical Pitch*. 1st ed. New York: Oxford University Press.

Krumhansl, C. & Kessler, E. J., 1982. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, 89(4), pp. 334-368.

Lee, K. & Slaney, M., 2008. 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), pp. 291-301.

Leman, M., 1991. The Ontogenesis of Tonal Semantics: Results of a Computer Study. In: P. Todd & D. Loy, eds. *Music and Connectionism*. Cambridge: Massachusetts Institute of Technology, pp. 100-127.

Lerch, A., 2012. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. 1st ed. s.l.:Wiley-IEEE Press.

Lerdahl, F., 2001. *Tonal Pitch Space*. New York: Oxford University Press.

Lewis, J. P., 1991. Creation by Refinement and the Problem of Algorithmic Music Composition. In: P. Todd & D. Loy, eds. *Music and Connectionism*. Cambridge: Massachusetts Institute of Technology, pp. 212-228.

- Lin, C.-J., Lee, C.-L. & Peng, C.-C., 2011. *Chord recognition using neural networks based on particle swarm optimization*. San Jose, California, U.S.A., 2011 International Joint Conference on Neural Networks.
- Li, T., Ogihara, M. & Tzanetakis, G., 2011. *Music Data Mining*. Boca Raton: CRC Press.
- Lowry, T., 1988. *Complete Beatles*. 1st ed. Milwaukee: Hal Leonard Publishing Corporation.
- Loy, G., 2006. *Musimathics: The Mathematical Foundations of Music (Volume 1)*. Cambridge: MIT Press.
- Loy, G., 2007. *Musimathics: The Mathematical Foundations of Music (Volume 2)*. Cambridge: MIT Press.
- Mauch, M. & Dixon, S., 2010. Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6), pp. 1280-1289 .
- Müller, M. & Ewert, S., 2011. *Chroma toolbox: MATLAB implementations for extracting variants of chroma-based audio features*. Miami, Florida, U.S.A., 12th International Society for Music Information Retrieval Conference.
- Neal, R., 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2), pp. 249-265.
- Ni, Y., McVicar, M., Santos-Rodriguez, R. & De Bie, T., 2012. An end-to-end machine learning system for harmonic analysis of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6), pp. 1771-1783.
- Noland, K. & Sandler, M., 2009. Influences of signal processing, tone profiles, and chord progressions on a model for estimating the musical key from audio. *Computer Music Journal*, 33(1), pp. 42-56.
- Oudre, L., Févotte, C. & Grenier, Y., 2011. Probabilistic template-based chord recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8), pp. 2249-2259.
- Pachet, F. & Roy, P., 2008. *Hit song science is not yet a science*. Philadelphia, Pennsylvania, U.S.A , 9th International Conference on Music Information Retrieval.

Païement, J.-F., Eck, D. & Bengio, S., 2005. *A probabilistic model for chord progressions*. London, U.K., 6th International Conference on Music Information Retrieval.

Papadopoulos, H. & Peeters, G., 2012. Local key estimation from an audio signal relying on harmonic and metrical structures. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(4), pp. 1297-1312.

Paulus, J., Müller, M. & Klapuri, A., 2010. *Audio-based music structure analysis*. Utrecht, Netherlands, 11th International Society for Music Information Retrieval Conference.

Pauwels, J., Martens, J.-P. & Leman, M., 2011. *Improving the key extraction performance of a simultaneous local key and chord estimation system*. Barcelona, Spain, 2011 IEEE International Conference on Multimedia and Expo.

Pauws, S., 2004. *Musical key extraction from audio*. Barcelona, Spain, 5th International Conference on Music Information Retrieval.

Peeters, G., 2006. *Musical key estimation of audio signal based on hidden Markov modeling of chroma vectors*. Montreal, Canada, 9th International Conference on Digital Audio Effects.

Pollack, A. W., n.d. *Notes on ... Series*. [Online]
Available at: <http://www.icce.rug.nl/~soundscapes/DATABASES/AWP/awp-alphabet.shtml>
[Accessed 13 October 2011].

Raphael, C. & Stoddard, J., 2003. *Harmonic analysis with probabilistic graphical models*. Baltimore, Maryland, U.S.A., 4th International Conference on Music Information Retrieval.

Rasmussen, C., 2000. The Infinite Gaussian Mixture Model. In: S. Solla, T. Leen & K. Müller, eds. *In Advances in Neural Information Processing Systems 12*. Cambridge: MIT Press, pp. 554-560.

Rhodes, C., Lewis, D. & Mullensiefen, D., 2007. *Bayesian model Selection for Harmonic Labelling*. Berlin, Germany, First International Conference of the Society for Mathematics and Computation in Music.

Rocher, T., Robine, M., Hanna, P. & Oudre, L., 2010. *Concurrent estimation of chords and keys from audio*. Utrecht, Netherlands, 11th International Society for Music Information Retrieval Conference.

Roelleke, T., 2013. *Information Retrieval Models: Foundations and Relationships (Synthesis Lectures on Information Concepts, Retrieval, and Services)*. 1st ed. San Rafael: Morgan & Claypool Publishers.

Ryynanen, M. & Klapuri, A., 2008. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3), pp. 72-86.

Schinder, S. & Schwartz, A., 2008. *Icons of Rock: An Encyclopedia of the Legends Who Changed Music Forever*. Westport: Greenwood Press.

Sheh, A. & Ellis, D., 2003. *Chord segmentation and recognition using EM-trained hidden Markov models*. Baltimore, Maryland, U.S.A., 4th International Conference on Music Information Retrieval.

Sleator, D. & Temperley, D., 2001. *The Melisma Music Analyzer*. [Online] Available at: <http://www.link.cs.cmu.edu/music-analysis/>

Snoman, R., 2013. *Dance Music Manual: Tools, Toys, and Techniques*. 3rd ed. Burlington: Focal Press.

Stephenson, K., 2002. *What to Listen For in Rock: A Stylistic Analysis*. New Haven: Yale University Press.

Swain, J., 2002. *Harmonic Rhythm: Analysis and Interpretation*. New York: Oxford University Press.

Temperley, D., 2007. *Music and Probability*. Cambridge: MIT Press.

Todd, P. & Loy, D., 1991. *Music and Connectionism*. Cambridge: Massachusetts Institute of Technology.

Toiviainen, P. & Eerola, T., 2004. *MIDI Toolbox: MATLAB Tools for Music Research*, Jyväskylä, Finland: Department of Music of the University of Jyväskylä.

Ueda, Y. et al., 2010. *HMM-Based Approach for Automatic Chord Detection Using Refined Acoustic Features*. Dallas, Texas, U.S.A., 35th International Conference on Acoustics Speech and Signal Processing.

Varewyck, M., Pauwels, J. & Martens, J.-P., 2008. *A novel chroma representation of polyphonic music based on multiple pitch tracking techniques*. Vancouver, British Columbia, Canada, 16th ACM International Conference on Multimedia.

Wakefield, G., 1999. *Mathematical representation of joint time-chroma distributions*. Denver, Colorado, U.S.A., SPIE Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations IX.

Weil, J., Sikora, T., Durrieu, J.-L. & Richard, G., 2009. *Automatic generation of lead sheets from polyphonic music signals*. Kobe, Japan, 10th International Society for Music Information Retrieval Conference.

Weller, A., Ellis, D. & Jebara, T., 2009. *Structured prediction models for chord transcription of music audio*. Miami, Florida, U.S.A., 8th International Conference on Machine Learning and Applications .

West, M., Muller, P. & Escobar, M., 1994. Hierarchical priors and mixture models, with application in regression and density estimation. In: P. R. Freeman & A. F. M. Smith, eds. *Aspects of Uncertainty: A Tribute to D.V. Lindley*. Hoboken: Wiley, pp. 363-386.

Wood, F. & Black, M., 2008. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173(1), pp. 1-12.

Yang, Y.-H. & Chen, H. H., 2011. *Music Emotion Recognition*. Boca Raton: CRC Press.

Yan, R., 2007. *Base Wavelet Selection Criteria for Non-Stationary Vibration Analysis in Bearing Health Diagnosis*. Doctoral disseration in Mechanical Engineering: University of Massachusetts, Amherst.

Yoshii, K. & Goto, M., 2012. A nonparametric Bayesian multipitch analyzer based on infinite latent harmonic allocation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3), pp. 717-730.

Zhu, Y., Kankanhalli, M. & Gao, S., 2005. *Music key detection for musical audio*. Melbourne, Australia, 11th International Conference on Multi-Media Modelling.

Biography

Yun-Sheng Wang received his BS degree in Transportation Engineering and Management from Feng Chia University, Taiwan. He served two years in Taiwan's Marine Corp with distinguished service. He earned a MS degree in Civil Engineering from University of Virginia after his military service. While working full-time in the engineering and information technology fields, across private and public sectors, he completed a MS degree in Computer Science from George Mason University. In the IT field, he works as an enterprise application architect. He is moving toward doing research at the intersection of audio signal processing, Bayesian inference, and pattern recognition.