Automated Analysis of Musical Structure

by

Wei Chai

B.S. Computer Science, Peking University, China 1996
M.S. Computer Science, Peking University, China 1999
M.S. Media Arts and Sciences, MIT, 2001

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial
fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 2005

Author .....
Program in Media Arts and Sciences
August 5, 2005

Certified by .....
Barry L. Vercoe
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by.....
Andrew B. Lippman
Chairman, Departmental Committee on Graduate Students

Automated Analysis of Musical Structure

by Wei Chai

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on August 5, 2005,
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Abstract

Listening to music and perceiving its structure is a fairly easy task for humans, even for listeners without formal musical training. For example, we can notice changes of notes, chords and keys, though we might not be able to name them (*segmentation based on tonality and harmonic analysis*); we can parse a musical piece into phrases or sections (*segmentation based on recurrent structural analysis*); we can identify and memorize the main themes or the catchiest parts – hooks - of a piece (*summarization based on hook analysis*); we can detect the most informative musical parts for making certain judgments (*detection of salience for classification*). However, building computational models to mimic these processes is a hard problem. Furthermore, the amount of digital music that has been generated and stored has already become unfathomable. How to efficiently store and retrieve the digital content is an important real-world problem.

This dissertation presents our research on automatic music segmentation, summarization and classification using a framework combining music cognition, machine learning and signal processing. It will inquire scientifically into the nature of human perception of music, and offer a practical solution to difficult problems of machine intelligence for automatic musical content analysis and pattern discovery.

Specifically, for segmentation, an HMM-based approach will be used for key change and chord change detection; and a method for detecting the self-similarity property using approximate pattern matching will be presented for recurrent structural analysis. For summarization, we will investigate the locations where the catchiest parts of a musical piece normally appear and develop strategies for automatically generating music thumbnails based on this analysis. For musical salience detection, we will examine methods for weighting the importance of musical segments based on the confidence of classification. Two classification techniques and their definitions of confidence will be explored. The effectiveness of all our methods will be demonstrated by quantitative evaluations and/or human experiments on complex real-world musical stimuli.

Thesis supervisor: Barry L. Vercoe, D.M.A.
Title: Professor of Media Arts and Sciences

Thesis Committee

Thesis Advisor

Barry Vercoe
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis Reader

Tod Machover
Professor of Music and Media
Massachusetts Institute of Technology

Thesis Reader

Rosalind Picard
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Acknowledgements

I have been very lucky to work in the Music Mind and Machine Group of the Media Laboratory for the past six years. This allowed me to collaborate with many brilliant researchers and musicians. My period of graduate study at MIT has been one of the most challenging and memorable so far in my life. I am happy to have learned about many new technologies, new cultures, and especially the innovative ways people carry out research at MIT. This dissertation work was funded under MIT Media Laboratory Digital Life Consortium. I would like to thank everyone who has made my research fruitful and this dissertation possible.

I am indebted to my advisor, Professor Barry Vercoe. He is not only an excellent academic advisor, who always gave me his support to pursue my own interests and his valuable suggestions to inspire new ideas from me, but also a sophisticated mental advisor, who helped me a lot adapt to the culture completely new to me and plan my future career.

I would like to express my sincerest thanks to my committee members: Professor Roz Picard and Professor Tod Machover, for their thoughtful comments, criticism, and encouragement provided throughout the thesis writing process. Especially, it was my first class at MIT - Signal and Systems taught by Roz - that provided me with the fundamental concepts on audio signal processing and brought me into my research field.

Also, I am grateful for the encouragement, help and insight I have received from current and past members of the Music Mind and Machine Group – Victor Adan, Judy Brown, Ricardo Garcia, John Harrison, Tamara Hearn, Youngmoo Kim, Nyssim Lefford, Elizabeth Marzloff, Joe Pompei, Rebecca Reich, Connie Van Rheenen, Eric Scheirer, Paris Smaragdis, and Kristie Thompson. I have had the great fortune of working with these brilliant and talented people. I especially thank my officemate, Brian Whitman - one of the most insightful researchers in my field and the most warmhearted person who always gave me great help and suggestions.

I have been assisted and influenced by many other members of the Media Lab community. In particular, I thank Yuan Qi, who gave me his code for Pred-ARD-EP and many good suggestions, and Aggelos Bletsas, who had interesting discussions with me on probability.

Special thanks to my friends at MIT: Hong He, Wenchao Sheng, Yunpeng Yin, Rensheng Deng, Minggang She, who gave me endless help and made my life at MIT enjoyable.

My greatest debt of gratitude is owed to my family, for their love and support.

Table of Contents

List of Figures

Chapter 1 Introduction

Listening to music and perceiving its structure is a fairly easy task for humans, even for listeners without formal musical training. For example, we can notice changes of notes, chords and keys, though we might not be able to name them (*tonality and harmonic analysis*); we can parse a musical piece into phrases or sections (*recurrent structural analysis*); we can identify and memorize main themes or hooks of a piece (*summarization*); we can detect the most informative musical parts for making certain judgments (*detection of salience for classification*). However, building computational models to mimic this process is a hard problem. Furthermore, the amount of digital music that has been generated and stored has already become unfathomable. How to efficiently store and retrieve the digital content is an important real-world problem.

This dissertation presents our research on automatic music segmentation, summarization and classification using the framework combining music cognition, machine learning and signal processing. It will inquire scientifically into the nature of human perception of music, and offer a practical solution to difficult problems of machine intelligence for automatic musical content analysis and pattern discovery.

In particular, the computational models will automate the analysis of the following: What is the progression of chords and keys underlying the surface of notes? What is the recurrent structure of a piece? What are the repetitive properties of music at different levels, which are organized in a hierarchical way? What is the relation between the musical parts and the whole? Which parts are most "informative" for the listeners to make judgments? What are the most "representative" parts that make the piece unique or memorable?

Solutions to these problems should benefit intelligent music editing systems and music information retrieval systems for indexing, locating and searching for music. For example, consider the following scenarios: A system can segment a musical recording phrase-by-phrase or section-by-section and present the result for users to quickly locate the part they are interested in; A system can analyze the tonality, harmony and form of a musical piece for musical instruction; A system can generate a twenty-second thumbnail of a musical piece and present it to the customers for them to decide whether they would like to buy the whole piece; A system can identify the characteristics of an artist by "hearing" a collection of his works and comparing them to works by other artists for aesthetic analysis or copyright protection. These are some of the scenarios in which our proposed models can be employed.

The topics are also closely related to music understanding, human mental representations of music, musical memory, and the attentive listening process. Successful computational models to mimic the perception of musical structure will contribute to the study of music understanding and cognition.

First, music inherently contains large amounts of structure. Perception and analysis of structure is essential for understanding music. Some of the tasks addressed in this dissertation are very similar to the tasks in natural language understanding, where the semantic meaning of language is supported by a hierarchically organized structure based on words, phrases, sentences, paragraphs; and some key points typically need to be emphasized by being repeated and put at some structurally accentuated locations.

Second, it is still unclear why some music or part of music is more memorable than another. It should not be coincident that almost all genres of music in the world have some kind of repetitions. One explanation is the close relationship between poetry and music: music is a way of adding more dimensions to poems though variations of pitches and time, while poems have repetitions. But still it does not explain why repetition is so important for these forms of art. Our hypothesis is that repetition adds more redundancy of information, which can reduce the

processing of human brain and relieve some mental resources for other aesthetic purposes. That probably is in part what allows music to make humans more emotionally involved and immersed.

Third, this dissertation is, to some extent, all about the relation between part and whole. It will talk about various kinds of segmentations, based on key changes, chord changes, repetitions, representativeness of phrases, and categorical salience of phrases, etc., since only when we can chunk the whole into parts and look closely into their relations, we can really understand how music works.

Fourth, similarity is an important concept in cognition. "We live by comparisons, similarities and dissimilarities, equivalences and differences." (R. D. Laing) Making judgment of difference or similarity by comparison is our primary way of learning. This dissertation is also related to musical similarity. Various models of musical similarity have been employed for different purposes, including geometric models, alignment-based models, statistical models, and multidimensional scaling. This is reasonable, since the famous Ugly Duckling Theorem reveals that "*there is no problem-independent or privileged or 'best' set of features or feature attributes; even the apparently simple notion of similarity between patterns is fundamentally based on implicit assumptions about the problem domain.*" (Duda, 2001) The human mind is quite good at combining different models for comparing things.

1.1 Contributions

The main contribution of this dissertation is two-fold: a set of algorithms and techniques for real-world problems in building intelligent music systems; findings and hints we can obtain for the study of human perception of musical structure and meaning.

This dissertation proposes a novel framework for music segmentation. First, a Hidden Markov Model based method is employed for detecting key or chord changes as well as identifying keys or chords. This is different from most previous approaches that attempted to do key or chord detection without considering segmentation. Additionally, some but a limited amount of prior musical knowledge is incorporated in the system to solve the problem due to lack of enough training data.

Second, a Dynamic Time Warping based method is proposed for detecting the recurrent structure and self-similarity of music and parsing a piece into sections. This is probably the first attempt of building a system to give the overall formal structure of music from acoustic signals; previous research typically tried to find only the most repeated patterns. The ultimate goal of this research would be to derive the hierarchical structure of music, which is also addressed in the dissertation.

Comprehensive metrics for evaluating music segmentation are proposed, while most previous research had only one or two examples for demonstrating the promise of their methods rather than quantitative evaluations.

Besides segmentation, a novel method for music summarization based on the recurrent structural analysis is proposed. An online human experiment is conducted to set up the ground truth for music summarization. The results are used in this dissertation to develop strategies for summarization and can be used in the future for further investigation of the problem.

Finally, this dissertation proposes a new problem – musical salience for classification - and corresponding methods that detect the most "informative" part of music for making certain judgments. What "informative" means really depends on the tasks - listeners pay attention to different parts of music depending on what kind of information they want to obtain during the listening process. We explore how musical parts are weighted differently for different classification tasks and whether the weighting is consistent with human intuition.

In general, our approach has been applying psychoacoustic and music cognition principles as bases, and employing musical signal processing and machine learning techniques as front-end tools for developing representations and algorithms to mimic various aspects of human music listening process. We also focus on listeners without professional training. This implies that "real" musical signals will be the main stimuli in the structural analysis studies and only a limited amount of prior musical knowledge will be employed in the processing.

1.2 Overview and Organizations

This dissertation consists of four correlated components for automated analysis of musical structure from acoustic signals: tonality analysis, recurrent structural analysis, hook analysis and salience analysis, mainly for three types of applications - segmentation, summarization and classification. Only a limited amount of prior musical knowledge and patterns extracted from symbolic musical data (i.e., musical scores) will be employed to help build models - either statistical models such as Hidden Markov Models for key/chord detection and discriminative models for classification, or rule-based models such as approximate pattern matching for self-similarity analysis and structural accentuation for thumbnailing. Figure 1-1 shows an overview of the dissertation.

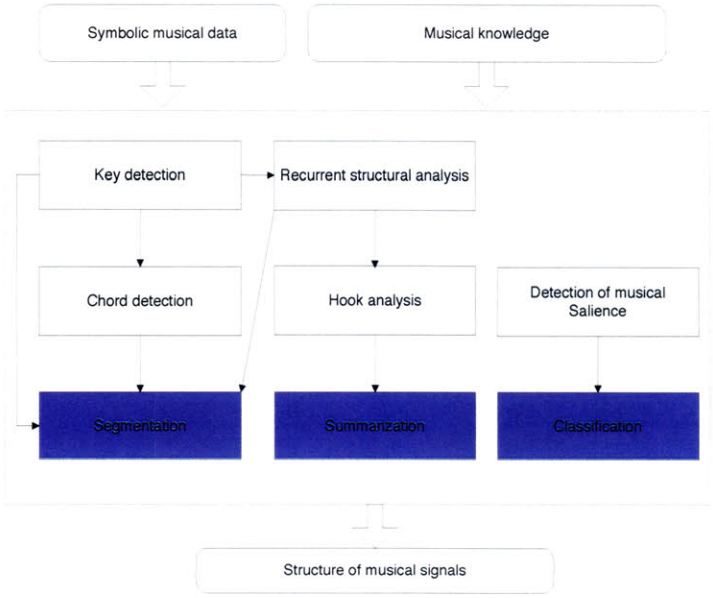


Figure 1-1: Overview of the dissertation.

Accordingly, the remainder of this dissertation is organized along the following chapters: Chapter 2 provides background material of our research, highlighting other studies that we consider most relevant to our goals within the fields of music cognition, musical signal processing, and music information retrieval systems.

In Chapter 3, we describe the system used for tonality and harmonic analysis of music using a probabilistic Hidden Markov Model. It can detect key changes or chord changes with a limited amount of prior musical knowledge. Evaluation methods for music segmentation are also proposed in this chapter.

Chapter 4 presents the system used to detect the recurrent structure of music using dynamic time warping and techniques for self-similarity analysis. It can parse a musical piece into sections and detect the form of it.

Based on the result from chapter 4, chapter 5 proposes strategies for music summarization. An online human experiment is conducted to investigate various problems involved in music summarization and set up a ground truth for developing and evaluating different summarization strategies.

Chapter 6 is related to a problem called musical salience for classification. We describe the problem, present our approach, and demonstrate our theory by experimental results.

We conclude with Chapter 7, in which we evaluate the potential of the framework and discuss some of the system's inherent limitations. We also suggest potential improvements to the framework as well as some general directions for future research.

Additionally, this dissertation with all the sound examples and data sets is online at http://web.media.mit.edu/~chaiwei/thesisWeb/.

Chapter 2 Background

We begin with a short explanation of musical structure emphasizing the music cognition perspective. Following that, we briefly summarize previous and current research on musical signal processing and music information retrieval systems, providing relevant techniques and application contexts related to this dissertation.

2.1 Musical Structure and Meaning

Music inherently contains large amounts of structure. For example, melodic structure and chord structure emphasize the pitch relation between simultaneous or sequential components; formal structure and rhythmic structure emphasize the temporal relation between musical segments. The relation between "surface structure" and "deep structure" of music has also been analyzed (Deliège, 1996). Although it is not clear why humans need music, theories have suggested that hearing the structure of music plays an important role in satisfying this need.

> *Most adults have some childlike fascination for making and arranging larger structures out of smaller ones. One kind of musical understanding involves building large mental structures out of smaller, musical parts. Perhaps the drive to build those mental music structures is the same one that makes us try to understand the world.* (Minsky, 1989)

Thus, music is, in a sense, a mental game for humans, in which we learn to perceive and construct complex structures, especially temporal structures. The music listening process is the one in which we analyze, understand and memorize the musical structure.

> *One of the primary ways in which both musicians and non-musicians understand music is through the perception of musical structure. This term refers to the understanding received by a listener that a piece of music is not static, but evolves and changes over time. Perception of musical structure is deeply interwoven with memory for music and music understanding at the highest levels, yet it is not clear what features are used to convey structure in the acoustic signal or what representations are used to maintain it mentally.* (Scheirer, 1998)

In general, perception and analysis of structure is essential for understanding meanings of things. Minsky (1989) stated that "a thing has meaning only after we have learned some ways to represent and process what it means, or to understand its parts and how they are put together." In natural languages, the syntax of an expression forms a structure, on which meaning is superimposed. Music is also organized in a way through which musicians convey certain meanings and listeners can perceive them, though, unlike natural languages, the meaning of music is more subjective, auto-referential and not easily describable in words. Music has a more direct connection to human emotion than a natural language does; and musical structure is the carrier of the emotional meaning and the expressive power of music. It is the articulate form of music (perceived via various surface cues such as tempo, dynamics, texture) that makes musical sound vivid, expressive and meaningful.

Analysis of musical structure is a fairly broad topic. Our research will focus on building computational models for automating the analysis of the sequential grouping structure of music, including parsing music into parts at various levels, extracting recurrent patterns, exploring the relation between musical parts, and finding the most informative or representative musical parts based on different tasks. In addition, tonality and harmonic structure will also be addressed in the dissertation. Other important aspects related to musical structure, such as metrical structure, melodic structure, sound texture/simultaneous grouping, and emotional meanings of music, will not be the main concerns in the dissertation.

2.2 Musical Signal Processing

2.2.1 Pitch Tracking and Automatic Transcription

Pitch is a perceptual concept, though in normal context monophonic pitch tracking means finding the fundamental frequency (f_0) of the acoustic signal. The monophonic pitch tracking algorithms can be divided into three categories (Rabiner, 1976) (Roads, 1994):

Time domain algorithms, which operate directly on the waveform to estimate the pitch period. Classical algorithms include zero-crossing periodicity detector, peak periodicity detector, autocorrelation pitch detector, etc.

Frequency domain algorithms, which use the property that if the signal is periodic in the time domain, then the frequency spectrum of the signal will consist of a series of impulses at the fundamental frequency and its harmonics. Classical algorithms include Short Time Fourier Transform (STFT) based pitch detector, adaptive filter pitch detector, tracking phase vocoder analysis, cepstrum analysis, etc.

Time- and frequency-domain algorithms, which incorporate features of both the time-domain and the frequency-domain approaches to pitch tracking. For example, a hybrid pitch tracker might use frequency-domain techniques to provide a spectrally flattened time waveform, and then use autocorrelation measurements to estimate the pitch period.

Automatic transcription, which attempts to transcribe acoustic musical signals into score-based representations, involves polyphonic pitch tracking and harmonic analysis. Although the whole problem is not completely solved, several algorithms for multiple-pitch estimation have been proposed (Jbira, 2000) (Klapuri, 2000) (Sterian, 2000).

2.2.2 Tempo and Beat Tracking

Tempo and beat tracking from acoustic musical signals is very important for musical analysis. Goto (2001) attempted to infer the hierarchical beat structure of music based on the onset times of notes, chord changes and drum patterns. Laroche (2001) did transient analysis on the musical signal first and then used a probabilistic model to find the most likely tempo and beat locations. Scheirer (1998) employed comb filters to detect the periodicities in each frequency range and combined the results to infer the beats.

2.2.3 Representations of Musical Signals

Various representations have been proposed for musical signals. The time-domain representation (waveform) and frequency-domain (STFT or spectrogram) representation are the very basic and most widely used ones. Some variations to spectrogram split the frequency range unevenly. For example, constant-Q and cochlear filter bank are designed for simulating the human auditory system. Chromagram is specifically employed for musical signals; it combines the frequency components belonging to the same pitch class and results in a 12-dimensional representation (corresponding to C, C#, D, D#, E, F, F#, G, G#, A, A#, B). Autocorrelogram allows simultaneous representation of pitch and spectral shape for multiple harmonic sounds.

For musical signals, the result of automatic transcription, such as pitch or beat estimation sequence, can serve as a mid-level representation. Foote (1999, 2000) proposed a representation called *similarity matrix* for visualizing and analyzing the structure of music. Each cell in the matrix denotes the similarity between a pair of frames in the musical signal.

Finally, many timbre-related features have been proposed for analyzing musical or instrumental sounds (Martin, 1999), such as spectral centroid, spectral irregularity, pitch range, centroid modulation, relative onset time of partial frequencies, etc.

2.2.4 Music Matching

Many music applications, such as query-by-humming and audio fingerprinting, need to align two musical sequences (either in symbolic or in acoustic format). To tolerate the time flexibility of music, dynamic time warping and hidden Markov models are widely used for aligning speech signals as well as musical signals. Other methods attempted to take rhythm into account in the alignment (Chai, 2001) (Yang, 2001).

2.3 Music Information Retrieval

With the emergence of digital music on the Internet, automating access to music information through the use of computers has intrigued music fans, librarians, computer scientists, information scientists, engineers, musicologists, cognitive scientists, music psychologists, business managers and so on. However, current methods and techniques for building real-world music information retrieval systems are far from satisfactory.

The dilemma was pointed out by Huron (2000). Music librarians and cataloguers have traditionally created indexes that allow users to access musical works using standard reference information, such as the name of the composer and the title of the work. While this basic information remains important, these standard reference tags have surprisingly limited applicability in most music-related queries.

Music is used for an extraordinary variety of purposes: the military commander seeks music that can motivate the soldiers; the restaurateur seeks music that targets certain clientele; the aerobics instructor seeks music of a certain tempo; the film director seeks music conveying a certain mood; an advertiser seeks a tune that is highly memorable; the physiotherapist seeks music that helps provide emotional regulation to a patient; the truck driver seeks music that will keep him alert; the music lover seeks music that can entertain him. Although there are many other uses for music, music's preeminent functions are social, emotional and psychological. The most useful retrieval methods are those that can facilitate searching according to such social, emotional and psychological functions. In fact, an international standard called MPEG7 has been proposed to standardize the metadata for multimedia content and make the retrieval methods more effective.

This section summarizes the status of current research in the field of music information retrieval.

2.3.1 Music Searching and Query by Examples

Music information retrieval systems help provide the users a way to search for music based on its content rather than the reference information. In other words, the system should be able to judge what is similar to the presented query. Retrieving audio based on timbre similarity was studied by Wold (1996), Foote (1999) and Aucouturier (2002). For music, some systems attempted to search for symbolic music based on a hummed tune, called Query-by-humming systems (Ghias, 1995; McNab, 1996; Chai, 2001). Some other systems were developed to retrieve musical recordings based on MIDI data (Shalev-Shwartz, 2002), or based on a short clip of a musical recording (Yang, 2001; Haitsma, 2001). Various audio matching techniques were applied to these systems. In addition, there were studies on query-by-rhythm systems (Foote, 2002). Systems that attempt to combine various aspects of musical similarity for retrieval have also been built. The Cuidado Music Browser (Pachet, 2004) is such a system that can extract the editorial and acoustic metadata from musical signals and retrieve the musical content based on acoustic and cultural similarities.

2.3.2 Music Classification

Music classification is another popular topic in the field of music information retrieval. Some of the research used symbolic data. For example, Dannenberg (1997) presented his work for performance style classification using MIDI data. Chai (2001) conducted an experiment of classifying folk music from different countries based on melodic information using hidden Markov models (HMMs).

The acoustic musical signals were directly used for classification as well. One typical method is to segment the musical signal into frames, classify each frame using various features (e.g., FFT, MFCC, LPC, perceptual filterbank) and different machine learning techniques (e.g., Support Vector Machines, Gaussian Mixture Models, k-NN, TreeQ, Neural Networks), and then assign the piece to the class to which most of the frames belong. This technique works fairly well for timbre-related classifications. Pye (2000) and Tzanetakis (2002) studied genre classification. Whitman (2001), Berenzweig (2001, 2002) and Kim (2002) investigated artist/singer classification. In addition to this frame-based classification framework, some other research on music classification attempted to use features of the whole musical piece for emotion detection (Liu, 2003), or use models capturing the dynamic of the piece (Explicit Time Modelling with Neural Network and Hidden Markov Models) for genre classification (Soltau, 1998).

2.3.3 Music Segmentation and Summarization

Music summarization (or *music thumbnailing*) aims at finding the most representative part, often assumed to be the most frequently repeated section, of a musical piece. Pop/rock music was often used for investigating this problem. Some research (Hsu, 2001) on music thumbnailing dealt with symbolic musical data (e.g., MIDI files and scores). There have also been studies on thumbnailing of acoustic musical signals. Logan (2000) attempted to use a clustering technique or Hidden Markov Models to find key phrases of songs. Bartsch (2001) used the similarity matrix and chroma-based features for music thumbnailing. A variation of the similarity matrix was also proposed for music thumbnailing (Peeters, 2002).

Dannenberg (2002) presented a method to automatically detect the repeated patterns of musical signals. The process consists of searching for similar segments in a musical piece, forming clusters of similar segments, and explaining the musical structure in terms of these clusters. Although the promise of this method was demonstrated by several examples, there was no quantitative evaluation of the method in their paper. Furthermore, it could only give the repeated patterns rather than an overall formal structure of the piece or a semantic segmentation.

A topic closely related to music thumbnailing is *music segmentation*. Most previous research in this area attempted to segment musical pieces by detecting the locations where significant changes of statistical properties occur (Aucouturier, 2001). This method is more appropriate for segmenting local events rather than segmenting the semantic components within the global structure.

Chapter 3 Tonality and Harmony Analysis

Tonality is an important aspect of musical structure. It describes the relationships between the elements of melody and harmony - tones, intervals, chords, and scales - to give the listeners the sense of tonal center. The tonality of music has also been proven to have an impact on the listener's emotional response of music. Furthermore, chords are important harmonic building blocks of tonality. Much literature attempts to analyze the musical structure in terms of chords and chord progression in a way similar to analyzing semantic structure of language in terms of words and grammar.

From the practical perspective, tonality and harmony analysis is a critical step for semantic segmentation of music and detection of repeated patterns in music (shown in Chapter 4), which are important for intelligent music editing, indexing and searching. Therefore, this chapter presents an HMM-based generative model for automatic analysis of tonality and harmonic structure of music.

3.1 Chromagram – A Representation for Musical Signals

The chromagram, also called the Pitch Class Profile features (PCP), is a frame-based representation of audio, very similar to Short-time Fourier Transform (STFT). It combines the frequency components in STFT belonging to the same pitch class (i.e., octave folding) and results in a 12-dimensional representation, corresponding to C, C#, D, D#, E, F, F#, G, G#, A, A#, B in music, or a generalized version of 24-dimensional representation for higher resolution and better control of noise floor. (Sheh, 2003)

Specifically, for the 24-dimensional representation, let $X_{STFT}[K,n]$ denote the magnitude spectrogram of signal $x[n]$, where $0 \le K \le NFFT-1$ is the frequency index, NFFT is the FFT length. The chromagram of $x[n]$ is

$$X_{PCP}[\tilde{K},n] = \sum_{K:\, P(K)=\tilde{K}} X_{STFT}[K,n] \tag{3-1}$$

The spectral warping between frequency index K in STFT and frequency index $\tilde{K}$ in PCP is

$$P(K) = [24 \cdot \log_2(K / NFFT \cdot f_s / f_1)] \bmod 24 \tag{3-2}$$

where f_s is the sampling rate, f_1 is the reference frequency corresponding to a note in the standard tuning system, for example, MIDI note C3 (32.7Hz),.

In the following, we will use the 24-dimensional PCP representation. To investigate some properties of the 24-dimensional PCP representation $X_{PCP}[K,n]$ (K=1, ..., 24; n=1, ..., N) of a musical signal of N frames, let us denote

$$m[K,n] = \begin{cases} 1, \textit{if } X_{PCP}[K,n] >= X_{PCP}[K-1,n] \textit{ and } X_{PCP}[K,n] >= X_{PCP}[K+1,n] \\ 0, \textit{otherwise.} \end{cases} \tag{3-3}$$

where we define $X_{PCP}[0,n] = X_{PCP}[24,n]$ and $X_{PCP}[25,n] = X_{PCP}[1,n]$ for the boundary conditions. Thus, $m[K,n]$ is a binary matrix denoting whether the magnitude at a particular frequency in the PCP representation is the local maximum comparing to magnitudes at its two

neighboring frequencies. We then can count the number of local maxima appearing at the odd frequency indexes or appearing at the even frequency indexes, and compare them:

$$l_{odd} = \frac{1}{24 \cdot N} \sum_{n=1}^{N} \sum_{K \text{ is odd}} m[K,n] \tag{3-4}$$

$$l_{even} = \frac{1}{24 \cdot N} \sum_{n=1}^{N} \sum_{K \text{ is even}} m[K,n] \tag{3-5}$$

$$r_{even/odd} = \frac{l_{even}}{l_{odd}} \tag{3-6}$$

If all the instruments in a musical piece are well tuned (tones are strongly pitched and the pitches match the twelve pitch classes perfectly) and the energy of each tone concentrates on its fundamental frequency (f_0), we can easily conclude that $l_{odd} >> l_{even}$ and $r_{even/odd} \to 0$. However, if the instruments are not well tuned, or some instruments are not strongly pitched (e.g., drum, some fricatives in vocal), or the harmonics of tones are strong (f_1, f_2, etc), then $l_{even} \to l_{odd}$ and $r_{even/odd} \to 1$ (It would be rare that l_{even} gets bigger than l_{odd}). This property can be related to the musical genre. To show this, l_{odd}, l_{even} and $r_{even/odd}$ were computed for 21 classical piano pieces and 26 Beatles songs (Appendix A), respectively, and their distributions are plotted in Figure 3-1. The left plot shows the distribution of (l_{odd}, l_{even}), in which each point corresponds to a musical piece. The right plot shows the Gaussian probability density estimation of $r_{even/odd}$. The result is consistent with the above analysis.

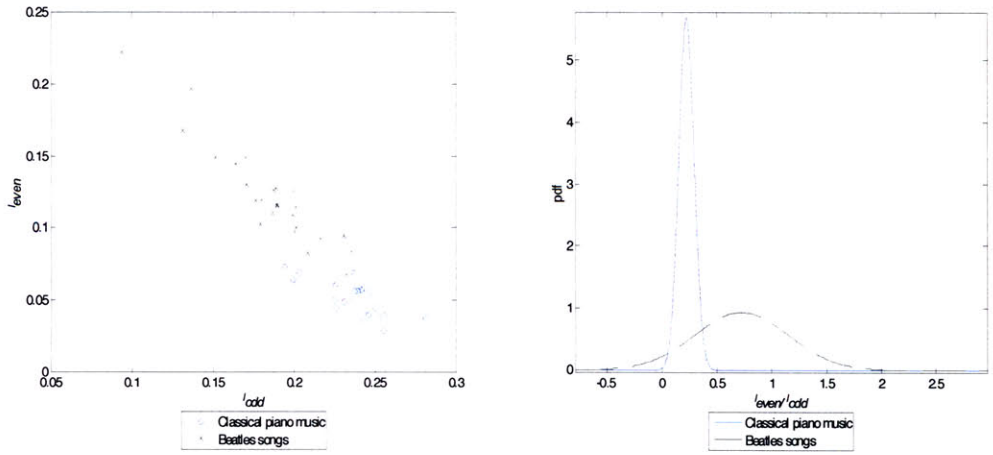


Figure 3-1: Scatterplot of (l_{odd}, l_{even}) (left) and the Gaussian probability density estimation of $r_{even/odd}$ (right) for classical piano music and Beatles songs.

In the following thesis (except Chapter 6), we will focus on the chromagram representation for further analysis of musical structure, simply because of its advantage of direct mapping to musical notes. It doesn't mean it is best for all types of applications or all musical genres. In some comparisons between different representations for music structural analysis tasks (Chai, 2003), it was shown that no representation is significantly better for all musical data. Therefore, we will

focus on one representation in this dissertation; all the following approaches can be generalized fairly easily using other representations.

3.2 Detection of Key Change

This section describes an algorithm for detecting the key (or keys) of a musical piece. Specifically, given a musical piece (or part of it), the system will segment it into sections based on key change and identify the key of each section. Note that here we want to segment the piece and identify the key of each segment at the same time. A simpler task could be: given a segment of a particular key, detect the key of it.

3.2.1 Musical Key and Modulation

In Music theory, the **key** is the tonal center of a piece. It is designated by a note name (the tonic), such as C, and can be either in major or minor mode. Other modes are also possible. The major mode has half-steps between scale steps 3 and 4, and 7 and 8. The natural minor mode has half-steps between 2 and 3, and 5 and 6.

A **scale** is an ascending or descending series of notes or pitches The **chromatic scale** is a musical scale that contains all twelve pitches of the Western tempered scale. The **diatonic scale** is most familiar as the major scale or the "natural" minor scale. The diatonic scale is a very important scale. Out of *all* the possible seven note scales it has the highest number of consonant intervals, and the greatest number of major and minor triads. The diatonic scale has *six* major or minor triads, while all of the remaining prime scales (the harmonic minor, the harmonic major, the melodic and the double harmonic) have just *four* major or minor triads. The diatonic scale is the **only** seven note scale that has just **one** tritone (augmented fourth/diminished fifth). All other scales have **two**, or more, tritones. In the following, we will often assume diatonic scales where it is necessary.

A piece may change key at some point. This is called **modulation**. Modulation is sometimes done by just starting in the new key with no preparation - this kind of key change is common in various kinds of popular music, when a sudden change to a key a whole tone higher is a quite frequently heard device at the end of a song. In classical music, however, a "smoother" kind of key change is more usual. In this case, modulation is usually brought about by using certain chords, which are made up of notes ("pivot notes") or chords ("pivot chords") common to both the old key and the new one. The change is solidified by a cadence in the new key. Thus, it is smoother to modulate to some keys (i.e., nearly related keys) than others, because certain keys have more notes in common with each other than others, and therefore more possible pivot notes or chords. Modulation to the **dominant** (a fifth above the original key) or the **subdominant** (a fourth above) is relatively easy, as are modulations to the relative major of a minor key (for example, from C minor to E flat major) or to the relative minor of a major key (for example, from C major to A minor). These are the most common modulations, although more complex changes are also possible.

The purpose of modulation is to give direction and variety in music structure. Modulation in a piece is often associated with the formal structure of a piece. Using modulation properly can increase the expressiveness, expand the chromatic contrast, support the development of the theme, and adapt better to the range of the instruments and voice.

At times there might be ambiguity of key. It can be hard to determine the key of quite long passages. Some music is even **atonal**, meaning there is no tonal center. Thus, in this dissertation, we will focus on tonal music with the least ambuiguity of tonal center.

3.2.2 Hidden Markov Models for Key Detection

In the following, the task of key detection will be divided into two steps:

1. Detect the key without considering its mode. For example, both C major and A minor will be denoted as key 1, C# major and A# minor will be denoted as key 2, and so on. Thus, there could be 12 different keys in this step.

2. Detect the mode (major or minor).

The task is divided in this way because diatonic scales are assumed and relative modes share the same diatonic scale. Step 1 attempts to determine the height of the diatonic scale. And again, both steps involve segmentation based on key (mode) change as well as identification of keys (modes).

The model used for key change detection should be able to capture the dynamic of sequences, and to incorporate prior musical knowledge easily since large volume of training data is normally unavailable. We propose to use Hidden Markov Models for this task, because HMM is a generative model for labeling structured sequence and satisfying both of the above properties.

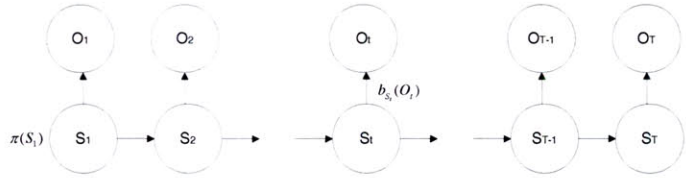


Figure 3-2: Demonstration of Hidden Markov Models.

HMM (Hidden Markov Model) is a very powerful tool to statistically model a process that varies in time. It can be seen as a doubly embedded stochastic process with a process that is not observable (sequence of hidden states) and can only be observed through another stochastic process (sequence of observable states) that produces the time set of observations. Figure 3-2 shows a graph of HMM used for key change detection. The hidden states correspond to different keys (or modes). The observations correspond to each frame represented as 24-dimensional chromagram vectors. The task will be decoding the underlying sequence of hidden states (keys or modes) from the observation sequence using Viterbi approach.

The parameters of HMM that need to be configured include:

- The number of states N corresponding to the number of different keys (=12) or the number of different modes (=2), respectively, in the two steps.

- The state transition probability distribution $\mathbf{A} = \{a_{ij}\}$ corresponding to the probability of changing from key (mode) *i* to key (mode) *j*. Thus, A is a 12×12 matrix in step 1 and a 2×2 matrix in step 2, respectively.

- The initial state distribution $\Pi = \{\pi_i\}$ corresponding to the probability at which a piece of music starts from key (mode) *i*.

- The observation probability distribution $\mathbf{B} = \{b_j(v)\}$ corresponding to the probability at which a chromagram *v* is generated by key (mode) *j*.

Due to the small amount of labeled audio data and the clear musical interpretation of the parameters, we will directly incorporate the prior musical knowledge by empirically setting Π and A as follows:

$$\Pi = \frac{1}{12} \cdot \mathbf{1}$$

where $\mathbf{1}$ is a 12-dimensional vector in step 1 and a 2-dimensional vector in step 2. This configuration denotes equal probabilities of starting from different keys (modes).

$$\mathbf{A} = \begin{bmatrix} stayprob & b & \dots & b \\ b & stayprob & \dots & b \\ b & b & \dots & b \\ b & b & \dots & stayprob \end{bmatrix}_{d \times d}$$

where d is 12 in step 1 and is 2 in step2. *stayprob* is the probability of staying in the same state and $stayprob + (d-1) \cdot b = 1$. For step 1, this configuration denotes equal probabilities of changing from a key to a different key. It can be easily shown that when *stayprob* gets smaller, the state sequence gets less stable (changes more often). In our experiment, *stayprob* will be varying within a range (e.g., [0.9900 0.9995]) in step 1 and be set to $1-10^{-12}$ or $1-10^{-20}$ in step 2 to see how it impacts the performance.

For observation probability distribution, instead of Gaussian probabilistic models, commonly used for modeling observations of continuous random vectors in HMM, the cosine distances between the observation (the 24-dimensional chromagram vector) and pre-defined template vectors were used to represent how likely the observation was emitted by the corresponding keys or modes, i.e.,

$$b_j(v) = \frac{v.\theta_j}{\| v \| . \| \theta_j \|} \qquad (3\text{-}7)$$

where θ_j is the template of state j (corresponding to the j^{th} key or mode). The advantage of using cosine distance instead of Gaussian distribution is that the key (or mode) is more correlated with the relative amplitudes of different frequency components rather than the absolute values of the amplitudes. Figure 3-3 shows an example for demonstrating this. Suppose points A, B and C are three chromagram vectors. Based on musical knowledge, B and C are more likely to be generated by the same key (or, mode) than A and C, because B and C have more similar energy profiles. However, if we look at the Euclidean space, A and C are closer to each other than B and C; thus, if we use a Gaussian distribution to model the observation probability distribution, A and C will be more likely to be generated by the same key, which is not true.

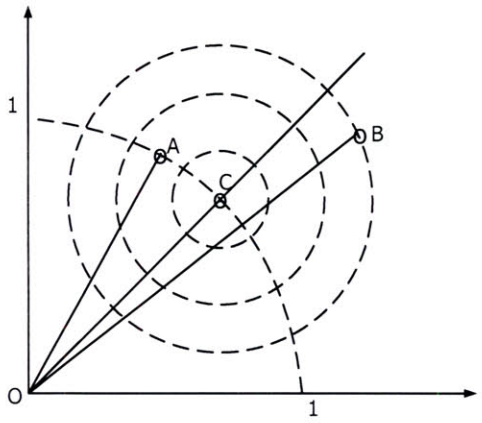


Figure 3-3: Comparison of observation distributions of Gaussian and cosine distance.

For step 1, two ways for configuring the templates of keys were explored:

1) The template of a key was empirically set corresponding to the diatonic scale of that key. For example, the template for key 1 (C major or A minor) is $\theta_1^{odd} = [1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1]^T$ (Figure 3-4), $\theta_1^{even} = \mathbf{0}$, where θ_1^{odd} denotes the sub-vector of θ_1 with odd indexes (i.e., $\theta_1(1:2:23)$) and θ_1^{even} denotes the sub-vector of θ_1 with even indexes (i.e., $\theta_1(2:2:24)$). This means we ignore the elements with even indexes when calculating the cosine distance. The templates of other keys were set simply by rotating θ_1 accordingly:

$$\theta_j = r(\theta_1, 2\cdot(j-1)) \tag{3-8}$$

$$\beta = r(\alpha, k),\ s.t.\ \beta[i] = \alpha[(k+i) \bmod 24]$$

where j=1, 2, …, 12 and i, k=1, 2, …, 24. Let us also define $24 \bmod 24 = 24$.

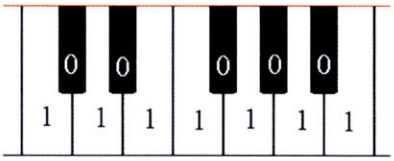


Figure 3-4: Configuration of the template for C major (or A minor).

2) The template of a key was learned from symbolic musical data. The symbolic data set used to train the template includes 7,673 folk music scores, which are widely used for music informatics research. The template was generated as follows: get the key signature of each piece and assume it is the key of that piece (occasionally the key of a piece might be different from the key signature); count the number of times that each note (octave-equivalent and relative to the key) appears (i.e., a 12-dimensional vector corresponding to do-do#-re-re#-mi-fa-fa#-sol-sol#-la-la#-ti); average the vectors over all pieces and normalize it. Similar to method 1), we assign θ_1^{odd} to be the normalized vector, $\theta_1^{even} = \mathbf{0}$, and $\theta_j = r(\theta_1, 2\cdot(j-1))$.

A comparison of the templates generated by the above two ways is shown in Figure 3-5.

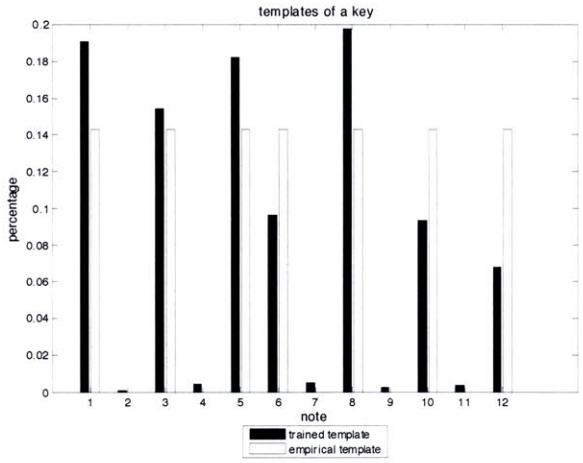


Figure 3-5: Configurations of templates - θ_1^{odd} (trained template and empirical template).

For step 2, the templates of modes were empirically set as follows:

$$\theta_{major}^{odd} = [0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0]^T,$$

$$\theta_{minor}^{odd} = [0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0]^T,$$

$$\theta_{major}^{even} = \theta_{minor}^{even} = 0,$$

This setting comes from musical knowledge that typically in a major piece, the dominant (G in C major) appears more often than the submediant (A in C major), while in a minor piece, the tonic (A in A minor) appears more often than the subtonic (G in A minor). Note the templates need to be rotated accordingly (Equation 3-8) based on its key detected from step 1.

The above is a simplified model and there can be several refinements of it. For example, if we consider the prior knowledge of modulation, we can encode in $\mathbf{A}$ the information that each key tends to change to its "close" keys rather than the other keys. The initial key or mode of a piece may not be uniformly distributed as well. But to quantize the numbers, we will need a very large corpus of pre-labeled musical data, which is not available here.

3.3 Detection of Chord Progression

Using the same approach, this section describes the algorithm to analyze the chord progression. Specifically, given a section (or part of it) and the key (assuming no key change within the section), we want to segment it based on chord change and identify the chord of each segment. Additionally, the algorithm does not require an input of mode and, if the mode is not provided, the algorithm can identify the mode (major or minor) of the section based on the result of chord progression analysis. That means this section provides another way of mode detection besides the one presented in the last section.

The most commonly used chords in western music are **triads**, which are the basis of diatonic harmony and are composed of three notes: a root note, a note which is an interval of a third above the root, and a note which is an interval of a fifth above the root.

In the model for detecting chord progression, two HMMs were built – one for major and one for minor. The given section will be analyzed by the two models separately and classified to the mode whose corresponding model outputs a larger log-likelihood. Each model has 15 states corresponding to 14 basic triads for each mode (see below; uppercase Roman numerals are used for major triads; lowercase Roman numerals for minor triads; a small circle superscript for diminished) and one additional state for "other triads":

Major: I, ii, iii, IV, V, vi, vii°; i, II, III, iv, v, VI, vii

Minor: i, ii°, III, iv, V, VI, vii°; I, ii, iii, IV, v, vi, V

Again, the parameters for either HMM were set empirically based on their musical interpretation:

$$\pi_{chord}(1) = 0.63$$

$$\pi_{chord}(2:7) = 0.05$$

$$\pi_{chord}(8:14) = 0.01$$

$$\pi_{chord}(15) = 0$$

$$A_{chord} = \begin{bmatrix} stayprob & b & \dots & b \\ b & stayprob & \dots & b \\ b & b & \dots & b \\ b & b & \dots & stayprob \end{bmatrix}_{15\times 15}$$

where *stayprob* will be varying in a range (e.g., [0.70 0.90]) to see how it impacts the performance. Again, this configuration denotes equal probabilities of changing from a triad to a different triad. The configuration of the initial state probabilities denotes uneven probabilities of starting from different triads: most likely to start from tonic, less likely to start from other diatonic triads, and least likely to start from other triads, assuming the input section starts from the beginning of a musical phrase.

Similarly, the observation probability distributions were obtained by calculating the cosine distances between observations and templates of triads. The template of a triad is configured to correspond to the three notes of that triad. For example, the template with odd indexes for a major tonic triad (I in major mode) in key 1 is $\theta_1^{odd} = [1\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0]^T$; the template for a minor tonic triad (i in minor mode) is $\theta_1^{odd} = [1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0]^T$. Note that since we have been given the key of the section, we can rotate the 24-dimensioanl chromagram representation accordingly (Equation 3-8) in advance to always make the first dimension the tonic for major mode or the 10th dimension the tonic for minor mode.

3.4 Evaluation Method

To evaluate the results, two aspects need to be considered: label accuracy (how the computed label of each frame is consistent with the actual label) and segmentation accuracy (how the detected locations of transitions are consistent with the actual locations).

Label accuracy is defined as the proportion of frames that are labeled correctly, i.e.,

$$Label\ accuracy = \frac{\#\ frames\ labeled\ correctly}{\#\ total\ \ frames} \tag{3-9}$$

Two metrics were proposed and used for evaluating segmentation accuracy. *Precision* is defined as the proportion of detected transitions that are relevant. *Recall* is defined as the proportion of relevant transitions detected.

Thus, if B={relevant transitions}, C={detected transitions} and $A = B \cap C$, from the above definition,

$$Precision = \frac{A}{C} \tag{3-10}$$

$$Recall = \frac{A}{B} \tag{3-11}$$

Figure 3-6: An example for measuring segmentation performance (above: detected transitions; below: relevant transitions).

To compute precision and recall, we need a parameter *w*: whenever a detected transition t_1 is close enough to a relevant transition t_2 such that $|t_1-t_2|<w$, the transitions are deemed identical (a *hit*). Obviously, greater *w* will result in higher precision and recall. In the example shown in Figure 3-6, the width of each shaded area corresponds to $2w-1$. If a detected transition falls into a shaded area, there is a *hit*. Thus, the precision in this example is 3/6=0.5; the recall is 3/4=0.75. Given *w*, higher precision and recall indicates better segmentation performance. In our experiment (512 window step at 11kHz sampling rate), *w* will vary within a range to see how precision and recall vary accordingly: for key detection, *w* varies from 10 frames (~0.46s) to 80 frames (~3.72s); for chord detection, it varies from 2 frames (~0.09s) to 10 frames (~0.46s). The range of *w* for key detection is fairly large because modulation of music (change from one key to another key) is very often a smooth process that may take several bars.

Now, we can analyze the baseline performance of random segmentation for future comparison of computed results. Assume we randomly segment a piece into $(k+1)$ parts, i.e., *k* random detected transitions. Let *n* be the length of the whole piece (number of frames in our case) and let *m* be the number of frames "close enough" to each relevant transition, i.e., $m=2w-1$. Also assume there are *l* actual segmenting points. To compute average precision and recall of random segmentation, the problem can be categorized as a hyper-geometric distribution: if we choose *k* balls randomly from a box of *ml* black balls (i.e., *m* black balls corresponding to each segmenting point) and $(n-ml)$ white balls (assuming no overlap occurs), what is the distribution of the number of black balls we get. Thus,

$$\text{Precision} = \frac{\text{E}[\#\ black\ balls\ chosen]}{k} = \frac{1}{k} \cdot \frac{mlk}{n} = \frac{ml}{n} \tag{3-12}$$

$$\text{Recall} = \frac{E[\#\ detected\ \ segmenting\ \ points]}{l} = \frac{l \cdot P(B>0)}{l} = 1 - P(B=0)$$
$$= 1 - \frac{C_m^0 C_{n-m}^{k-0}}{C_n^k} = 1 - (1 - \frac{k}{n})(1 - \frac{k}{n-1})...(1 - \frac{k}{n-m+1}) \quad (3\text{-}13)$$

where B denotes the number of black balls chosen corresponding to a particular segmenting point and C_n^k is the notation of combination corresponding to the number of ways of picking k *unordered* outcomes from n possibilities. If we know the value of l in advance and make $k=l$ (thus, not completely random), and $n>>m$,

$$\text{Recall} \approx 1 - (1 - \frac{l}{n})^m \quad (3\text{-}14)$$

The equations shown that, given n and l, precision increases by increasing w (i.e., increasing m); recall increases by increasing k *or* w. Equation 3-12 and 3-14 will be used later as the baseline (upper bound of the performance of random segmentation) to be compared to the performance of the segmentation algorithms.

3.5 Experiments and Results

3.5.1 Performance of Key Detection

Ten classical piano pieces (see Appendix A-1) were used in the experiment of key detection, since the chromagram representation of piano music has a good mapping between its structure and its musical interpretation (Section 3.1). These pieces were chosen randomly as long as they have fairly clear tonal structure (relatively tonal instead of atonal). The "truth" was manually labeled by the author based on the score notation for comparison with the computed results. The data were mixed into 8-bit mono and down-sampled to 11kHz. Each piece was segmented into frames of 1024 samples with 512 samples overlap.

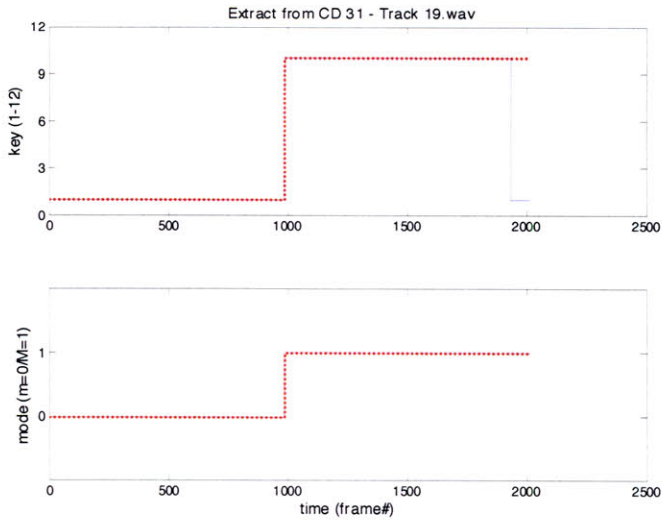


Figure 3-7: Detection of key change in "Mozart: Sonata No. 11 In A 'Rondo All Turca, 3rd movement'" (solid line: computed key; dotted line: truth)

Figure 3-7 shows key detection result of Mozart's piano sonata No. 11 with *stayprob*=0.996 in step 1 and *stayprob2*=1-.1^{-20} in step 2. The figure above presents the result of key detection without considering mode (step 1) and the figure below presents the result of mode detection (step 2).

To show label accuracy, recall and precision of key detection averaged over all the pieces, we can either fix *w* and change *stayprob* (Figure 3-8), or fix *stayprob* and change *w* (Figure 3-9). For either case, there are four plots corresponding to four different combinations: *stayprob2*=1-.1^{-12} or *stayprob2*=1-.1^{-20}; using empirically configured templates or trained templates.

In Figure 3-8, two groups of results are shown in each plot: one corresponds to the performance of step 1 without considering modes; the other corresponds to the overall performance of key detection with mode into consideration. It clearly shows that, when *stayprob* increases, precision also increases while recall and label accuracy decrease. By comparing the plots in the left column to the plots in the right column, we will see when *stayprob2* increases, precision and label accuracy increase while recall decreases. Surprisingly, by comparing the plots in the upper row to the plots in the lower row, we can find the performance using trained templates is worse than the performance using empirically configured ones. It suggests that the empirical configuration encodes the musical meaning well and thus is closer to the ground truth than the trained ones based on our symbolic data corpus due to insufficiency of data or mismatch of musical styles (classical piano music and folk music).

In Figure 3-9, three groups of results are shown in each plot: one corresponds to the performance of step 1 without considering modes; one corresponds to the overall performance of key detection with mode taken into consideration; and one corresponds to recall and precision based on random segmentation (Equation 3-12 and 3-14).

Additionally, label accuracy based on random should be around 8%, without considering modes. It clearly shows that when *w* is increasing, the segmentation performance (recall and precision) is also increasing. Note that label accuracy is irrelevant to *w*. Again, by comparing the plots in the left column to the plots in the right column, we will see when *stayprob2* gets bigger, precision and label accuracy get bigger while recall gets smaller. By comparing the plots in the upper row to the plots in the lower row, we can find the performance using trained templates is worse than the performance using empirically configured ones. The figure also shows that the segmentation performance (recall and precision) base on the algorithm is significantly better than random segmentation.

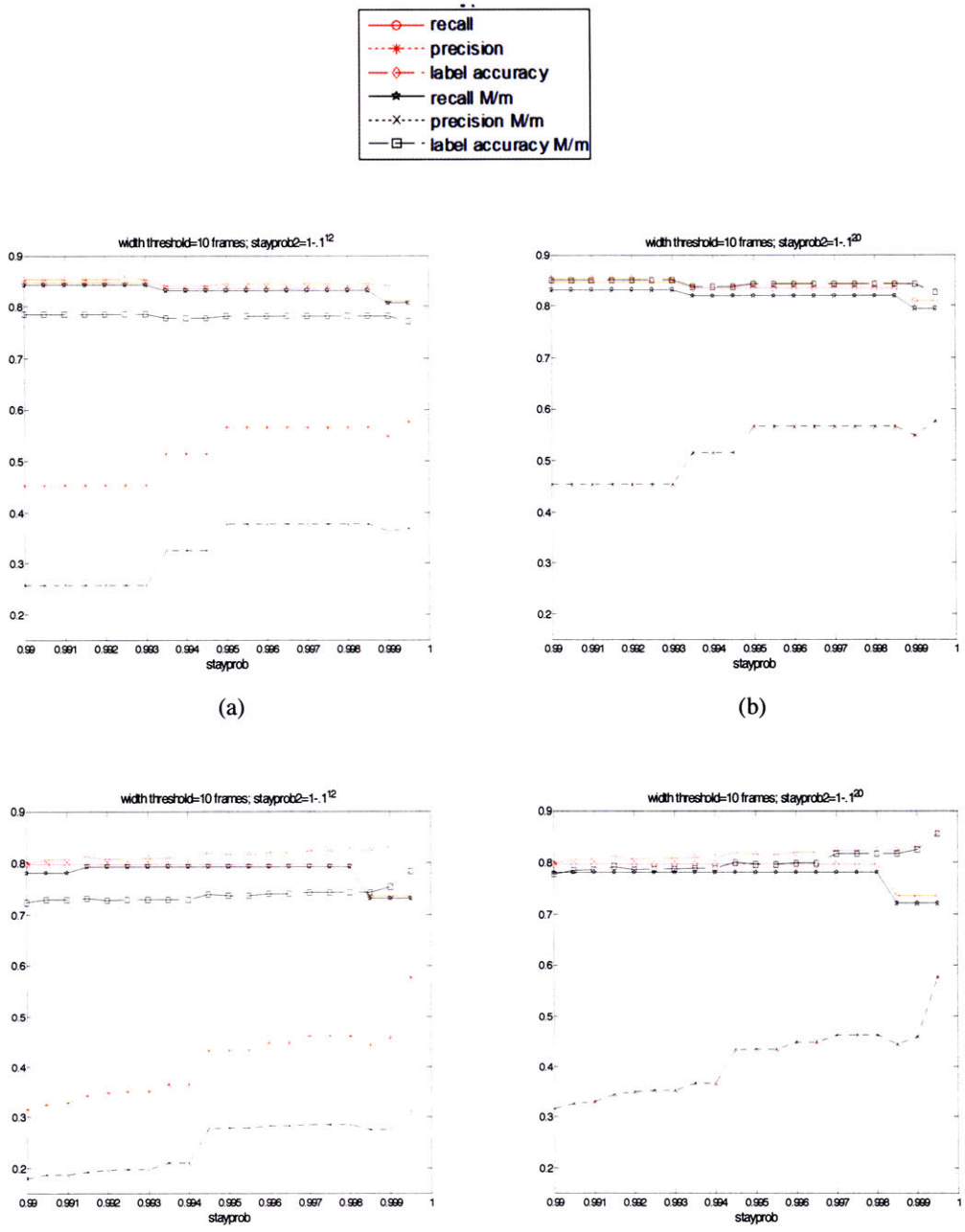


Figure 3-8: Performance of key detection with varying *stayprob* (*w*=10). (a) Empirical templates, *stayprob2*=1-.1^{-12}; (b) Empirical templates, *stayprob2*=1-.1^{-20}; (c) Trained templates, *stayprob2*=1-.1^{-12}; (d) Trained templates, *stayprob2*=1-.1^{-20}.

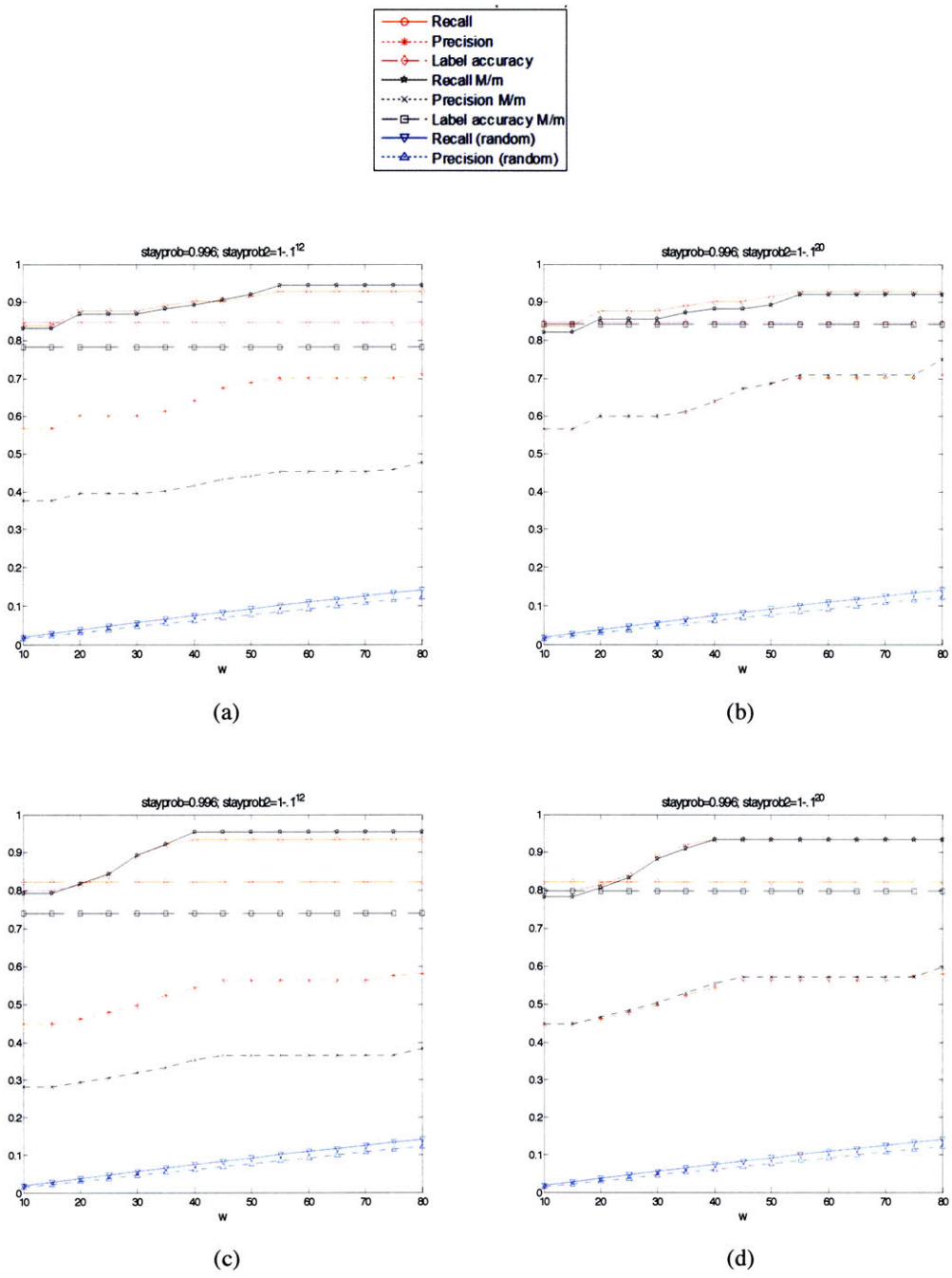


Figure 3-9: Performance of key detection with varying *w* (*stayprob*=0.996). (a) Empirical templates, stayprob2=1-.1^{-12}; (b) Empirical templates, stayprob2=1-.1^{-20}; (c) Trained templates, stayprob2=1-.1^{-12}; (d) Trained templates, stayprob2=1-.1^{-20}.

3.5.2 Performance of Chord Detection

For investigating the performance of chord detection, we truncated the first 8 bars of each of the ten piano pieces and labeled the truth based on the score notation. Since the chord system we investigated is a simplified set of chords, which includes only diatonic triads, each chord was labeled the one in the simplified set that was closest to the original one. For example, a dominant seventh (e.g., G7 in C major) will be labeled as a dominant triad (e.g., G in C major).

Figure 3-10 shows chord detection result of Rubenstein's Melody In F with *stayprob*=0.85. The legend indicates the detected mode and the actual mode.

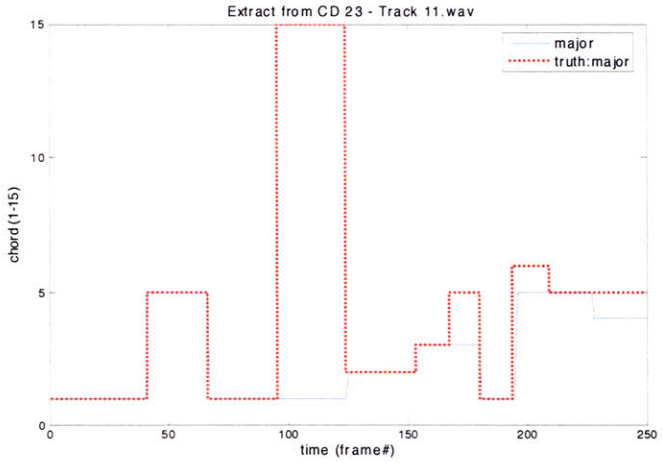


Figure 3-10: Chord detection of "Rubenstein: Melody In F" (solid line: computed chord progression; dotted line: truth)

Similar to the last section, to show label accuracy, recall and precision of chord detection averaged over all the pieces, we can either fix *w* and change *stayprob* (Figure 3-11), or fix *stayprob* and change *w* (Figure 3-12). Figure 3-11 clearly shows that when *stayprob* is increasing, precision is also increasing while recall and label accuracy are decreasing. Figure 3-12 clearly shows that when *w* is increasing, the segmentation performance (recall and precision) is also increasing. Again, label accuracy is irrelevant to *w* and label accuracy using random segmentation should be around 7% given the mode. Therefore, the segmentation performance (recall and precision) and label accuracy base on the algorithm are significantly better than random segmentation. Note that the plotted recall of random segmentation is only an upper bound of the actual recall because it assumes to know the number of chord transitions in advance (Equation 3-14).

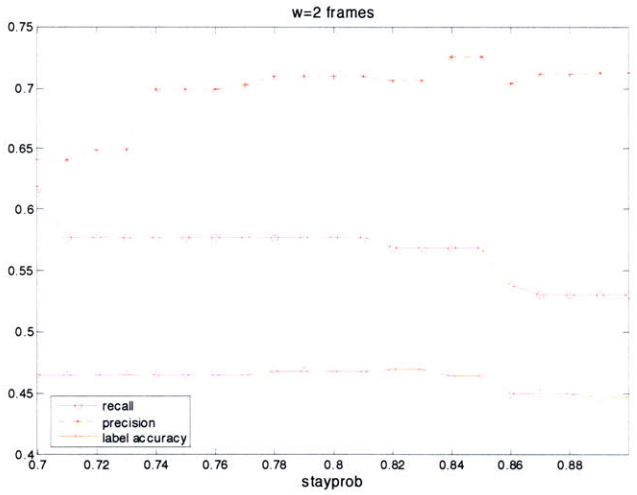


Figure 3-11: Performance of chord detection with varying *stayprob* (*w*=2).

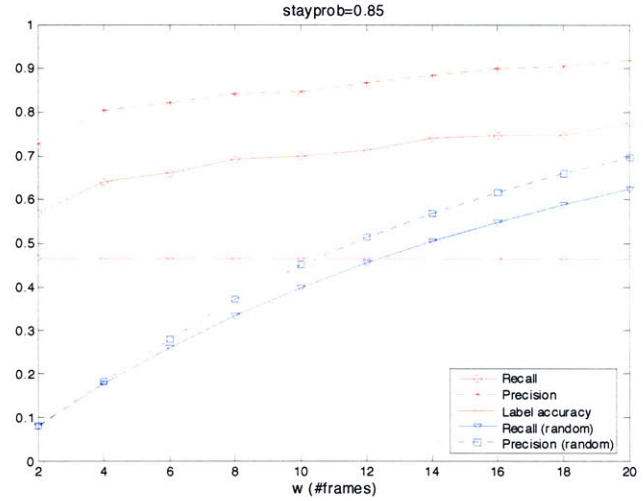


Figure 3-12: Performance of chord detection with varying *w* (*stayprob*=0.85).

The algorithm of chord detection also outputs the mode of each segment based on the log-likelihoods of HMMs, assuming there is no key/mode change during the 8 bars and the segment is long enough to get a sense of mode. The performance of mode detection is that nine out of ten were correct. The ten segments include 8 major pieces and 2 minor pieces. The only error occurs on one of the minor pieces.

3.6 Discussion

Ideally, all the HMM parameters should be learned from a labeled musical corpus. The training can be made (efficiently) using a maximum likelihood (ML) estimate that decomposes nicely since all the nodes are observed. In particular, if the training set has the similar timbre property as the test set, the observation distribution can be more accurately estimated employing the timbre information besides prior musical knowledge, and the overall performance should be further improved.

However, this training data set must be very huge; and manually labeling it will involve a tremendous amount of work. For example, if the training data set is not big enough, the state transition matrix will be very sparse (0's at many cells) and this may result in many test errors, because any transition that does not appear in the training set will not be recognized. Figure 3-13 shows the chord transition matrix trained by the chord detection data set in the same experiment. One can imagine that the key transition matrix will be much sparser even with a much bigger data set because key changes less often than chords. One possibility for future improvement is using Bayesian approach to combine the prior knowledge (via empirical configurations) and the information obtained from a small amount of training data.

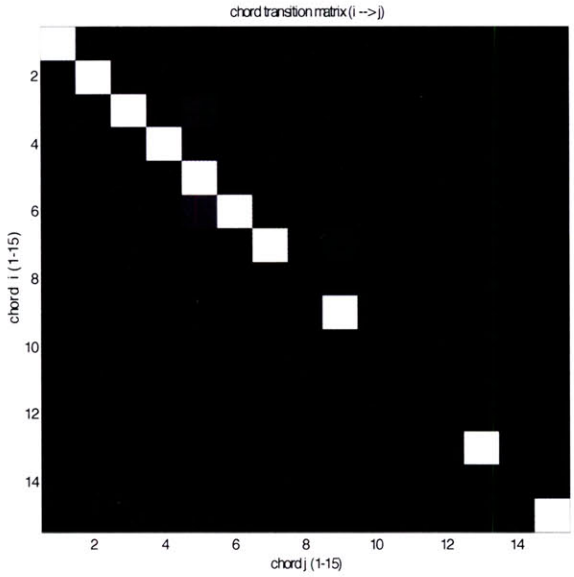


Figure 3-13: Chord transition matrix based on the data set in the experiment.

Another interesting thing to investigate is how the algorithm was confused with keys or chords and whether the errors make musical sense. Figure 3-14 shows the confusion matrices of key detection (without considering modes; *stayprob*=0.996; *stayprob2*=1-.1^{-20}) and chord detection (*stayprob*=0.85).

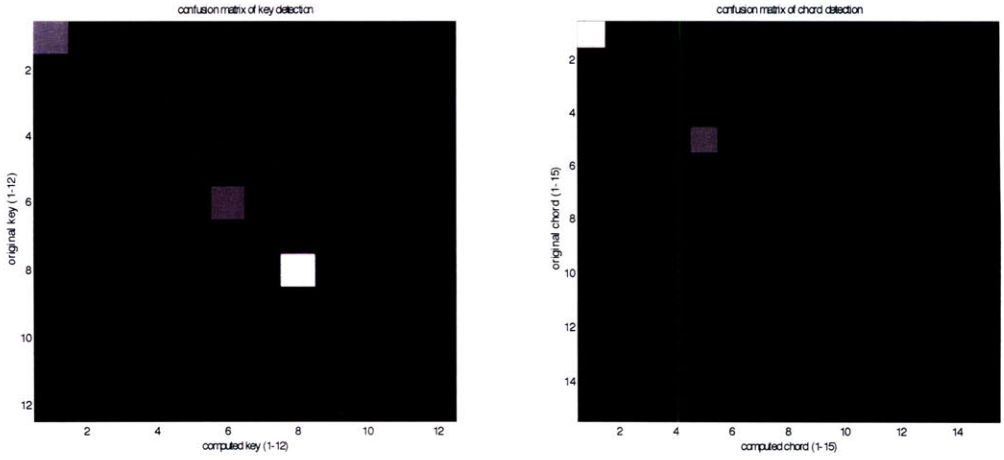


Figure 3-14: Confusion matrix (left: key detection; right: chord detection).

For key detection, most errors came from confusion between the original key and the dominant or sub-dominant key (e.g., F → C, G → C, F# → C#). This is consistent with music theory that these keys are closer to each other and share more common notes. For chord detection, most errors came from confusion between two triads that share two common notes (e.g., I → iii or i → III, I → vi or i → VI), or, less frequently, from confusion between a triad and its dominant or sub-dominant triad (IV → I or iv → I, V → I or V → i).

Finally, segmentation based on chord change can be another path to beat or tempo detection, because chords typically change on beat or bar boundaries. Previous research on beat tracking typically focused on energy information to infer beats while ignored chord analysis.

We have used two data sets, classical piano music (same as the one used in Section 3.5) and Beatles songs (same as the one used for Figure 3-1) to investigate whether the chord detection result is correlated with beat change. We manually labeled the average tempo of each piece, ran the chord detection algorithm for each whole piece, and computed the ratio of each chord change interval and the beat duration. Figure 3-15 shows the distribution of the ratios on these two data sets. Interestingly, there are two peaks, corresponding to ratios equal to 1 and 2, for piano music, while there is one peak, corresponding to ratio equal to 4. This is consistent with our intuition, suggesting chords tend to change every one or two beats in the classical piano music, while they tend to change every measure (four beats) in Beatles songs. For either case, it shows chord change detection result has a good consistency with beat change and thus the algorithm can be used as a supplemental way for beat detection.

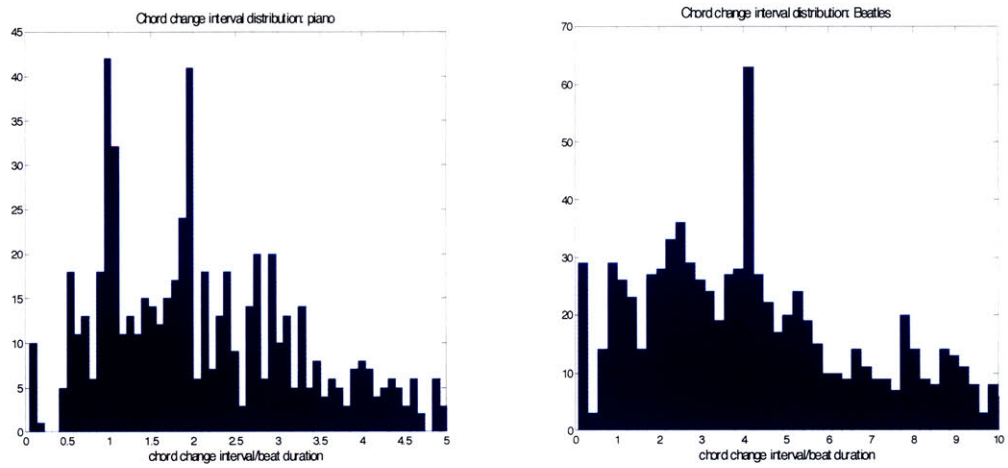


Figure 3-15: Distribution of chord change interval divided by beat duration (left: classical piano music; right: Beatles songs).

3.7 Summary

This chapter presented an HMM-based approach for detecting key change and chord progression. Although constraints on music have been made to build simplified models, e.g., diatonic scales, simplified chord sets, the framework should be easily generalized to handle more complicated music. Each step was carefully designed with consideration of its musical meaning: from using chromagram representation, to employing cosine-distance observation probability distribution, to empirical configurations of HMM parameters. The experimental results, significantly better than random segmentation, have demonstrated the promise of the approach. Future improvement could

be adding a training stage (if training data is available) to make this general model customized to specific types of music. Furthermore, the HMM parameters should be chosen properly according to different applications: for segmentation-based applications, we should maximize precision and recall; for key relevant applications (such as detecting repeated patterns that will be presented in the next chapter), we should maximize label accuracy.

Chapter 4 Musical Form and Recurrent Structure

Music typically has a recurrent structure. Methods for automatically detecting the recurrent structure of music from acoustic signals are valuable for information retrieval systems. For example, the result can be used for indexing the digital music repository, segmenting music at transitions for intelligent editing systems, and summarizing the thumbnails of music for advertisement or recommendation. This chapter describes research into automatic identification of the recurrent structure of music from acoustic signals. Specifically, an algorithm will be presented to output structural information, including both the form (e.g., AABABA) and the boundaries indicating the beginning and the end of each section. It is assumed that no prior knowledge about musical forms or the length of each section is provided, and the restatement of a section may have variations (e.g., different lyrics, tempos). This assumption requires both robustness and efficiency of the algorithm. The result will be quantitatively evaluated by structural similarity metrics, in addition to the qualitative evaluation presented by figures.

4.1 Musical Form

Musical structure has various layers of complexity in any composition. These various layers exist in a continuum ranging from the micro (small) level to the macro (large) level of musical structure. At the micro-level, the smallest complete unit of musical structure is a *phrase*, which comprises patterns of material fashioned from *meter*, *tempo*, *rhythm*, *melody*, *harmony*, *dynamics*, *timbre*, and *instruments*. A *phrase* is a length of musical material existing in real time with a discernible beginning and ending. Each individual melodic phrase may be broken down into smaller incomplete units of melodic structure known as *motives*. Thus, the micro level of musical structure comprises two units — the smaller and incomplete *motive*, and the larger and complete *phrase*.

The mid-level of musical structure is the *section*. Phrases combine to form larger sections of musical structure. Sections are often much longer and punctuated by strong cadences. Longer songs and extended pieces of music are usually formed into two or more complete sections, while shorter songs or melodies may be formed of phrases and have no sectional structure.

At the macro-level of musical structure exists the complete work formed of *motives*, *phrases* and *sections*. Both phrases and sections are concluded with cadences; however, the cadence material at the end of a section is stronger and more conclusive in function.

These are the micro-, mid- and macro-levels of musical structure — *motives*, *phrases* and *sections* and the complete composition. This is the manner in which western music is conceptualized as structure. Almost all world musics are conceptualized in a similar manner.

Furthermore, if we look at the structure of pop song writing, typical songs consist of three sections:

1. The **Verse** contains the main story line of the song. It is usually four or eight lines in length. A song normally has the same number of lines in each verse. Otherwise, the song will not sound smooth. Most songs have two or three verses.

2. The **Chorus** states the core of the song. The title often appears in the first and/or last line of the chorus. The chorus is repeated at least once, and is usually the most memorable part of a song. It differs from the verse musically, and it may be of shorter or longer length than that of the verse.

3. A section called the **Bridge** is found is some, but not all songs. It has a different melody from either the Verse or the Chorus. It is often used instead of a third verse to break the monotony of simply repeating another verse.

Most songs contain two or three verses and a repeating chorus. Two common song forms are: Verse/Chorus/Verse/Chorus/Verse/Chorus and Verse/Chorus/Verse/Chorus/Bridge/Chorus.

In addition, a **refrain** is usually a two-line ending to a verse that contains the title or hook (the catchiest part of a song). In contrast, a chorus can stand alone as a verse on its own, while the refrain cannot - it needs the verse to first define the refrain.

PreChorus, is also referred to as a climb, lift, or build. It is used at the end of a verse and prior to the chorus. Its purpose is to musically and lyrically rise from the verse allowing tension to build until the song climaxes in to the chorus. Its length is usually one or two phrases.

In the following, we will focus on finding the section-level structure of music, though the hierarchical structure of music will also be explored at the end of this chapter. Letters A, B, C, etc., will be used to denote sections. For example, a musical piece may have a structure of ABA, indicating a three-part compositional form in which the second section contrasts with the first section, and the third section is a restatement of the first. In this chapter, we will not distinguish functions of different sections (e.g., verse or chorus), which will be addressed in the next chapter for music summarization.

4.2 Representations for Self-similarity Analysis

4.2.1 Distance Matrix

For visualizing and analyzing the recurrent structure of music, Foote (1999, 2000) proposed a representation called *self-similarity matrix*. Each cell in the matrix denotes the similarity between a pair of frames in the musical signal. Here, instead of using similarity, we will use distance between a pair of frames, which results in a *distance matrix*. Specifically, let $V = v_1 v_2 ... v_n$ denote the feature vector sequence of the original musical signal x. It means we segment x into overlapped frames x_i and compute the feature vector v_i of each frame (e.g., FFT, MFCC, chromagram). We then compute the distance between each pair of feature vectors according to some distance metric and obtain a matrix DM, which is the distance matrix. Thus,

$$DM(V) = [d_{ij}] = [\| v_i - v_j \|] \tag{4-1}$$

where $\| v_i - v_j \|$ denotes the distance between v_i and v_j.

Since distance is typically symmetric, i.e., $\| v_i - v_j \| = \| v_j - v_i \|$, the distance matrix is also symmetric. One widely used definition of distance between vectors is based on cosine distance:

$$\| v_i - v_j \| = 0.5 - 0.5 \cdot \frac{v_i \bullet v_j}{| v_i | | v_j |} \tag{4-2}$$

where we normalized the original definition of cosine distance to range from 0 to 1 instead of 1 to −1 to be consistent with the non-negative property of distance. Figure 4-1 shows an example of distance matrix using the chromagram feature representation and the distance metric based on Equation 4-2. One can easily see the diagonal lines in this plot, which typically correspond to repetitions (e.g., the beginning of the piece repeats itself from around frame 1000). However, not

all repetitions can be easily seen from this plot due to variations of the restatements: e.g., the beginning of the piece actually repeats again from around frame 2570 at a different key.

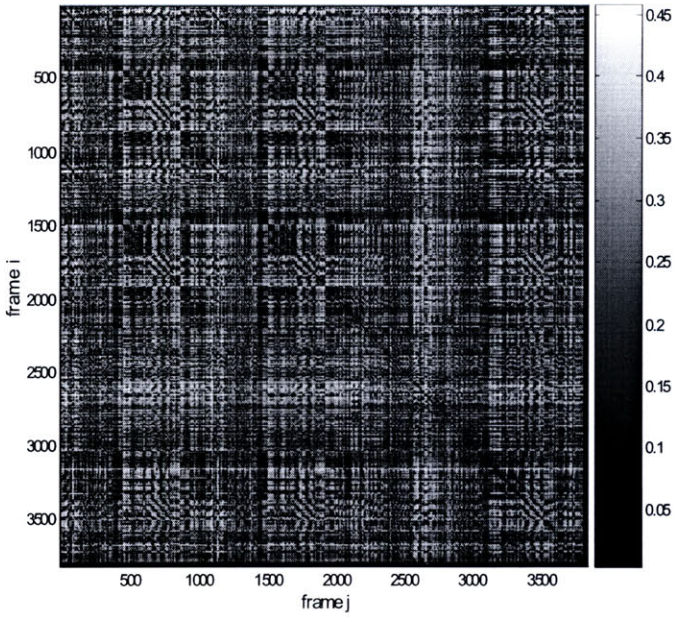


Figure 4-1: Distance matrix of "Mozart: Piano Sonata No. 15 In C".

4.2.2 Two Variations to Distance Matrix

Although the distance matrix is a good representation for analyzing the recurrent structure of general audio signals, the above example shows that one important property is ignored in this representation for analyzing musical signals: interval instead of absolute pitch is the one that most human listeners care about for recurrent structural analysis. For example, if a theme repeats at a different key, normal listeners can quickly adjust to the new key and recognize the repetition. However, with the distance matrix defined in Section 4.2.1, repetitions of this kind will not be effectively represented. Figure 4-1 showed one example of this.

We propose two variation representations to the distance matrix to solve this problem: Key-adjusted Distance Matrix (KDM) and Interval-based Distance Matrix (IDM).

KDM assumes that we know key change in advance, so that we can manipulate the feature vector to adjust to different keys within a musical piece. For example, if we use the 24-dimensional chromagram representation, we can rotate the chromagram vectors to make two vectors in the same key when computing the distance between them, i.e.,

$$KSM(V) = [d_{ij}] = [\| r(v_i, 2 \cdot (k_i - 1)) - r(v_j, 2 \cdot (k_j - 1)) \|] \quad (4\text{-}3)$$

where $r(v,k)$ was defined by Equation 3-8, and $\mathrm{K} = k_1 k_2 ... k_n$ denotes keys at different frames.

IDM does not assume that we know key change in advance. Instead, it attempts to capture the interval information between two consecutive frames. We will first convert $V = v_1 v_2 ... v_n$ into $U = u_1 u_2 ... u_{n-1}$:

$$u_i[j] = \| r(v_{i+1}, j) - r(v_i, 0) \| \qquad (4\text{-}4)$$

where j=1, 2, …, 24 and $r(v,k)$ was defined by Equation 3-8. Thus, u_i is a 24-dimensional vector whose component indexed by j denotes the distance between v_{i+1} and v_i after v_{i+1} is rotated by j. We then compute the distance matrix using U instead of V and obtain a (n-1)-by-(n-1) matrix, called IDM.

Figure 4-2 shows the two variations of distance matrix for the same piece in Figure 4-1.

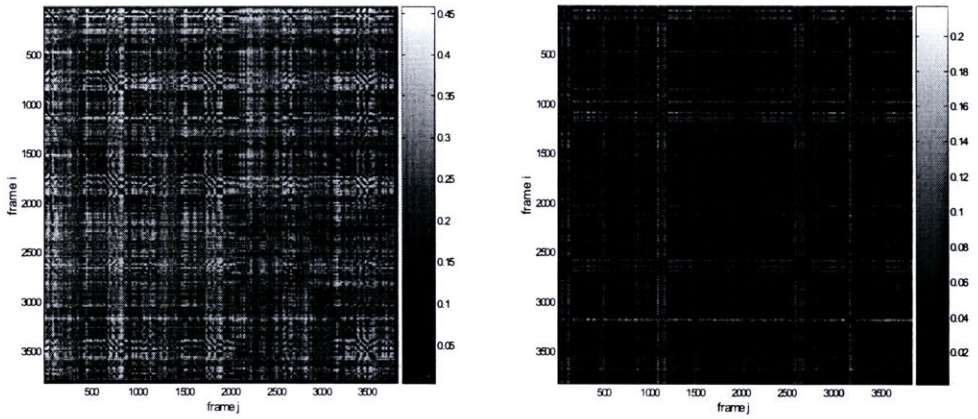


Figure 4-2: Two variations to the distance matrix of "Mozart: Piano Sonata No. 15 In C" (left: KDM; right: IDM).

If we zoom in these plots (Figure 4-3) and look at the patterns from around frame 2570, we will be able to visualize the diagonal lines from the two variations, which correspond to a repetition at a different key. This could not be seen from the original distance matrix representation.

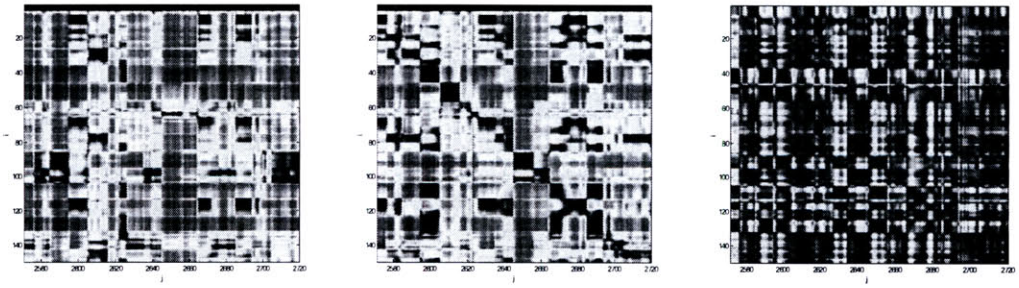


Figure 4-3: Zoom in of the last repetition in "Mozart: Piano Sonata No. 15 In C" (left: DM; middle: KDM; right: IDM)

4.3 Dynamic Time Warping for Music Matching

The above section showed that when part of the musical signal repeats itself nearly perfectly with key adjustment, diagonal lines appear in the distance matrix or its variation representations. However, if the repetitions have various variations (e.g., tempo change, different lyrics), which are very common in all kinds of music, the diagonal patterns will not be obvious. One solution is to consider approximate matching based on the self-similarity representation to allow flexibility of

repetitions, especially tempo flexibility. Dynamic time warping was widely used in speech recognition for similar purposes. Previous research has shown that it is also effective for music pattern matching (Yang, 2001). Note that dynamic time warping is often mentioned in the context of speech recognition, where similar technique is cited as dynamic programming for approximate string matching, and the distance between two strings based on it is often called edit distance.

Assume we have two sequences and we need to find the match between the two sequences. Typically, one sequence is the input pattern ($U = u_1 u_2 ... u_m$) and the other ($V = v_1 v_2 ... v_n$) is the one in which to search for the input pattern. Here, we allow multiple appearances of pattern U in V. Dynamic time warping utilizes dynamic programming approach to fill in an m-by-n matrix WM based on Equation 4-5. The initial condition (i=0 or j=0) is set based on Figure 4-4.

$$DM[i,j] = \min \begin{cases} DM[i-1,j] + c_D[i,j], & (i \geq 1, j \geq 0) \\ DM[i,j-1] + c_I[i,j], & (i \geq 0, j \geq 1) \\ DM[i-1,j-1] + c_S[i,j], & (i,j \geq 1) \end{cases} \quad (4\text{-}5)$$

where c_D is cost of deletion, c_I is cost of insertion, and c_S is cost of substitution. The definitions of these parameters are determined differently for different applications. For example, we can define

$$c_S[i,j] = \| u_i - v_j \|$$

$$c_D[i,j] = c_I[i,j] = 1.2 \cdot c_S[i,j]$$

to penalize insertion and deletion based on the distance between u_i and v_j. We can also define c_D and c_I to be some constant.

		v_1	v_2	v_3	⋮	⋮	⋮	⋮	⋮	v_n
	0	0	0	0	...	...	...	...	...	0
u_1	e									
u_2	2e									
...	...									
...	...									
u_m	*m* e									

Figure 4-4: Dynamic time warping matrix WM with initial setting. *e* is a pre-defined parameter denoting the deletion cost.

The last row of matrix WM (highlighted in Figure 4-4) is defined as a matching function $r[i]$ (i=1, 2, …, n). If there are multiple appearances of pattern U in V, local minima corresponding to these locations will occur in $r[i]$. We can also define the overall cost of matching U and V (i.e., edit distance) to be the minimum of $r[i]$, i.e., $\| U - V \|_{DTW} = \min_i \{r[i]\}$. In addition, to find the locations in V that match pattern U we need a trace-back step. The trace-back result is denoted as a trace-back function $t[i]$ recording the index of the matching point. Consider the following

example of matching two strings: U=bcd, V=abcdefbcedgbdaabcd, e=1, $c_D[i,j] = c_I[i,j] = 1$, $c_S[i,j] = 0$ if $u_i = v_j$, $c_S[i,j] = 1$ if $u_i \neq v_j$, substitution has the priority for tracing-back. Figure 4-5 shows the dynamic time warping matrix WM, the matching function $r[i]$ and the trace-back function $t[i]$.

		a	b	c	d	e	f	b	c	e	d	g	b	d	a	a	b	c	d
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	1	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1
c	2	2	1	0	1	2	2	2	0	1	2	2	1	1	2	2	1	0	1
d	3	3	2	1	0	1	2	3	1	1	1	2	2	1	2	3	2	1	0
r[i]	3	3	2	1	0	1	2	3	1	1	1	2	2	1	2	3	2	1	0
t[i]	0	1	2	2	2	2	2	5	7	7	7	7	12	12	12	12	16	16	16

Figure 4-5: An example of the dynamic time warping matrix WM, the matching function $r[i]$ and the trace-back function $t[i]$.

The time complexity of dynamic time warping is $O(nm)$, corresponding the computation needed for filling up matrix WM.

4.4 Recurrent Structure Analysis

This section presents an algorithm that will output structural information, including both the form (e.g., AABABA) and the boundaries indicating the beginning and the end of each section. Section 4.4.1 describes a clustering-based method for identifying musical form given segmentation of a piece, while Section 4.4.2 assumes that no prior knowledge about the musical form or the length of each section is provided. In any case, the restatement of a section may have variations (e.g., different lyrics, tempo, etc.). This assumption requires both robustness and efficiency of the algorithm.

4.4.1 Identification of Form Given Segmentation

This section presents a method for identifying musical form given segmentation of the piece. Specifically, assume we know a piece has N sections with boundaries denoting the beginning and the end of each section, i.e., each section is $U_i = v_{p_i} v_{p_i+1} ... v_{q_i}$ (i=1, 2, ..., N). We want to find a method to label the form of the piece (e.g., AABABA).

This problem can be modeled as a clustering problem: if we also know there are k different sections, we simply need to group the N sections into k groups, with different labels for different groups.

In general, there are three approaches for clustering sequences:

1. The model-based approach assumes each sequence is generated by some underlining stochastic process; thus, the problem can be changed to estimate the underlining model from the sequence and cluster the models.

2. The feature-based approach represents each sequence by a feature vector; thus, the problem can be changed to standard clustering problem of points.

3. The distance-based approach attempts to define some distance metric between sequences and use hierarchical agglomerative clustering to cluster the sequences. A hierarchical agglomerative clustering procedure produces a series of partitions of the data. At each particular stage the method merges the two clusters which are closest to each other (most similar).

Here, since it is very natural to define distance between sequences, we will employ the third method for this task. Thus, the major work here is to define the distance between each pair of sections, which results in a N-by-N distance matrix.

Distance between each pair of sections can be defined based on their edit distance as follows:

$$\| U_i - U_j \| = \frac{\| U_i - U_j \|_{DTW}}{\min\{\| U_i \|, \| U_j \|\}} \tag{4-6}$$

Two clustering techniques were explored:

1. Hierarchical agglomerative clustering: Since the distance matrix can be obtained, it is straightforward to use hierarchical clustering to get a cluster tree and cut it at a proper level based on the number of clusters for obtaining the final result.

2. K-means clustering: We also tried another method of clustering sequences based on the distance matrix. First, we used multidimensional scaling to map the sequences into points in a Euclidean space based on the distance matrix. Then, K-means clustering was applied for clustering these points.

   Multidimensional scaling (MDS) is a collection of methods to provide a visual representation of the pattern of proximities (i.e., similarities or distances) among a set of objects. Among the collection of methods, classical multidimensional scaling (CMDS) will be employed, whose metric scaling is based on the Singular Value Decomposition (SVD) of the double-centered matrix with Euclidean distances. (Kruskal, 1977)

4.4.2 Recurrent Structural Analysis without Prior Knowledge

This section presents a method for recurrent structural analysis without assuming any prior knowledge. This means we need to detect the recurrent patterns and boundaries of each section at the same time. Assuming that we have computed the feature vector sequence and the distance matrix DM (or either variation of it; in the following, we will simply use DM without mentioning the possibility of using its variation form), the algorithm follows four steps, which are illustrated in Figure 4-6:

1. Segment the feature vector sequence (i.e., $V = v_1 v_2 ... v_n$) into overlapped segments of fixed length l (i.e., $S = S_1 S_2 ... S_m$; $S_i = v_{k_i} v_{k_i+1} ... v_{k_i+l-1}$) and compute the repetitive property of each segment S_i by matching S_i against the feature vector sequence starting from S_i (i.e., $V_i = v_{k_i} v_{k_i+1} ... v_n$) using dynamic time warping based on DM. Thus, we can get a dynamic time warping matrix DM_i for each segment S_i;

2. Detect the repetitions of each segment S_i by finding the local minima in the matching function $r_i[j]$ of the dynamic time warping matrix DM_i obtained from step 1;

3. Merge consecutive segments that have the same repetitive property into sections and generate pairs of similar sections;

4. Segment and label the recurrent structure including the form and boundaries.

The following four sections explain each step in detail. All the parameter configurations are tuned based on the representation presented in the previous sections, and the experimental corpus that will be described in Section 4-6.

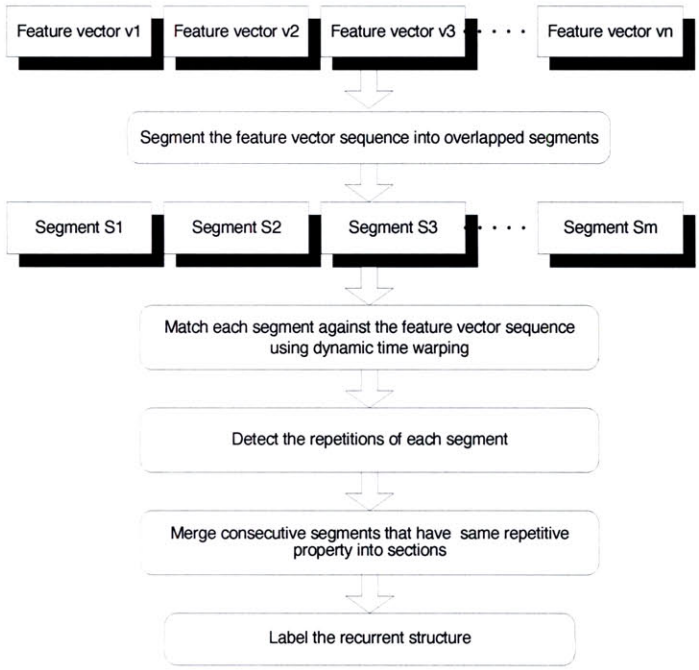


Figure 4-6: Analysis of recurrent structure without prior knowledge.

4.4.2.1. Pattern Matching

In the first step, we segment the feature vector sequence (i.e., $V = v_1 v_2 ... v_n$) into overlapped segments of fixed length l (i.e., $S = S_1 S_2 ... S_m$; $S_i = v_{k_i} v_{k_i+1} ... v_{k_i+l-1}$; e.g., 200 consecutive vectors with 150 vectors overlap) and compute the repetitive property of each segment S_i by matching S_i against the feature vector sequence starting from S_i (i.e., $V_i = v_{k_i} v_{k_i+1} ... v_n$) using dynamic time warping. We define the cost of substitution c_S to be the distance between each pair of vectors. It can be obtained directly from the distance matrix DM. We also define the costs of deletion and insertion to be some constant: $c_D[i, j] = c_I[i, j] = a$ (e.g., $a = 0.7$). For each matching between S_i and V_i, we obtain a matching function $r_i[j]$.

4.4.2.2. Repetition Detection

This step detects the repetition of each segment S_i. To achieve this, the algorithm detects the local minima in the matching function $r_i[j]$ for each i, because typically a repetition of segment S_i will correspond to a local minimum in this function.

There are four predefined parameters in the algorithm of detecting the local minima: the width parameter w, the distance parameter d, the height parameter h, and the shape parameter p. To detect local minima of $r_i[j]$, the algorithm slides the window of width w over $r_i[j]$. Assume the index of the minimum within the window is j_0 with value $r_i[j_0]$, the index of the maximum within the window but left to j_0 is j_1 (i.e., $j_1<j_0$) with value $r_i[j_1]$, and the index of the maximum within the window but right to j_0 is j_2 (i.e., $j_2>j_0$) with value $r_i[j_2]$. If the following conditions are all satisfied:

(1) $r_i[j_1]-r_i[j_0]>h$ *and* $r_i[j_2]-r_i[j_0]>h$ (i.e., the local minimum is deep enough);

(2) $\frac{r_i[j_1]-r_i[j_0]}{j_1-j_0}>p$ *or* $\frac{r_i[j_2]-r_i[j_0]}{j_2-j_0}>p$ (i.e., the local minimum is sharp enough);

(3) No two repetitions are closer than d,

then the algorithm adds the minimum into the detected repetition set. Figure 4-7 shows the repetition detection result of a particular segment for Beatles song *Yesterday*.

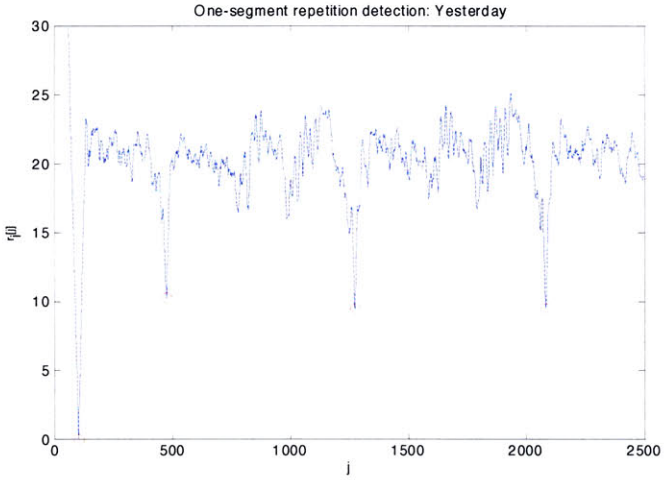


Figure 4-7: One-segment repetition detection result of Beatles song *Yesterday*. The local minima indicated by circles correspond to detected repetitions of the segment.

In Figure 4-7, the four detected local minima correspond to the four restatements of the same melodic segment in the song ("Now it looks as though they are here to stay …", "There is a shadow hanging over me …", "I need a place to hide away …", "I need a place to hide away …"). However, the repetitions detected may have add- or drop-errors, meaning a repetition is falsely detected or missed. The number of add-errors and that of the drop-errors are balanced by the predefined parameter h; whenever the local minimum is deeper than height h, the algorithm reports a detection of repetition. Thus, when h increases, there are more drop-errors but fewer add-errors, and vise versa. For balancing between these two kinds of errors, the algorithm can search within a range for the best value of h, so that the number of detected repetitions of the whole song is reasonable (e.g., $\#\ total\ detected\ repetitions / n \approx 2$).

For each detected minimum $r_i[j^*]$ for S_i, let $k^* = t_i[j^*]$; thus, it is detected that segment $S_i = v_{k_i} v_{k_i+1} \ldots v_{k_i+l-1}$ is repeated in V from $v_{k_i+k^*}$. Note that by the nature of dynamic programming, the matching part in V may not have length l due to the variations in the repetition.

4.4.2.3. Segment Merging

This step merges consecutive segments that have the same repetitive property into sections and generates pairs of similar sections.

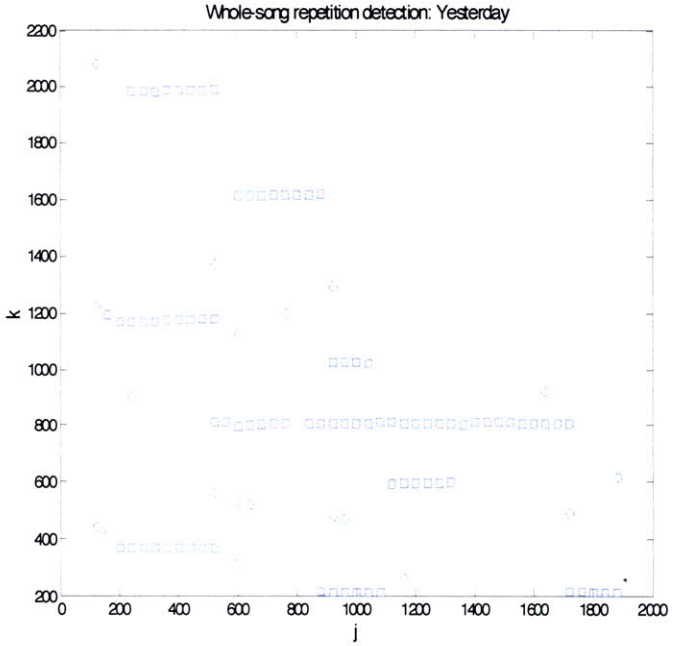


Figure 4-8: Whole-song repetition detection result of Beatles song *Yesterday*. A circle or a square at location (j, k) indicates that the segment starting from v_j is detected to repeat from v_{j+k}. Horizontal patterns denoted by squares correspond to detected section repetitions.

Figure 4-8 shows the repetition detection result of the Beatles song Yesterday after this step. In this figure, a circle or a square at (j, k) corresponds to a repetition detected in the last step (i.e., the segment starting from v_j is repeated from v_{j+k}). Since typically one musical phrase consists of multiple segments, based on the configurations in previous steps, if one segment in a phrase is repeated by a shift of k, all the segments in this phrase are repeated by shifts roughly equal to k. This phenomenon can be seen from Figure 4-8, where the squares form horizontal patterns indicating consecutive segments have roughly the same shifts.

By detecting these horizontal patterns (denoted by squares in Figure 4-8) and discarding other detected repetitions (denoted by circles in Figure 4-8), add- or drop-errors in repetition detection are further reduced.

The output of this step is a set of sections consisting of merged segments and the repetitive relation among these sections in terms of section-repetition vectors $[j_1\ j_2\ shift_1\ shift_2]$, indicating that the segment starting from v_{j_1} and ending at v_{j_2} repeats roughly from $v_{j_1+shift_1}$ to

$v_{j_2+shift_2}$. Each vector corresponds to one horizontal pattern in the whole-song repetition detection result. For example, the vector corresponding to the left-bottom horizontal pattern in Figure 4-8 is [200 520 370 370].

4.4.2.4. Structure Labeling

Based on the vectors obtained from the third step, the last step of the algorithm segments the whole piece into sections and labels each section according to the repetitive relation (i.e., gives each section a symbol such as "A", "B", etc.). This step will output the structural information, including both the form (e.g., AABABA) and the boundaries indicating the beginning and the end of each section.

To solve conflicts that might occur, the rule is to always label the most frequently repeated section first. Specifically, the algorithm finds the most frequently repeated section based on the first two columns in the section-repetition vectors, and labels it and its shifted versions as section A. Then the algorithm deletes the vector already labeled, repeats the same procedure for the remaining section-repetition vectors, and labels the sections produced in each step as B, C, D and so on. If conflicts occur (e.g., a later labeled section has overlap with the previous labeled sections), the previously labeled sections will always remain intact, and the currently labeled section and its repetition will be truncated, so that only the unoverlapped part will be labeled as new.

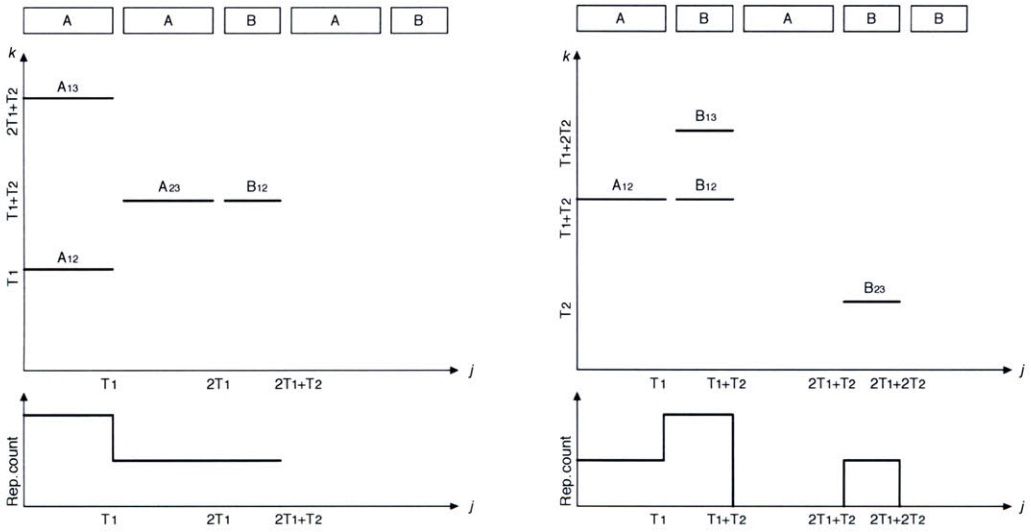


Figure 4-9: Idealized whole-song repetition detection results (left: form AABAB; right: form ABABB). Section A's are assumed to be of length T_1 and Section B's are assumed to be of length T_2. The bottom figures show the number of horizontal patterns that contain v_j for each *j*.

To illustrate how the algorithm works, two idealized examples are shown in Figure 4-9. In the first example with form AABAB, the section-repetition vectors obtained from the last step should be {[1 T_1 T_1 T_1>, [T_1 $2T_1+T_2$ T_1+T_2 T_1+T_2], [1 T_1 $2T_1+T_2$ $2T_1+T_2$]} corresponding to the three horizontal patterns A_{12}, $A_{23}B_{12}$ and A_{13} respectively. In the second example with form ABABB, the section-repetition vectors obtained from the last step should be {[$2T_1+T_2$ $2T_1+2T_2$ T_2 T_2], [1 T_1+T_2 T_1+T_2 T_1+T_2], [T_1 T_1+T_2 T_1+2T_2 T_1+2T_2]} corresponding to the three horizontal patterns B_{23}, $A_{12}B_{12}$ and B_{13} respectively. The structure labeling process is as follows:

1) Initialize X to be section symbol "A".

2) Find the most frequently repeated part by counting the number of horizontal patterns that contain v_j for each j. The bottom figures in Figure 4-9 show that the most frequently repeated part is [1 T_1] for the first example and [T_1 T_1+T_2] for the second example.

3) For each section-repetition vector [j_1 j_2 $shift_1$ $shift_2$] that contains the most frequently repeated part, label [j_1 j_2] and [$j_1 + shift_1$ $j_2 + shift_2$] as section X. If either one has an overlap with previously labeled sections, truncate both of them and label them in a way that is consistent with previous labels.

4) Delete the section-repetition vector that was just labeled. Let X be the next section symbol, e.g., "B", "C", "D", etc.

5) If there are unprocessed section-repetition vectors, go to 2).

In the above structure labeling process, two problems exist. The first, again, is how to solve a conflict, which means a later labeled section may have overlap with previously labeled sections. The rule is the previously labeled sections will always remain intact and the current section will be truncated. Only the longest truncated unoverlapped part, if it is long enough, will be labeled as a new section. The shifted version will be truncated accordingly as well, even if there is no conflict, to resemble the structure of its original version. In the first idealized example, the first loop of the algorithm will process vector [1 T_1 T_1 T_1] and [1 T_1 $2T_1$+T_2 $2T_1$+T_2] and label the three "A" sections. The second loop will process vector [T_1 $2T_1$+T_2 T_1+T_2 T_1+T_2]. Since conflicts occur here, the two "B" sections are generated by truncating the original and the shifted version of it.

The second problem is how to choose the order of processing section-repetition vectors in each loop. In the first example, the order of processing the two section-repetition vectors in the first loop will not affect the structure labeling result. However, in the second example, the order of first processing section-repetition vector [1 T_1+T_2 T_1+T_2 T_1+T_2] or [T_1 T_1+T_2 T_1+$2T_2$ T_1+$2T_2$] will change the result. If we choose to process section-repetition vector [T_1 T_1+T_2 T_1+$2T_2$ T_1+$2T_2$] first, the first and the third "B" sections will be labeled at the beginning. Next when we process section-repetition vector [1 T_1+T_2 T_1+T_2 T_1+T_2], the original version will be truncated to generate the first "A" section. The shifted version will resemble its original version, generating the second "A" section and the second "B" section. In this case, the structure labeling result is exactly "ABABB". On the other hand, if we choose to process section-repetition vector [1 T_1+T_2 T_1+T_2 T_1+T_2] first, the algorithm will label "AB" together as "A*". When we next process section-repetition vector [T_1 T_1+T_2 T_1+$2T_2$ T_1+$2T_2$], a conflict occurs and no new section is generated. The shifted version will be labeled as section "A*" as well. In this case, the structure labeling result is "AAA" (Figure 4-10).

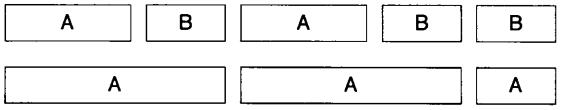


Figure 4-10: Different structure labeling results corresponding to different orders of processing section-repetition vectors in each loop.

In this idealized example, processing shorter section-repetition vectors first is preferred. However, the experiment shows that, due to the noisy data obtained from the previous steps, this order will result in small fragments in the final structure. Thus, the algorithm now is choosing the order based on the values of shifts (from small to large). This means, for structure like the second example, the output structure may combine "AB" together as one section. Labeling in this way also makes sense because it means we see the piece as repeating the section for three times with

the last restatement containing only part of the section. It will be discussed again in the context of a hierarchical structure in Section 4.8.

4.5 Evaluation Method

To qualitatively evaluate the results, figures as shown in Figure 4-11 are used to compare the structure obtained from the algorithm to the true structure obtained by manually labeling the repetitions. We will also use metrics of structural similarity to quantitatively evaluate the result.

Figure 4-11: Comparison of the computed structure using DM (above) and the true structure (below) of *Yesterday*. Sections in the same color indicate restatements of the section. Sections in the lightest gray correspond to the parts with no repetition.

Same metrics as in Chapter 3, including *label accuracy* (Equation 3-9), *precision* (Equation 3-10) and *recall* (Equation 3-11), will be used here for quantitatively evaluating the segmentation performance. In addition, one extra metric - *formal distance* - will be used to evaluate the difference between the computed form and the true form. It is defined as the edit distance between the strings representing different forms. For example, the formal dissimilarity between structure AABABA and structure AABBABBA is 2, indicating two insertions from the first structure to the second structure (or, two deletions from the second structure to the first structure; thus this definition of distance is symmetric). Note that how the system labels each section is not important as long as the repetitive relation is the same; thus, structure AABABA is deemed as equivalent (0-distance) to structure BBABAB, or structure AACACA.

4.6 Experiments and Results

Two experimental corpora were tested. One corpus is piano music same as the one used in Chapter 3. The other consists of the 26 Beatles songs in the two CDs *The Beatles* (1962-1966). All of these musical pieces have clear recurrent structures, so that the true recurrent structures were labeled easily for comparison. The data were mixed into 8-bit mono and down-sampled to 11kHz.

4.6.1 Performance: Identification of Form Given Segmentation

We tried three different self-similarity representations: DM, KDM and IDM, where KDM was obtained either by manually labeled key structure or computed key structure using the approach presented in Chapter 3. Thus, two clustering techniques (hierarchical clustering and k-means clustering) and four forms of distance matrix (DM, IDM, computed KDM and labeled KDM) were investigated. Figure 4-12 shows the performance in terms of average formal distance over all pieces of each corpus. For both corpora, the performance is fairly good using either clustering technique and any self-similarity representation of DM, computed KDM or labeled KDM. Especially, the average formal distance is 0 (i.e., the computed forms of all pieces are identical to the truth) using the combination of k-means clustering and labeled KDM representation. This suggests that, with the key-adjusted representation, the repetitions at different keys can be captured fairly well.

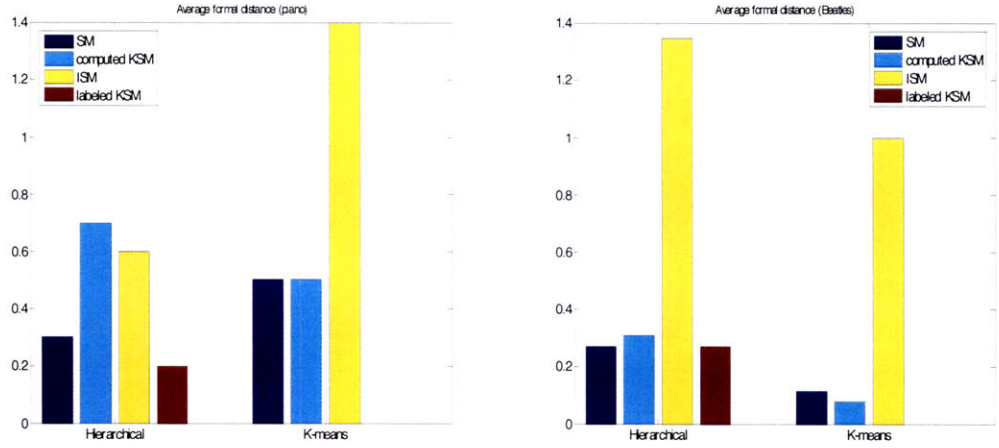
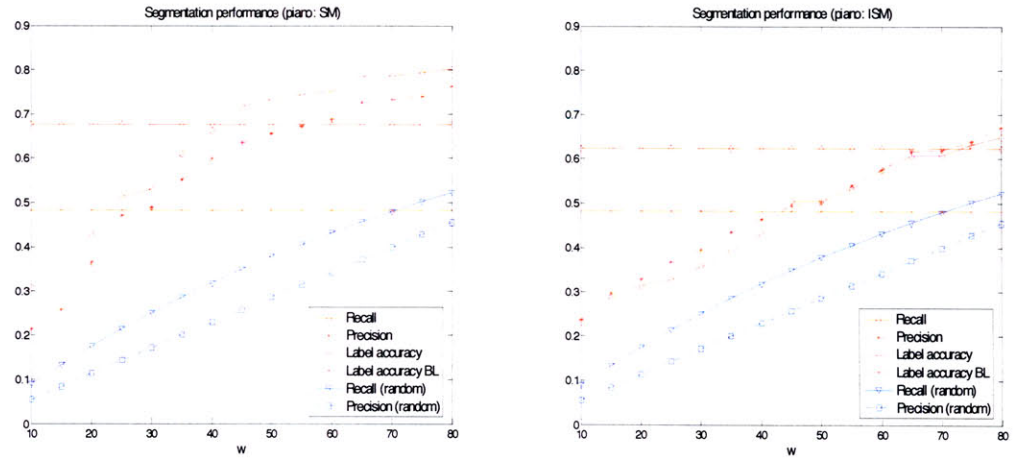


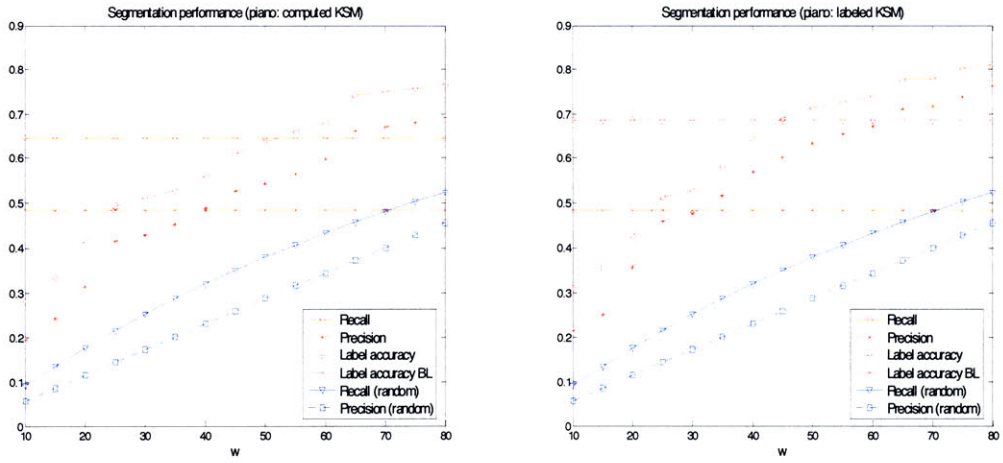
Figure 4-12: Formal distance using hierarchical and K-means clustering given segmentation (left: piano; right: Beatles).

4.6.2 Performance: Recurrent Structural Analysis without Prior Knowledge

In this experiment, for the first corpus, we tried all the four forms of distance matrix (DM, IDM, computed KDM and labeled KDM); for the second corpus, only DM was used, because key change rarely happened in this data set. Figure 4-13 and 4-14 show the segmentation performances of the two data corpora, respectively. Similar to Chapter 3, the x-axis denotes *w* varying from 10 frames (~0.46s) to 80 frames (~3.72s) for calculating recall and precision. In each plot, the bottom two curves correspond to upper bounds of recall and precision based on random segmentation. The bottom horizontal line shows the baseline label accuracy of labeling the whole piece as one section.



(a) (b)



(c) (d)

Figure 4-13: Segmentation performance of recurrent structural analysis on classical piano music (a: DM; b: IDM; c: computed KDM; d: labeled KDM).

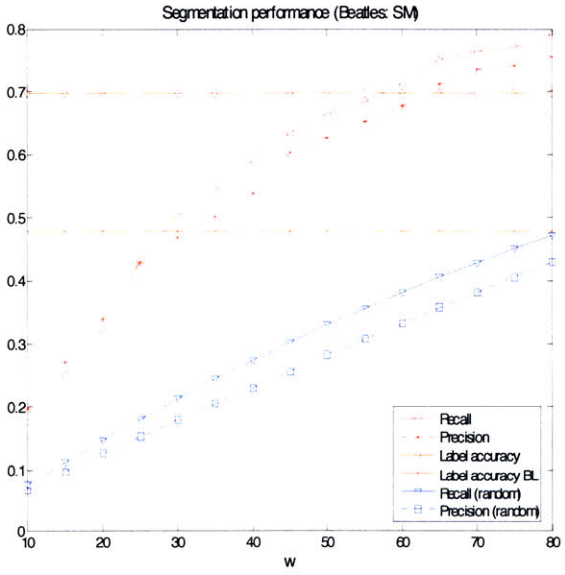


Figure 4-14: Segmentation performance of recurrent structural analysis on Beatles songs.

Figure 4-15 and 4-16 show the piece-by-piece performance of the two data corpora, respectively, including formal distance. Segmentation performance was evaluated at w=40.

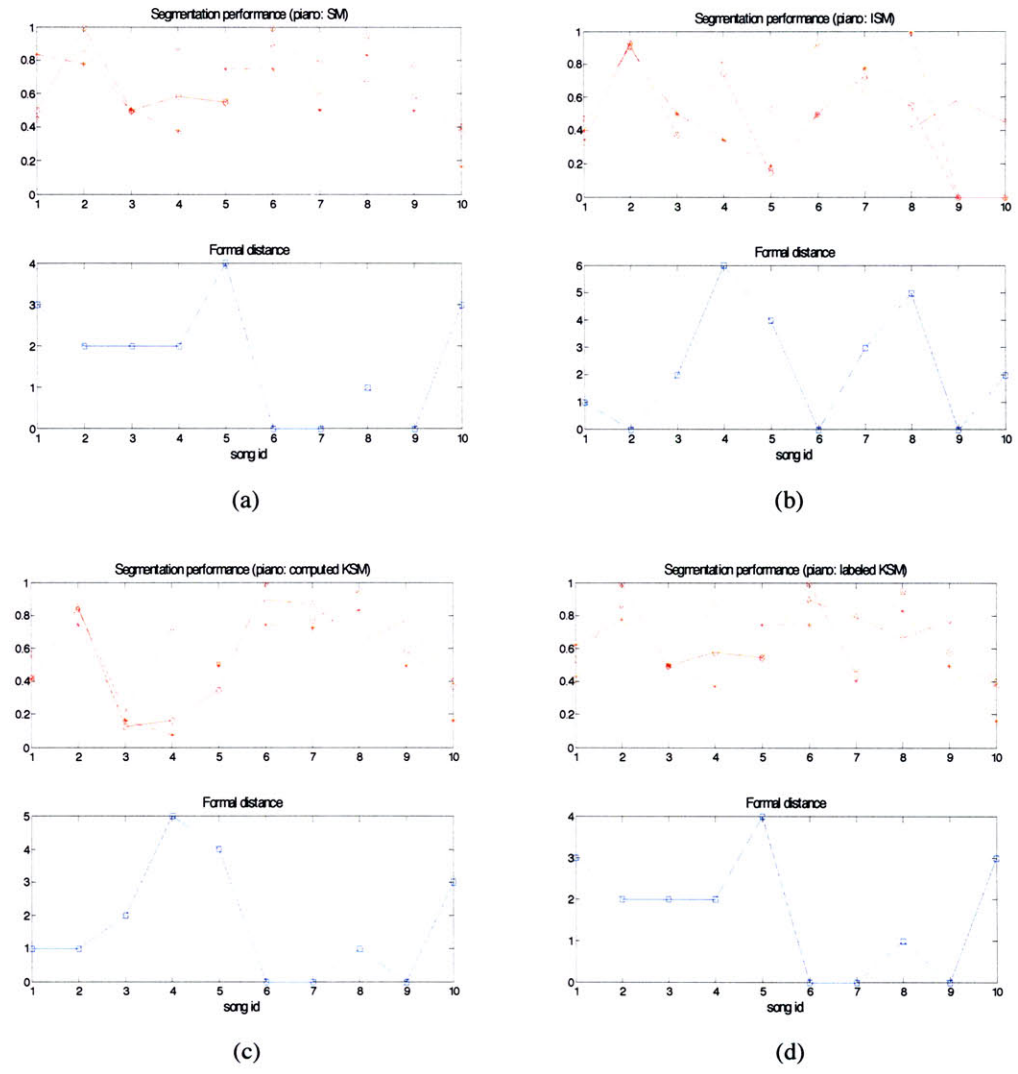


Figure 4-15: Segmentation performance and formal distance of each piano piece (w=40; a: DM; b: IDM; c: computed KDM; d: labeled KDM).

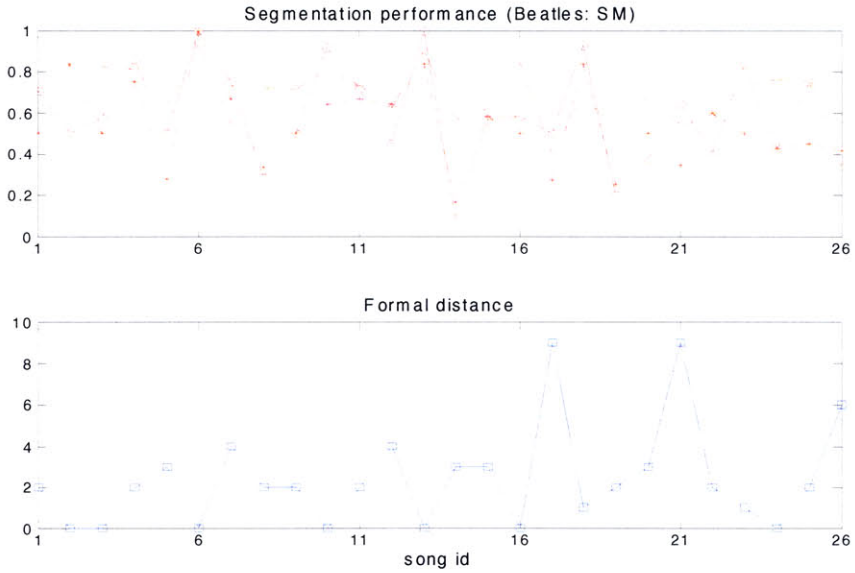
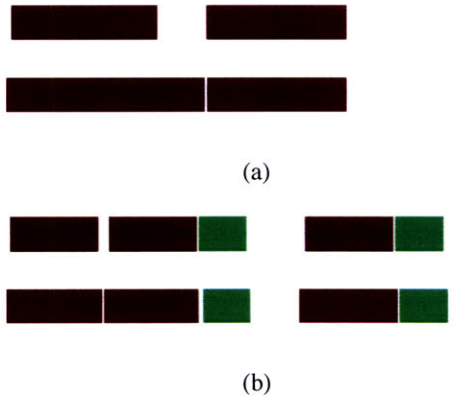


Figure 4-16: Segmentation performance and formal distance of each Beatles song (w=40).

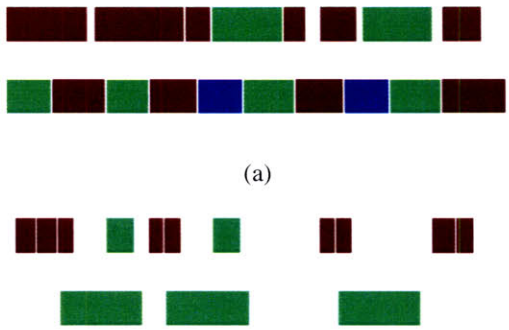
The performance on each piece is clearly illustrated by the above two figures. For example, the performance of the song Yesterday (the thirteenth song of Beatles; Figure 4-11) is: recall=1, precision=0.83, label accuracy=0.9, formal distance=0. Other examples with good performances are shown in Figure 4-17.



(a)

(b)

Figure 4-17: Comparison of the computed structure (above) and the true structure (below). (a) 6th piano piece *Chopin: Etude In E, Op. 10 No. 3 'Tristesse'* using DM; (b) 6th Beatles song *All my loving* using DM.



(a)

(b)

Figure 4-18: Comparison of the computed structure (above) and the true structure (below). (a) 5th piano piece *Paderewski: Menuett* using IDM (b) 17th Beatles song *Day tripper* using DM.

One interesting thing is, if we listen to the Beatles piece shown in Figure 4-18, the detected repeating sections actually correspond to repeating guitar solo patterns without vocal, while the truth was labeled by the author only based on the vocal part (the verse/chorus segmentation of lyrics from the web).

4.7 Discussion

The experimental result shows that, by DM or labeled KDM, the performance of 7 out of 10 piano pieces and 17 out of 26 Beatles songs have formal distances less than or equal to 2 (Figure 4-15 and 4-16). The label accuracy is significantly better than the baseline (Figure 4-13 and 4-14) and the segmentation performance is significantly better than random segmentation. This demonstrates the promise of the method.

Comparing the four forms of distance matrix, it is not so surprising that DM and labeled KDM worked the best with labeled KDM being slightly better. Labeled KDM worked slightly better because it considers key adjustment and can better capture repetitions at different keys; however, since repetitions at different keys do not happen often, the improvement is not obvious. Computed KDM did not work as well as labeled KDM because the label accuracy was not 100% accurate. IDM seems not able to capture the interval information well as we expected.

We also found that the computed boundaries of each section were often slightly shifted from the true boundaries. This was mainly caused by the inaccuracy of the approximate pattern matching. To tackle this problem, other musical features (e.g., chord progressions, change in dynamics) should be used to detect local events so as to locate the boundaries accurately. In fact, this suggests that computing only the repetitive relation might not be sufficient for finding the semantic structure. According to Balaban (1992),

> *The position of phrase boundaries in tonal melodies relates to a number of interacting musical factors. The most obvious determinants of musical phrases are the standard chord progressions known as cadence. Other factors include 'surface features' such as relatively large interval leaps, change in dynamics, and micropauses ('grouping preference rules'), and repeated musical patterns in terms of harmony, rhythm and melodic contour.*

In the result of Beatles song Eleanor Rigby (Figure 4-19), section "B" in the true structure splits into two sections "BB" due to an internal repetition (i.e., a phrase repeats right after itself) within

the "B" section. This split phenomenon happens in many cases of the corpus due to the internal repetitions not shown in the true structure.

Figure 4-19: Comparison of the computed structure (above) and the true structure (below) of the 25th Beatles song Eleanor Rigby using DM.

It also happens in some cases where several sections in the true structure merge into one section. For example, for Beatles song Help (Figure 4-20), section A in the computed structure can be seen as the combination of section A and B in the true structure. The merge phenomenon might be caused for three reasons:

1) No clue of repetition for further splitting. Figure 4-20 shows an example of this case. There is no reason to split one section into two sections "AB" as in the true structure only based on the repetitive property.

2) Deficiency of structural labeling. Figure 4-10 shows an example of this case.

3) Parameters in the algorithm are set in such a way that short-phrase repetitions are ignored.

Figure 4-20: Comparison of the computed structure (above) and the true structure (below) of the 14th Beatles song *Help!* using DM.

The split/merge phenomena suggest we further explore the hierarchical structure of music as the output of structural analysis and also evaluate the result considering the hierarchical similarity, which will be explained in the next section.

4.8 Generation and Comparison of Hierarchical Structures

Musical structure is hierarchical, and the size of the grain would have to vary from finer than a single sound to large groupings of notes, depending upon composed relationships. Listening to music is an active hierarchic process; therefore, what we hear (understand) will depend upon both the composed relationships and the grain of our listening (Erickson, 1975). A theory of the grouping structure was developed by Lerdahl (1983):

> *The process of grouping is common to many areas of human cognition. If confronted with a series of elements or a sequence of events, a person spontaneously segments or "chunks" the elements or events into groups of some kind. ... For music the input is the raw sequences of pitches, attack points, durations, dynamics, and timbres in a heard piece. When a listener has constructed a grouping structure for a piece, he has gone a long way toward "making sense" of the piece: he knows what the units are, and which units belong together and which do not. ... The most fundamental characteristic of musical groups is that they are heard in a hierarchical fashion. A motive is heard as part of a theme, a theme as part of a theme-group, and a section as part of a piece.*

Therefore, inferring the hierarchical structures of music and identifying the functionality of each section within the structure is a more complicated yet interesting topic. Additionally, we need metrics for comparing similarity of hierarchical structures, which will make more sense for evaluating the result of recurrent structural analysis shown in Section 4.6.

4.8.1 Tree-structured Representation

The split/merge phenomena shown in Section 4.7 and the theory about music structure all suggest us to consider the hierarchical structures of music; one good representation is the tree-structure. Although, for a given piece of music, we might not have a unique tree representation, it is usually natural to find one tree most appropriate to represent its repetitive property in multiple levels. For example, one tree representation corresponding to song *Yesterday* is shown in Figure 4-21. The second level of the tree corresponds to the true structure shown in Figure 4-11. The third level denotes that there are four phrases in Section B, among which the first and the third are identical.

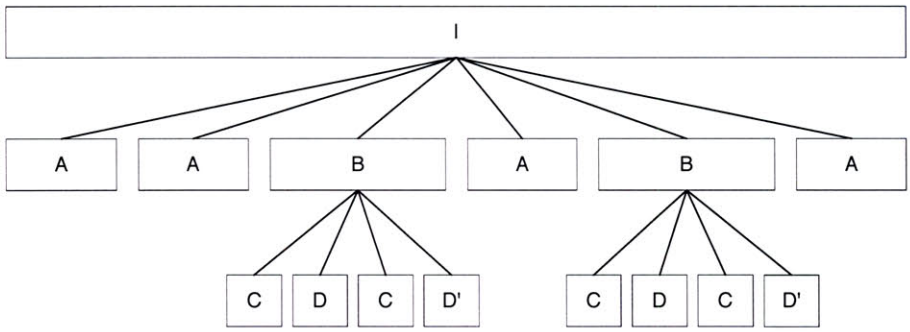


Figure 4-21: Tree representation of the repetitive structure of song *Yesterday*.

Inferring the hierarchical repetitive structures of music is apparently a more complicated yet interesting topic than the one-level structural analysis problem presented in the previous sections of this chapter. One possible solution is building the tree representation based on the one-level structural analysis result. Specifically, assuming we obtain the structural analysis result using the algorithm described above and it corresponds to a particular level in the tree, the task is how to build the whole tree structure similar to Figure 4-21 starting from this level. The algorithm can be divided into two processes: *roll-up process* and *drill-down process*. The roll-up process merges proper sections to build the tree structure up from this level to the top. The drill-down process splits proper sections to build the tree structure down from this level to the bottom.

4.8.2 Roll-up Process

Given the one-level structural analysis result, denoted as a section symbol sequence $Y=X_1X_2X_3...X_N$, where X_i is a section symbol such as "A", "B", "C", etc. Let S be the set of all the section symbols in Y. The roll-up process can be defined as follows:

1. Find substring Y_s ($|Y_s|>1$) of Y, such that, if all the unoverlapped substring Y_s's in Y are substituted by a new section symbol X_w, at least one symbol in S will not appear in the new Y.

2. Let S be the set of all the section symbols in the new Y. If $|Y|>1$, go to 1.

This algorithm iteratively merges sections in each loop, which corresponds to a particular level in the tree structure. Note, however, the algorithm is not guaranteed to give a unique solution. For example, Figure 4-22 shows two possible solutions corresponding to two different trees, given the

initial Y=AACDCD'ACDCD'A for song *Yesterday*. The first solution (left one) is consistent with the tree structure shown in Figure 4-21, while the second solution (right one) corresponds to an unnatural tree structure to represent the song. Therefore, the roll-up process can be seen as a search problem; how to build heuristic rules based on musical knowledge to search for the most natural path for merging sections needs to be explored in the future.

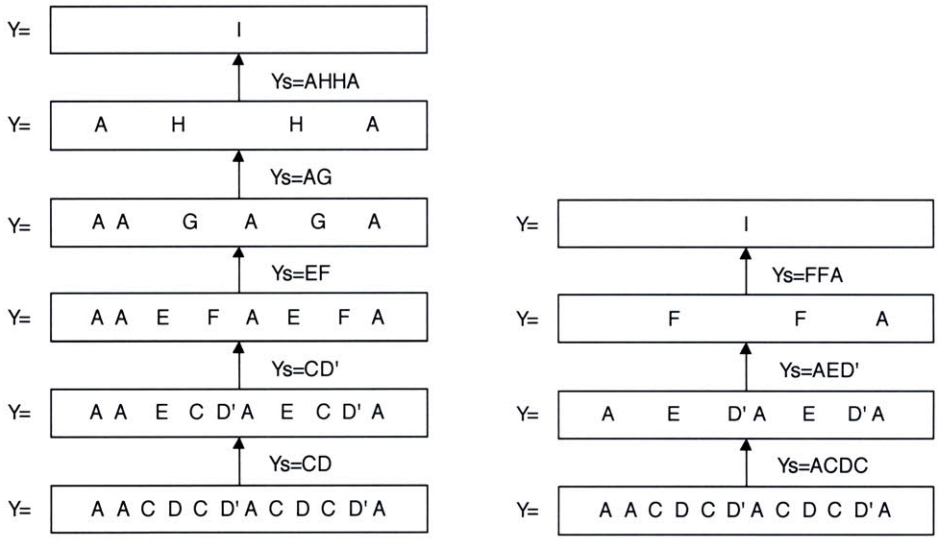


Figure 4-22: Two possible solutions of the roll-up process (from bottom to top) for song *Yesterday*.

4.8.3 Drill-down Process

The drill-down process is even algorithmically harder than the roll-up process. In the three reasons for the merge phenomenon shown in Section 4.7, the cases caused by the first two reasons may not be solved well without other musical cues (e.g., analysis of cadence), which does not have a straightforward solution. A solution to the cases caused by the third reason is using the one-level structural analysis algorithm again but focusing on each section instead of the whole piece to further split sections to explore the repetitive structure within the sections. To achieve this, parameters in the algorithm need to be tuned to serve the purpose of detecting short-segment repetitions.

4.8.4 Evaluation Based on Hierarchical Structure Similarity

In addition to generating the tree structure, the roll-up and drill-down processes can be used for comparing the structural similarity in a hierarchical context. The computed structure and the true structure might not be at the same level in the hierarchical tree, so comparing them directly as shown in Section 4.5 is not always reasonable. The one-level true structure labeled manually by humans might have bias as well. For example, when a short musical phrase repeats right after itself, we tended to label them as one section (a higher level in the tree) rather than two repeating sections (a lower level in the tree).

Thus, for the two examples shown by Figure 4-19 and 4-20, it would be unfair to compare the similarity between the computed structure and the true structure directly, because apparently the two structures are at different levels in the hierarchical tree. When the computed structure is at a lower level (e.g., AACDCD'ACDCD'A versus AABABA for *Yesterday*), there are splits; when

the computed structure is at a higher level (e.g., AABABA versus AACDCD'ACDCD'A for *Yesterday*), there are merges.

There are also cases where both splits and merges happen for one piece. Figure 4-23 gives an example of it, where the computed structure splits section A into two sections AA and merges two sections BB into one section B. If we evaluate the segmentation accuracy based on the one-level structure, the recall will be 6/8 and the precision will be 6/10. However, if we think about the structure in the hierarchical context, both structures make sense: if a musical phrase repeats right after itself, the two phrases might be seen as a refrain within one section or as two repeating sections. Therefore, it would be more meaningful to compare the two structures at the same level. For example, we can roll up both of the two structures to be ABA and thus get both recall and precision to be 1.

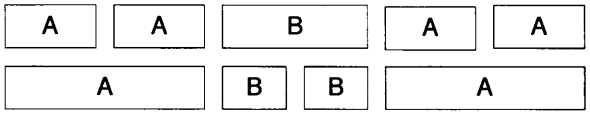


Figure 4-23: An example with both splits and merges involved.

Given the computed structure $Y = X_1 X_2 ... X_N$ and the true structure $\tilde{Y} = \tilde{X}_1 \tilde{X}_2 ... \tilde{X}_M$, the algorithm for comparing the similarity between these two structures by rolling-up them into the same level is as follows:

1. Roll up the computed structure Y:
   For each section $\tilde{X}_i$ in the true structure, if there are multiple sections $X_j X_{j+1} ... X_{j+k}$ in the computed structure that correspond to $\tilde{X}_i$ (i.e., the beginning of $\tilde{X}_i$ is roughly the beginning of X_j and the end of $\tilde{X}_i$ is roughly the end of X_{j+k}), then merge $X_j X_{j+1} ... X_{j+k}$ into one section. After this step, we obtain a new computed structure after roll-up, denoted as Y'.

2. Roll up the true structure $\tilde{Y}$:
   For each section X_i in the computed structure, if there are multiple sections $\tilde{X}_j \tilde{X}_{j+1} ... \tilde{X}_{j+k}$ in the true structure that correspond to X_i, then merge $\tilde{X}_j \tilde{X}_{j+1} ... \tilde{X}_{j+k}$ into one section. After this step, we obtain a new true structure after roll-up, denoted as $\tilde{Y}'$.

3. Compute the label accuracy, recall and precision using the new structures Y' and $\tilde{Y}'$.

Thus, the performance evaluated in this way will take split and merge phenomena into consideration and measure the similarity between two structures in the hierarchical context. Figure 4-24 and 4-25 show the performance measured in this way on the above two data corpora. Comparing to performance without considering the hierarchical structure (Figure 4-13 and 4-14), the result here is better, indicating split and merge phenomena did happen sometimes and the one-level evaluation could not capture them well.

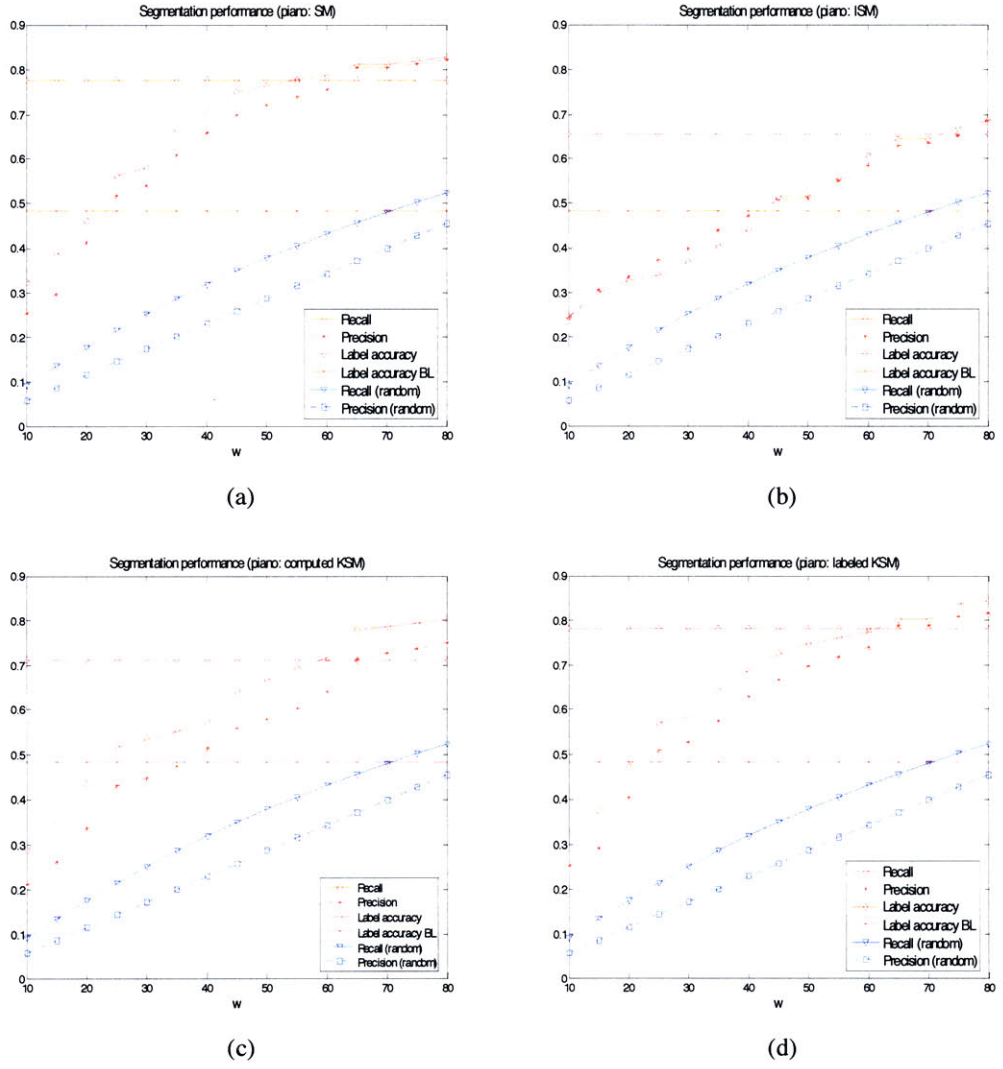


Figure 4-24: Segmentation performance of recurrent structural analysis based on hierarchical similarity for classical piano music (a: DM; b: IDM; c: computed KDM; d: labeled KDM).

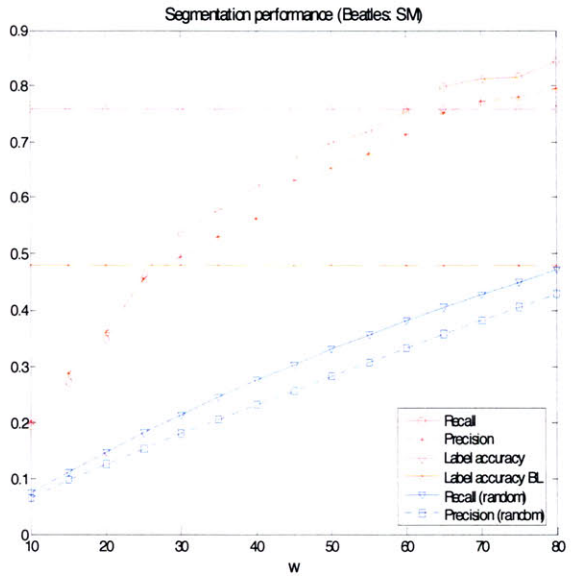


Figure 4-25: Segmentation performance of recurrent structural analysis based on hierarchical similarity for Beatles songs.

4.9 Summary

This chapter presented a method for automatically analyzing the recurrent structure of music from acoustic signals. Experimental results were evaluated both qualitatively and quantitatively, which demonstrated the promise of the proposed method.

The boundaries of the sections generated based on the result of structural analysis have significantly better consistency with musical transitions than boundaries produced by random. Although recurrent structural analysis is not sufficient for music segmentation by itself, it can be fused with other techniques (e.g., harmonic analysis described in the previous chapter) for local transition detection and musical phrase modeling to obtain good segmentation performance.

At the end of this chapter, we also proposed a framework towards hierarchical structural analysis of music and provided some preliminary results. Incorporating more musical knowledge might be helpful to make analysis of hierarchical structure more efficient. Methods for automatically tuning the parameters for different scales of repetition detection need to be developed as well.

Chapter 5 Structural Accentuation and Music Summarization

In the previous two chapters, the structures could be mostly inferred from the musical signals given proper definitions of keys, chords or recurrence, while reactions of listeners were not considerably addressed. In the following two chapters, we present two problems, music summarization and salience detection, involving human musical memory and the attentive listening process.

Music summarization (or, thumbnailing) aims at finding the most representative part of a musical piece. For example, for pop/rock songs, there are often catchy and repetitious parts (called the "hooks"), which can be implanted in your mind after hearing the song just once. This chapter analyzes the correlation between the representativeness of a musical part and its location within the global structure, and proposes a method to automate music summarization. Results will be evaluated both by objective criteria and human experiments.

5.1 Structural Accentuation of Music

The term "accent" in this chapter will be used to describe points of emphasis in the musical sound. Huron (1994) defined "accent" as "an increased prominence, noticeability, or salience ascribed to a given sound event." Lerdahl (1983) distinguished three kinds of accent: phenomenal, structural, and metrical. He especially described how structural accent is related to grouping: "structural accents articulate the boundaries of groups at the phrase level and all larger grouping levels."

Deliege (1987) stated that "in perceiving a difference in the field of sounds, one experiences a sensation of accent." Boltz (1986) proposed that "accents can arise from any deviation in pattern context." Thus, accents were hypothesized to occur at moments in which a change occurs in any of the auditory or visual aspects of the stimulus.

Additionally, in terms of long-time musical memory and musical events in larger scales, the repeating patterns will get the emphasis because they will strengthen their representations and connections in memory each time they repeat themselves.

Accents in music can happen at different levels. Here we are interested in accents at a higher and global level: which part of music is the theme or "hook" of the piece? Burns' paper (1987) on hook analysis summarized many definitions of "hook":

> *"It is the part of a song, sometimes the title or key lyric line, that keeps recurring" (Hurst and Delson 1980, p. 58)*
>
> *"A memorable catch phrase or melody line which is repeated in a song" (Kuroff 1982, p. 397)*
>
> *"An appealing musical sequence or phrase, a bit of harmony or sound, or a rhythmic figure that grabs or hooks a listener"(Shaw, 1982)*
>
> *"A musical or lyrical phrase that stands out and is easily remembered" (Monaco and Riordan's, 1980)*

From the perspective of song writing, there can be many types of hooks: rhythm, melody, harmony, lyric, instrumentation, tempo, dynamics, improvisation and accident, sound effects, editing, mix, channel balance, signal distortion, etc. Although the techniques of making hooks can be very different, the purpose is similar, which is to make the part of music unique and memorable through recurrence, variation and contrast, as Burns pointed out:

... repetition is not essential in a hook, but is not ruled out either. While hooks in the form of repetition may, to an extent, be 'the foundation of commercial songwriting' and record-making, repetition is meaningless without its opposite, change. ... Thus, repetition and change are opposite possibilities from moment to moment in music. The tension between them can be a source of meaning and emotion. Music-making is, to a large degree, the manipulation of structural elements through the use of repetition and change. Sometimes a repetition will be extreme, but often it will incorporate minor changes, in which case it is a variation. At certain points, major changes will occur.

Although it is not quite clear what makes a musical part a hook, the above has uncovered some properties of hooks. Thus, a hook should be a good balance between uniqueness and memorability. A hook should have some difference from the listener's previous listening experience, which makes it interesting rather than boring. On the other hand, a hook should be easy enough for memorizing. It should repeat itself for emphasizing and conform some cultural and aesthetic traditions to sound appealing.

5.2 Music Summarization via Structural Analysis

An ideal system for automatic music summarization should consider two aspects: one is the intrinsic property of a musical phrase, such as its melody, rhythm, instrumentation, from which we can infer how appealing or singable or familiar to the listeners it is; the other is the reference property of a musical phrase, such as the number of times it repeats and the location where it appears. A hook should appear at the right locations to make it catchier. Typically a good spot is the beginning or end of the chorus or refrain. It would be most memorable if placed there.

However, this dissertation will only emphasize the second aspect, because of the complexity and lack of psychological principles for the first aspect. That means where the main theme or "hook" of a musical piece normally appears (called *structurally accented locations* in this dissertation) will be mainly addressed as a key for music summarization. We will explore how the location of a musical phrase within the whole structure of the piece (relating to the number of repetitions, whether it is at the beginning of a section, etc.) affects its accent. Thus, although the location is by no means the only factor that determines the accentuation, since good musical works probably tend to make all the factors consistent, it should be effective enough to only look at the reference property of musical phrases. This is similar to summarizing a document: good articles tend to put key sentences at the beginning of each paragraph instead of the middle to catch the attention of readers.

Therefore, it would be helpful if the song has been segmented into meaningful sections before summarization for locating structurally accented locations, e.g., the beginning or the ending of a section, especially a chorus section. For example, among the 26 Beatles songs in Section 4.6, 6 songs have the song titles in the first phrase of a section; 9 songs have them in the last phrase of a section; and 10 songs have them in both the first and the last phrases of a section. Only one song has its title in the middle of a section. For many pop/rock songs, titles are contained in hooks. This information is very useful for music summarization: once we have the recurrent structure of a song, we can have different music summarization strategies for different applications or for different types of users.

In the following, the method we present will find the most representative part of music (specifically, hooks of pop/rock music) based on the result of recurrent structural analysis. Note that the summarization result using any of the following strategies will depend on the accuracy of the recurrent structural analysis.

5.2.1 Section-beginning Strategy (SBS)

The first strategy assumes that the most repeated part of the music is also the most representative part and the beginning of a section is typically essential. Thus, this strategy, illustrated by Figure 5-1, chooses the beginning of the most repeated section as the thumbnail of the music. The algorithm first finds the most repeated sections based on the structural analysis result, takes the first section among these and truncates its beginning (20 seconds in this experiment) as the thumbnail.

Figure 5-1: Section-beginning strategy.

5.2.2 Section-transition Strategy (STS)

We also investigated the music thumbnails at some commercial music web sites for music sales (e.g., Amazon.com, music.msn.com) and found that the thumbnails they use do not always start from the beginning of a section and often contain the transition part (end of section A and beginning of section B). This strategy assumes that the transition part can give a good overview of both sections and is more likely to capture the hook (or, title) of the song, though it typically will not give a thumbnail right at the beginning of a phrase or section.

Based on the structural analysis result, the algorithm finds a transition from section A to section B; and then it truncates the end of section A, the bridge and the beginning of section B (shown in Figure 5-2). The boundary accuracy is not very important for this strategy.

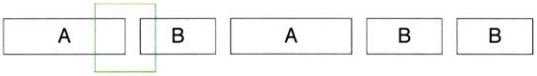


Figure 5-2: Section-transition strategy.

To choose the transition for summarization, three methods were investigated:

- STS-I: Choose the transition such that the sum of the repeated times of A and those of B is maximized; if there is more than one such transition, the first one will be chosen. In the above example, since there are only two different sections, either A → B or B → A satisfies the condition; thus the first transition from A to B will be chosen.

- STS-II: Choose the most repeated transitions between different sections; if there is more than one such transition, the first one will be chosen. In the above example, A → B occurs twice, B → A occurs once; thus the first transition from A to B will be chosen.

- STS-III: Choose the first transition right before the most repeated section. In the above example, B is the most repeated section; thus the first transition from A to B will be chosen.

Although in the above example, all these three methods will choose the same transition for summarization, we can come out with various other forms where the three methods will choose different transitions.

5.3 Human Experiment

The most difficult problem in music summarization is probably how to set up the ground truth to evaluate the generated summarizations. To some extent, if we know what a good summarization should be, we can always develop good strategies to generate a good summarization given the structure of music.

However, whether a summary is good or not is subjective and the answer may vary among different listeners. Here are some criteria for good music summarization summarized by Logan (2000) based on a survey from their human experiment:

1. *A vocal portion is better than instrumental.*
2. *It's nice to have the title sung in the summary.*
3. *The beginning of the song is usually pretty good; at least that gets an average.*
4. *It's preferable to start at the beginning of a phrase rather than in the middle.*

However, there was no quantitative result about how these criteria are important for evaluating summarizations. Therefore, we also conducted an online human experiment whose main purpose is to examine whether the structure of music and the location of phrases play a role in evaluating a summarization and how it varies from listener to listener.

5.3.1 Experimental Design

5.3.1.1 Data set

In the experiment, ten pieces were chosen, including five Beatles songs (various forms), three classical piano pieces, and two Chinese pop songs (Table 5-1). Titles of these pieces were not provided to the subjects during the experiment.

Table 5-1: Ten pieces used in the human experiment.

1	Beatles: Eight days a week
2	Beatles: I feel fine
3	Beatles: I want to hold your hand
4	Beatles: We can work it out
5	Beatles: Yellow submarine
6	Piano: Rubenstein: Melody In F
7	Piano: Beethoven: Minuet In G
8	Piano: Schumann: From Kinderszenen (1. Von Fremden Landern Und Menschen)
9	Chinese pop: Dong Feng Po
10	Chinese pop: Yu Jian

For each piece, five 20-second summarizations were generated as follows based on the true structure of each piece:

1) Random
2) Beginning of the second most repeated section, denoted as section A
3) Beginning of the most repeated section, denoted as section B
4) Transition A → B
5) Transition B → A

Three questions were asked for each summarization for rating from 1 (worst) to 7 (best):

Question 1: How does this summarization capture the gist of the song?
Question 2: How is this summarization good for advertising this song?
Question 3: How is it easy for you to identify the song based on the summarization?

5.3.1.2 Interface and Process

The subjects were instructed to go through the following process. They can stop at any point during the experiment and can resume from the stopping point if they wish:

1. Instruction of the experiment: The first page (Figure 5-3) illustrates the purpose and process of the experiment.

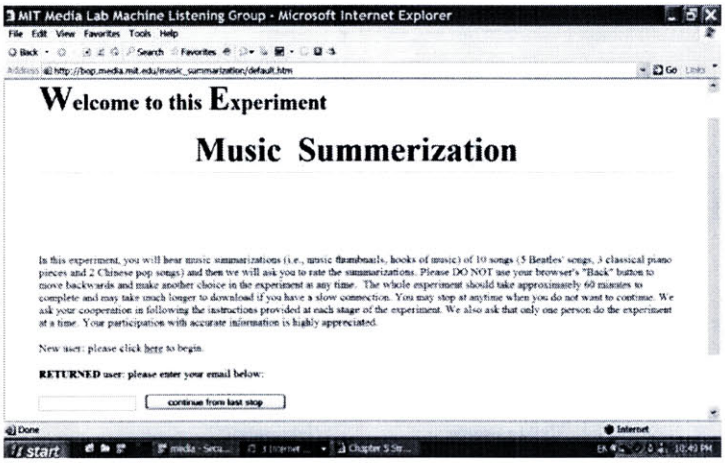


Figure 5-3: Instruction page.

2. Subject registration: Subjects provide their personal information related to the experiment, including age, gender, country, occupation, music experience, familiarity with Beatles songs, familiarity with western classical music, familiarity with Chinese pop music, language, etc. (Figure 5-4).

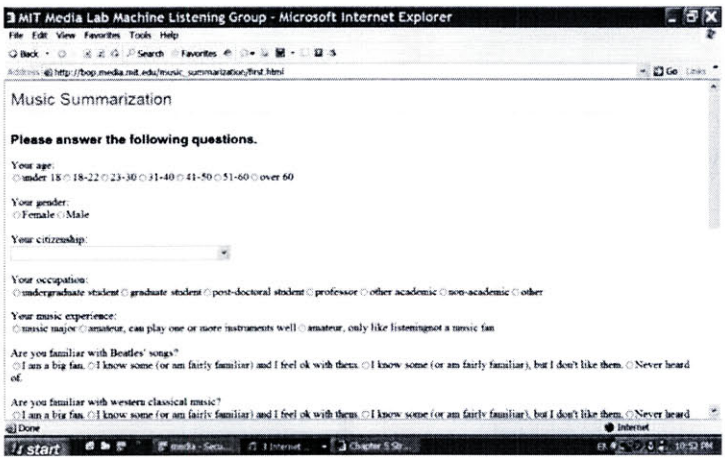


Figure 5-4: Subject registration page.

3. Thumbnail rating: Each page presents one song and its five summarizations for subjects to rate. Subjects also need to indicate their familiarity with the piece. The ten pieces come out with a random order for each subject to reduce the order effect and obtain roughly even samples for each piece in case some subjects do not complete the experiment.

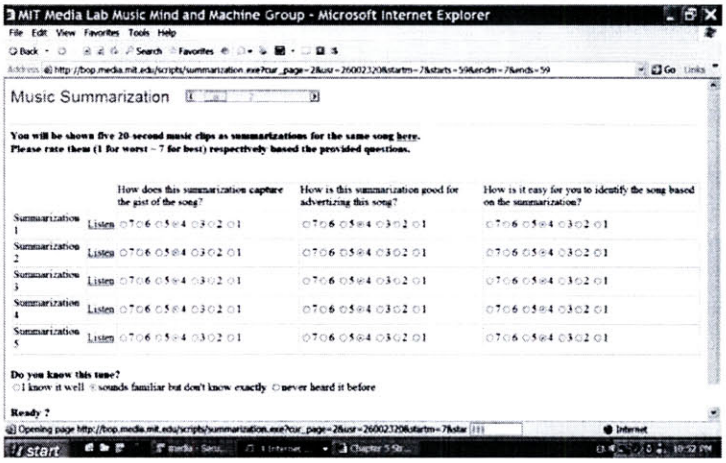


Figure 5-5: Thumbnail rating page.

4. Hook marking: After rating the summarizations of a piece, subjects are asked to manually input the hook in terms of the starting point and the ending point (Figure 5-6). Subjects also have the option to skip this question.

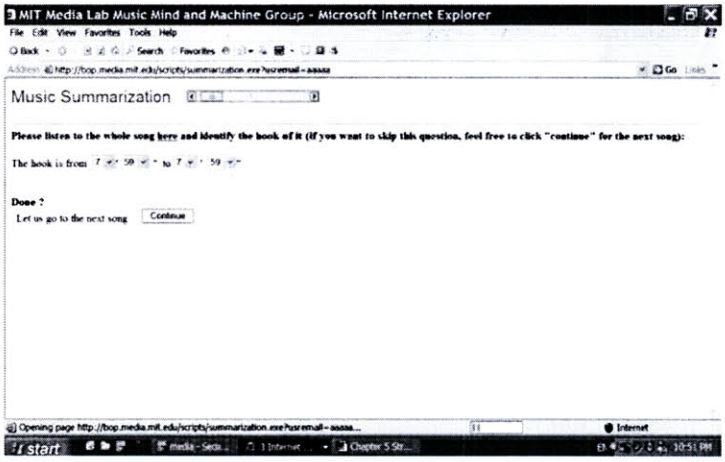


Figure 5-6: Hook marking page.

5. Survey: At the end of the experiment, subjects are asked to briefly describe how they choose the hook of a piece.

5.3.2 Subjects

Subjects were invited to the online experiment via emails to three different groups: MIT Media Lab mailing list, Chinese students at MIT including a Chinese chorus group, and Music Information Retrieval mailing list. Thus, most of the participants should be students and researchers around MIT or in the music information retrieval field.

Figure 5-7 shows a profile of sample size obtained. Duplicate records due to pressing the back button during the experiment were deleted before any of the following analysis. The left figure

indicates that about half of the registered subjects did not really participate in the experiment (i.e., did not rate any of the summarizations); the other half did part or the whole experiment. The right figure shows, for each of the ten pieces, how many subjects rated its summarizations or marked its hook, and how they were familiar with the piece.

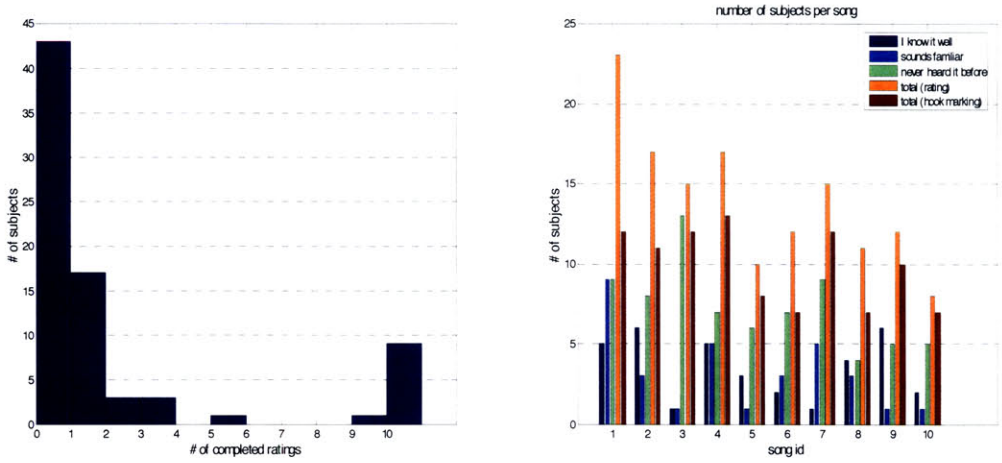


Figure 5-7: Profile of sample size (left: histogram of the number of competed ratings; right: number of subjects per song).

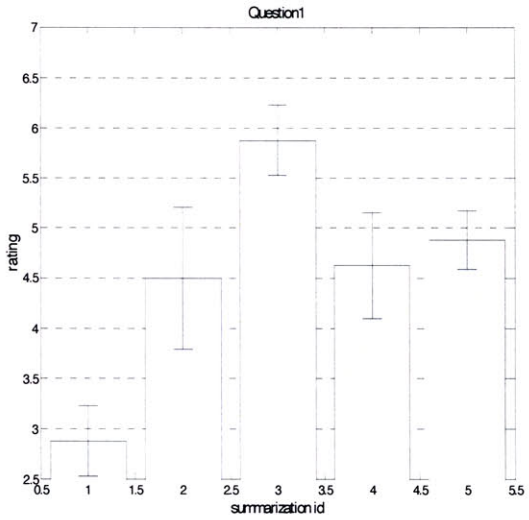
5.3.3 Observations and Results

In the following, we will present some questions we wanted to investigate and observations made from the experimental result:

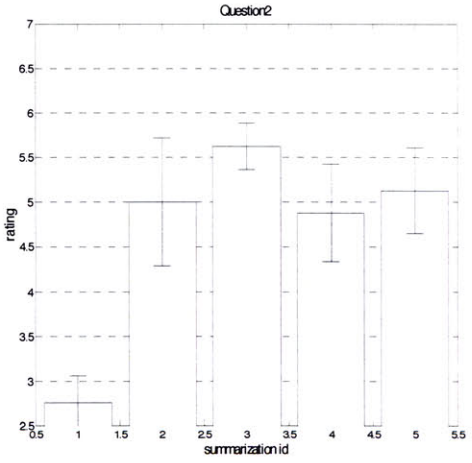
Question: What is the best summarization strategy? Are the summarizations generated based on the musical structure better than randomly generated ones?

Figure 5-8 shows the average ratings of the five summarizations (1 through 5 corresponding to the five generated summarizations described in Section 5.3.1.1) over all the rating samples for each type of summarization. Each error bar shows the standard error, which is the sample's standard deviation divided by $\sqrt{n}$ (n is sample size).

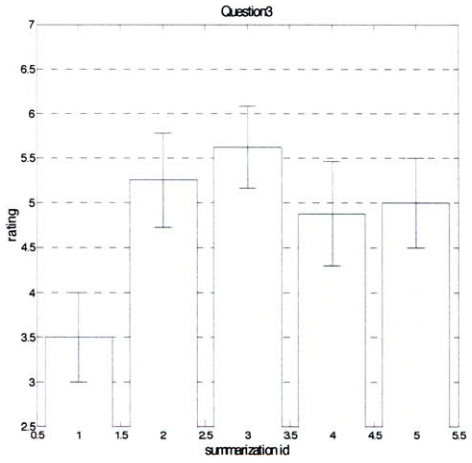
It clearly shows that all the four non-random summarizations are significantly better than the randomly generated summarization, for all the three questions. The third summarization corresponding to using the beginning of the most repeated section obtained the highest performance.



(**Question 1:** ***How does this summarization capture the gist of the song?***)



(**Question 2:** ***How is this summarization good for advertising this song?***)



(**Question 3:** ***How is it easy for you to identify the song based on the summarization?***)

Figure 5-8: Average ratings of the five summarizations.

Question: Are ratings for the three questions consistent?

In general, the ratings with regard to the three questions are quite consistent, but there are also slight variations. The rating for random summarization with regard to the third question obtained a higher score than those with regard to the first two questions, which suggests that even musical parts without hooks can help subjects identify the whole piece fairly easily. Although summarization 3 gets the highest ratings and summarization 1 gets the lowest ratings with regards to all the three questions, the orders of the other three summarizations are slightly different with regard to different questions. For example, summarization 2 obtained quite low average rating with high variation with regard to question 1, since the hook of the piece should be more likely to be contained the in section B rather than section A.

Question: Do the most repeated sections (section B) always get higher ratings than the second most repeated sections (section A)?

The result is, for 7 out of 10 pieces, section B got higher ratings. Among the other three, two are piano pieces (the 6th and 7th piece), where section A is very short and thus the 20-second summarization actually contains both section A and the beginning of section B. Therefore, for 6 out of the 7 pop/rock pieces, section B got higher ratings. Interestingly, for the 2nd piece (Beatles *I feel fine*) whose section A was rated higher, section B (the most repeated part) is actually the chorus part.

Question: How did the subjects identify the hooks?

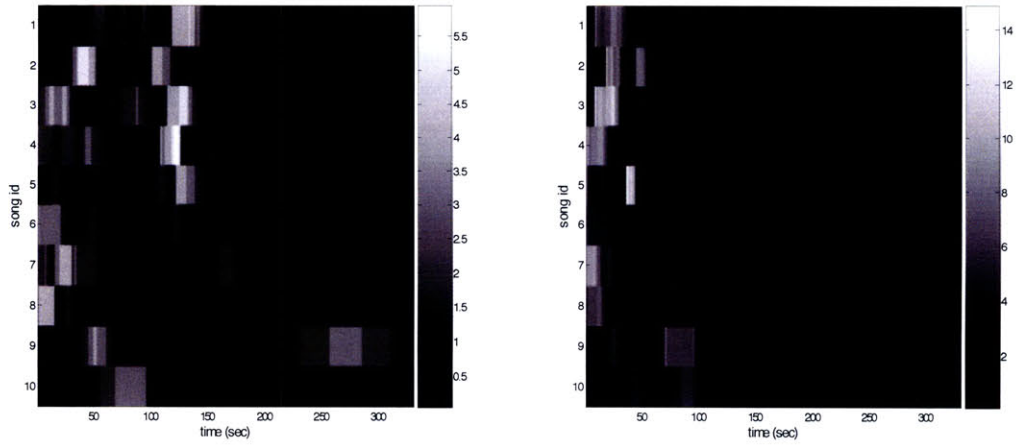


Figure 5-9: Hook marking result (left: without structural folding; right: with structural folding).

The left plot in Figure 5-9 shows a *hook-marking matrix*. The y-axis indicates different pieces. The x-axis indicates time. The color indicates how many times the part in this piece was included in the hooks marked by the subjects.

Since each piece is repetitive, the subject did not necessarily mark the first section or only one section that contains the hook. If we assume different appearances of the hook are equally important, we can fold the hook marking result in a way that if the subject mark the second or later appearance of the hook, we change it into the corresponding location of its first appearance. Thus, we can obtain another hook-marking matrix (i.e., *hook-marking matrix with structural folding*), shown as the right plot in Figure 5-9.

The most frequently marked hooks for the ten pieces in the experiment are summarized in Table 5-2 based on the right plot. It shows that,

1. Except for the piano pieces, all hooks are mainly vocal;
2. Except in the piano pieces, 3 out of 7 have hooks containing the titles of the songs;
3. Hooks for all the piano pieces start at the beginning of the pieces; for the pop/rock songs, only 2 out of 7 start the hooks at the beginning of the songs;
4. All hooks start at the beginning of a phrase; 7 out of 10 start the hooks at the beginning of a section.

Table 5-2: Most frequently marked hooks for the ten pieces.

1	"hold me, love me"
2	"Baby's good to me, you know, She's happy as can be, you know, She said so."
3	"Then I'll say that something, I wanna hold your hand,"
4	"Try to see it my way, Do I have to keep talking till I can't go on?"
5	"We all live in a yellow submarine, Yellow submarine, yellow submarine"
6	First 12 seconds of the beginning
7	First 10 seconds of the beginning
8	First 15 seconds of the beginning
9	The whole section B (chorus part containing the song's title)
10	The last two phrases in section B

Eight subjects answered the question about how they chose the hooks. Three subjects stated they would consider the repeating times of phrases; one subject said he/she would need a transition to both themes; one chose the parts he/she liked most; two mentioned it was hard to describe; one mentioned three aspects including repeating times, the beginning of the piece (especially for classical music) and the climax. This suggests that different subjects seem have different criteria for choosing summarizations, though the repeating time of a phrase is quite important for most listeners.

Question: How lyrics are important for hook identification? Is there any cultural difference when identifying hooks?

We chose two Chinese pop songs in the data set in order to investigate whether lyrics are important for hook identification and whether there is any difference of hook perception between people who understand the lyrics and people who do not understand the lyrics.

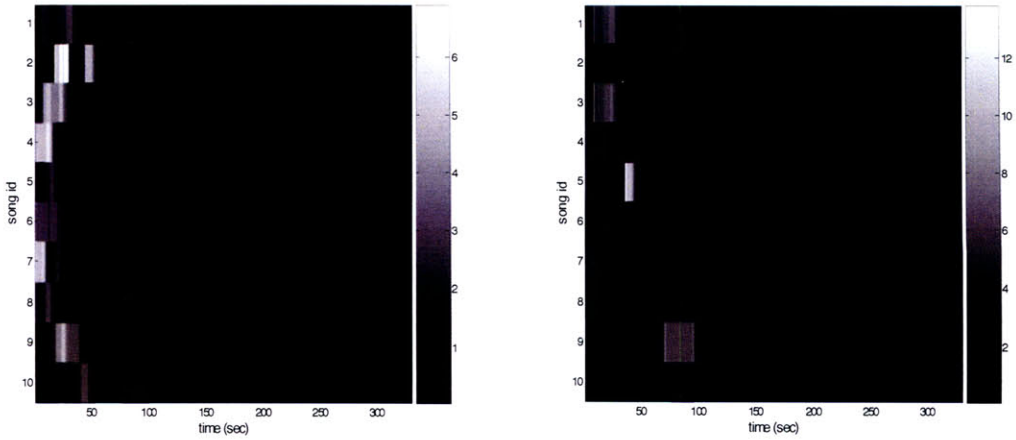


Figure 5-10: Hook marking result with structural folding (left: non-Chinese subjects; right: Chinese subjects).

Figure 5-10 shows the hook marking result with structural folding from people who do not understand Chinese (left) and from people who understand Chinese (right). It does show some difference. The biggest difference occurs for the ninth song, which is a Chinese pop song. Almost all subjects who do not understand Chinese marked section A as the hook, while all subjects who understand Chinese marked section B (containing the song's title) as the hook. It seems lyrics do play a role in hook identification. However, bigger sample size might be necessary to draw a conclusion on this.

5.4 Objective Evaluation

Based on the previous human experiment, five criteria for pop/rock music are considered for evaluating the summarization result. These criteria include:

1) The percentage of generated thumbnails that contain a vocal portion;

2) The percentage of generated thumbnails that contain the song's title;

3) The percentage of generated thumbnails that start at the beginning of a section;

4) The percentage of generated thumbnails that start at the beginning of a phrase.

5) The percentage of generated thumbnails that capture a transition between different sections.

Since using the beginning of a piece seems a fairly good strategy for classical music, we will only consider pop/rock music in the following. Table 5-3 shows the performance of all the strategies (SBS, STS-I, STS-II and STS-III) presented in Section 5.2 using the 26 Beatles songs (Appendix A). For evaluating transition criterion (5^{th} column), only the 22 songs in our corpus that have different sections were counted.

The comparison of the thumbnailing strategies clearly shows that the section-transition strategies generate a lower percentage of thumbnails starting at the beginning of a section or a phrase, while these thumbnails are more likely to contain transitions. SBS has the highest chance to capture the vocal and STS-I has the highest chance to capture the title.

It is possible, though, to achieve better performance using this strategy, if we can improve the structural analysis accuracy in the future.

Table 5-3: 20-second music summarization result.

	Vocal	Title	Beginning of a section	Beginning of a phrase	Transition
SBS	100%	65%	62%	54%	23%
STS-I	96%	73%	42%	46%	82%
STS-II	96%	62%	31%	46%	91%
STS-III	96%	58%	31%	50%	82%

5.5 Summary

An online human experiment has been conducted to help set up the ground truth about what makes a good summarization of music. Strategies for thumbnailing based on structural analysis were also

proposed in this chapter. There can be other variations as well. We should choose appropriate strategies for different applications and different length constraint. For example, the section-beginning strategy might be good for indexing query-based applications, because it is more likely that the user will query from the beginning of a section or a phrase. The section-transition strategy might be good for music recommendation, where it might be more important to capture the title in the thumbnail.

Music segmentation, summarization and structural analysis are three coupled tasks. Discovery of effective methods for undertaking any one of these three tasks will benefit the other two. Furthermore, the solution to any of them depends on the study of human perception of music, for example, what makes a part of music sounds like a complete phrase and what makes it memorable or distinguishable. Human experiments are always necessary for exploring such questions.

Chapter 6 Musical Salience for Classification

Another problem related to Chapter 5 is about musical salience or the "signature" of music: the most informative part of music when we make a judgment of its genre, artist, style, etc. Thus, this problem is similar to music summarization, since both try to identify important parts of music for some purposes. The difference here is the salient parts extracted are not necessarily used to identify the musical piece itself, as in music summarization, but to identify its category in some sense. This chapter investigates whether a computer system can detect the most "informative" parts of music for a classification task.

6.1 Musical Salience

If humans are asked to listen to a piece of music and tell who is the singer or who is the composer, we typically will hold our decision until we get to a specific point that can show the characteristics of that singer or composer in our mind (called the *signature* of the artist). For example, one of the author's favorite Chinese female singers, Faye Wong, has a very unique voice in her high pitch range. Whenever we hear that part of her song, we can immediately identify she is the singer. Here is another example. Many modern Chinese classical composers attempted to incorporate Chinese traditional elements in their western-style work. One way they did it was adding one or two passages in their work using Chinese traditional instruments and/or tunes, so that the listeners can easily get the sense of the characteristic of that piece.

Even if we are not asked to do any judgments of the above kinds explicitly, we will naturally do these while we are listening to music. And this is related to our musical memory and attentive listening process. Deliège concluded that categorization is a universal feature of attentive listening. Typically, a listener has some exposure to various types of music - music of different genres, of different styles, by different singers, by different composers, of a happy or sad mode, etc. The characteristics of a type of music will be stored in our long-time musical memory as a prototype, if we have been familiar enough with that type. Thus, there must be a training stage of forming these prototypes. When we hear a new piece of music, we will compare the elements from the piece with various types of music in our mind and make all kinds of the judgments accordingly. Imagine if there existed a type of music that has no elements in common with the types of music we are familiar with, we would not be able to make any sense of it the first time we hear it, because we cannot make any judgment based on our previous musical experience. Therefore, investigating musical salience can greatly help us understand the musical listening process and how human musical memory is organized.

6.2 Discriminative Models and Confidence Measures for Music Classification

6.2.1 Framework of Music Classification

Methods for music classification can be summarized into two categories as summarized in Section 2.3.2. The first method is to segment the musical signal into frames, classify each frame independently, and then assign the sequence to the class to which most of the frames belong. It can be regarded as using multiple classifiers to vote for the label of the whole sequence. This technique works fairly well for timbre-related classifications. Pye (2000) and Tzanetakis (2002) studied genre classification. Whitman (2001), Berenzweig (2001, 2002) and Kim (2002) investigated artist/singer classification. In addition to this frame-based classification framework, the second method attempted to use features of the whole sequence (e.g., emotion detection by Liu, 2003), or use models capturing the dynamic of the sequence (e.g., Explicit Time Modeling with Neural Network and Hidden Markov Models for genre classification by Soltau, 1998) for music classification.

This chapter focuses on the first method for music classification, investigating the relative usefulness of different musical parts when making the final decision of the whole musical piece, though the same idea might also be explored for the second method.

Thus, the question that this chapter addresses is which part of a piece should contribute most to a judgment about music's category when applying the first classification framework and whether what is "important" for machines (measured by *confidence*) is consistent with human intuition. When voting for the label of the whole sequence at the last step, we will consider the confidence of each frame. Here, we want to explore several definitions of confidence, and see whether we can throw away the "noisy" frames and use only the "informative" frames to achieve equally good or better classification performance. This is similar to Berenzweig's work (2002), which tried to improve the accuracy of singer identification by first locating the vocal part of the signal and then using only that part for identifying the singer. The main difference here is that we do not assume any prior knowledge about which parts are "informative" (e.g., the vocal part is more informative than the accompaniment part for singer identification); on the contrary, we let the classifier itself choose the most "informative" parts by having been given a proper definition of confidence. We then can analyze whether the algorithmically chosen parts are consistent with our intuition. Thus, to some extent, it is a reverse problem of Berenzweig's: if we define the confidence well, the algorithm should choose the vocal parts automatically for singer identification.

There is another possibility, of course: the algorithmically chosen parts are not consistent with human intuition. If this happens, two possibilities need to be considered: first, the algorithm and the definition of confidence can be improved; second, computers can use the information humans cannot observe, or the information we do not realize we are using. One example of this is the "album effect": when doing artist identification, the classifier actually identifies the album instead of the artist himself/herself by learning the characteristics of audio production in the recording. Thus, although the classification accuracy might be high, we cannot expect it to perform equally well for samples under a different recording condition.

Specifically, in the following, the first three steps are the same as the most-widely used approach for music classification:

1. Segment the signal into frames and compute the feature of each frame (e.g., Mel-Frequency Cepstral Coefficients);

2. Train a classifier using frames of the training signals independently;

3. Apply the classifier to the frames of the test signals independently; each piece is assigned to the class to which most of the frames belong;

Following these is one additional step:

4. Instead of using all the frames of a test signal for determining its label, a portion of the frames are chosen according to a specific rule (e.g., choose randomly, choose the ones of large values of confidence) to determine the label of the whole signal.

The last step can be regarded as choosing from a collection of classifiers for the final judgment. Thus, the confidence measure should be able to capture the reliability of the classification, i.e. how certain that the classification is correct. And we want to compare the performances of using different rules for choosing the frames, and examine whether the algorithmically selected parts of a piece are consistent with human's intuition.

6.2.2 Classifiers and Confidence Measures

Let us consider discriminative models for classification. Suppose the discriminant function $S(\mathbf{x}) = \hat{y}$ is obtained by training a classifier, the confidence of classifying a test sample $\mathbf{x}$ should be the predictive posterior distribution:

$$C(\mathbf{x}) = P(y = \hat{y} \mid \mathbf{x}) = P(y = S(\mathbf{x}) \mid \mathbf{x}) \tag{6-1}$$

However, it is generally not easy to have the posterior distribution. Thus, we need a way to estimate it, which is natural for some types of classifiers, while not so natural for some others.

In the following, we will focus on linear classification, i.e., $S(\mathbf{x}) = \hat{y} = sign(\mathbf{w}^T\mathbf{x})$, since nonlinearity can easily be incorporated by kernelizing the input point. Among the linear classifiers, the Support Vector Machine (SVM) is representative of the non-Bayesian approach, while the Bayes Point Machine (BPM) is representative of the Bayesian approach. Thus, this chapter will investigate these two linear classifiers and their corresponding confidence measures.

For BPM, $\mathbf{w}$ is modeled as a random vector instead of an unknown parameter vector. Estimating the posterior distribution for BPM was extensively investigated by Minka (2001) and Qi (2002; 2004). Here, Predictive Automatic Relevance Determination by Expectation-propagation (Pred-ARD-EP), an iterative algorithm for feature selection and sparse learning, will be used for classification and estimating the predictive posterior distribution:

$$\begin{aligned} C(\mathbf{x}) &= P(y = \hat{y} \mid \mathbf{x}, D) \\ &= \int_{\mathbf{w}} P(\hat{y} \mid \mathbf{x}, \mathbf{w}) p(\mathbf{w} \mid D) d\mathbf{w} = \Psi(z) \end{aligned} \tag{6-2}$$

$$z = \frac{(\hat{y}\mathbf{m}_w)^T \mathbf{x}}{\sqrt{\mathbf{x}^T \mathbf{V}_w \mathbf{x}}} \tag{6-3}$$

where D is the training set, $\mathbf{x}$ is the kernelized input point, $\hat{y}$ is the predictive label of $\mathbf{x}$. $\Psi(a)$ can be a step function, i.e., $\Psi(a) = 1$ if $a > 0$ and $\Psi(a) = 0$ if $a \le 0$. We can also use the logistic function or probit model as $\Psi(\cdot)$. Variables $\mathbf{m}_w$ and $\mathbf{V}_w$ are mean and covariance matrix of the posterior distribution of $\mathbf{w}$, i.e., $p(\mathbf{w} \mid t, \alpha) = N(\mathbf{m}_w, \mathbf{V}_w)$, where α is a hyper-parameter vector in the prior of $\mathbf{w}$, i.e., $p(\mathbf{w} \mid \alpha) = N(0, \mathrm{diag}(\alpha))$.

Estimating the posterior distribution for SVM might not be very intuitive, because the idea for SVM is to maximize the margin instead of estimating the posterior distribution. If we mimic the confidence measure for BPM, we obtain

$$C(\mathbf{x}) = \Psi(z) \tag{6-4}$$

$$z = (\hat{y}\mathbf{w})^T \mathbf{x} \tag{6-5}$$

Thus, the confidence measure for Pred-ARD-EP is similar to that for SVM except that it is normalized by the square root of the covariance projected on the data point. The confidence measure for SVM is proportional to the distance between the input point and the classification boundary.

6.2.3. Features and Parameters

For both SVM and Pred-ARD-EP, a RBF basis function (Equation 6-6) was used with $\sigma = 5$. A Probit model was used as $\Psi(\cdot)$. The maximum lagrangian value in SVM (i.e., C) was set to 30. All the parameters were tuned based on several trials (several splits of training and testing data) to obtain the highest possible accuracy.

$$K(x, x') = \exp(-\frac{\| x - x' \|^2}{2\sigma^2}) \qquad (6\text{-}6)$$

The features used for both experiments were Mel-frequency Cepstral Coefficients (MFCC) representation, which are widely used for speech and audio signals.

6.3 Experiment 1: Genre Classification of Noisy Musical Signals

Two data sets were chosen for the convenience of analyzing the correlation between algorithmically selected frames based on confidence and intuitively selected frames based on prior knowledge. The first is a genre classification data set with the first half of each sequence replaced by white noise for further investigation of consistency between the important part of each sequence (i.e., the second half) and the part with high confidence. The second is a data set of monophonic singing voice for gender classification. In both cases, we only consider binary classifications. Specifically, for either experiment, the data set was sampled at 11kHz sampling rate. Analysis was performed using frame size of 450 samples (~40 msec) and frames were taken every 225 samples (~20 msec). MFCCs were computed for each frame. Only every 25th data frame was used for training and testing because of the computer memory constraint. 30% of the sequences were used for training, while 70% were used for testing. The performance was averaged over 60 trials.

The data set used in the first experiment consists of 65 orchestra recordings and 45 Jazz recordings of 10 seconds each. The MFCCs of the first half frames of each sequence (both training and testing) were replaced by random noise normally distributed with $m = m_0$ and $\sigma = \sigma_0$ or $\sigma = 0.1 \cdot \sigma_0$, where m_0 and σ_0 are the mean and standard deviation of the original data (including data from both classes).

Figure 6-1 gives an example of the distribution of added noise (all data points in the plots were generated for illustration and were not the original musical data). Since the noise is added to training signals of both classes, these noisy points are labeled as a mixture of points from class 1 and points from class 2. Therefore, in bother of the cases shown in Figure 6-1, the data set is not linear separable. However, when $\sigma = \sigma_0$, the noisy points are mixed with the original data points; when $\sigma = 0.1 \cdot \sigma_0$, the noisy points are separable from the original data points.

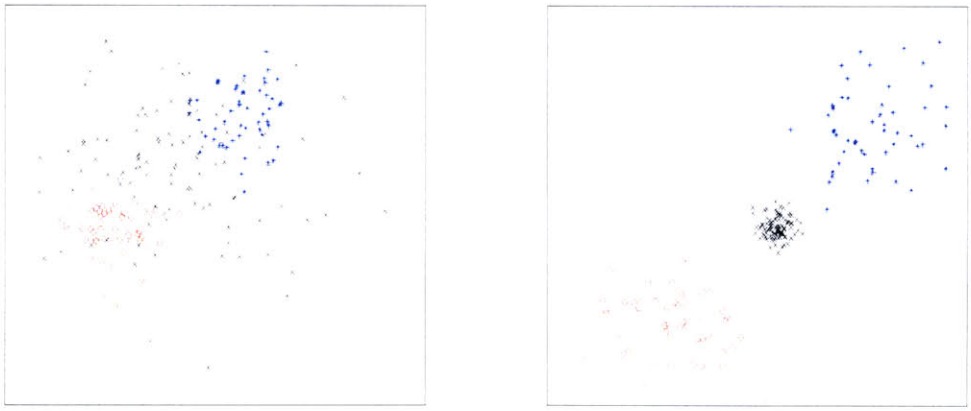


Figure 6-1: Distribution of added noise (left: $\sigma = \sigma_0$; right: $\sigma = 0.1 \cdot \sigma_0$): o for class 1, * for class 2 and x for added noise.

Figure 6-2 shows the result when $\sigma = \sigma_0$. The accuracy is evaluated by the percentages of test pieces correctly classified.

In Figure 6-2, the x-axis denotes the selection rate, which means the percentage of frames selected according to some criterion. The two horizontal lines are baselines, corresponding to the performances using all the frames available to each sequence (the above is confidence-weighted meaning each frame contributes differently to the label assignment of the whole signal based on confidence; the below is not confidence-weighted). The other four curves, *a* through *d* from top to the bottom, correspond to:

a) Selecting frames appearing later in the piece (thus, larger frame indexes and fewer noisy frames),
b) Selecting frames with highest confidence,
c) Selecting randomly,
d) Selecting frames with lowest confidence.

All these four curves approach the lower baseline when the selection rate goes to 1. It is easy to explain the peaks at selection rate 50% in curve *a*, since half of the frames were replaced by noise. The order of these four curves is consistent with our intuition. Curve *a* performed the best because it used the prior knowledge about data.

Figure 6-3 show the results when $\sigma = 0.1 \cdot \sigma_0$. In this case, the added noise has a distribution more separable from the original data. Interestingly, comparing to the result when $\sigma = \sigma_0$, the performance using SVM got much lower, while the performance using Pred-AD-EP did not get lower and its performance with frame selection based on confidence even got higher than its performance with frame selection based on prior knowledge for selection rate below 50%.

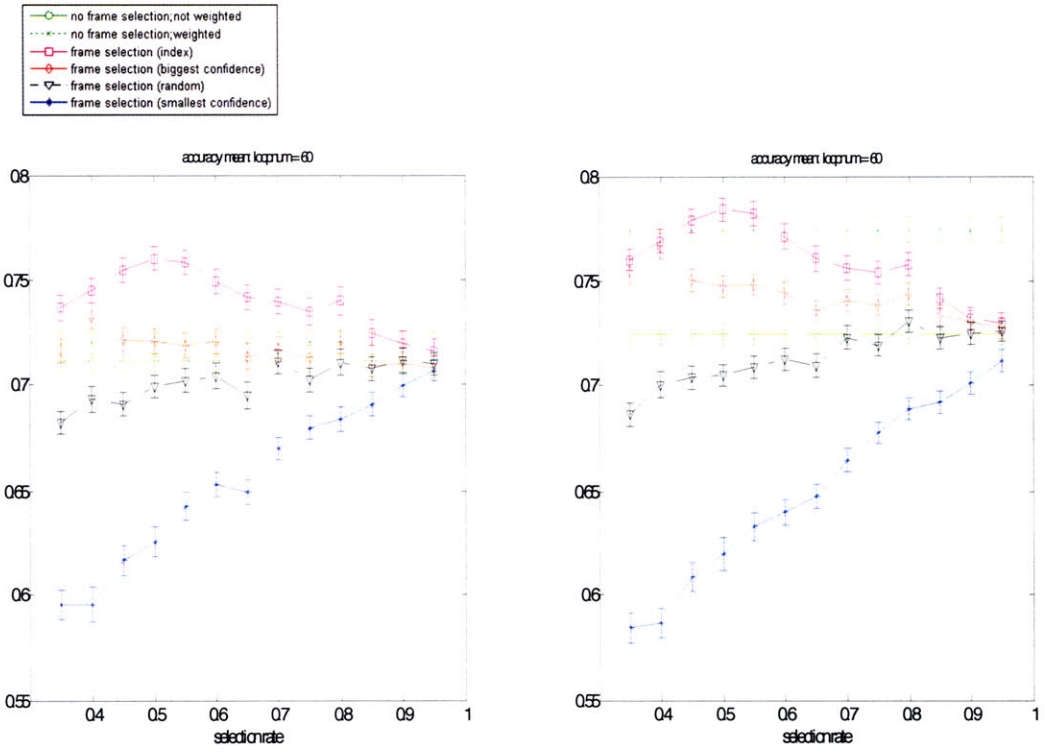


Figure 6-2: Accuracy of Genre Classification with Noise $\sigma = \sigma_0$ (left: Pred-ARD-EP; right: SVM)

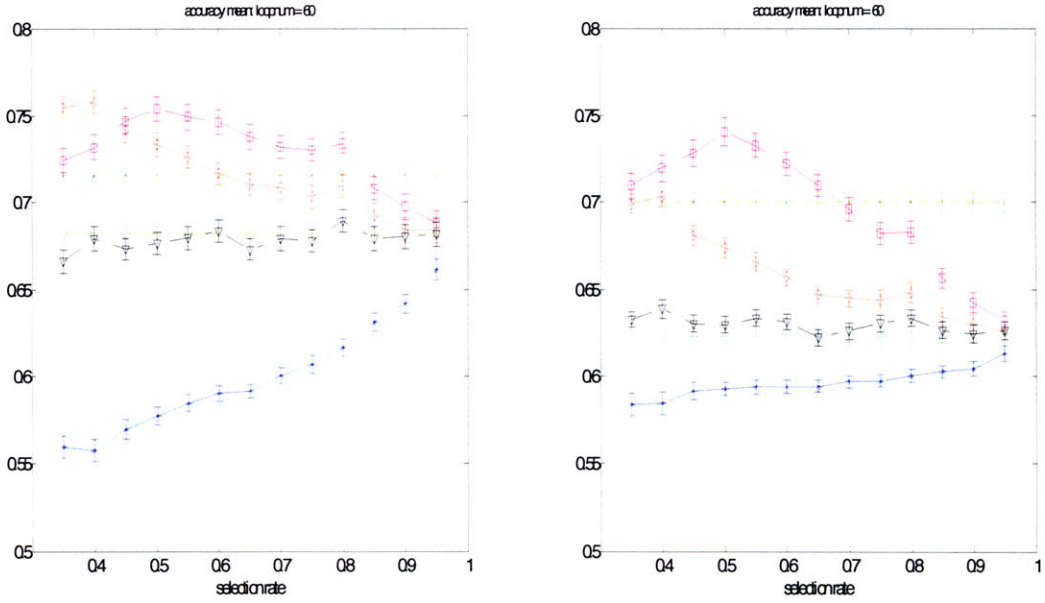


Figure 6-3: Accuracy of Genre Classification with Noise $\sigma = 0.1 \cdot \sigma_0$ (left: Pred-ARD-EP; right: SVM)

We also want to know the property of the selected frames. Figure 6-4 and 6-5 show the percentage of selected frames (selecting by random, by confidence and by index) that are noise (first half of each piece) or not noise (second half of each piece) at selection rate 50%. As we expected, frame selection based on confidence does select slightly more frames at the second half of each piece (not entirely though). In particular, if we look at the distribution of selected frames in Figure 6-5 and compare it to Figure 6-4, Pred-AD-EP did tend to select much more un-noisy data for the final decision, which explains why its corresponding performance is good.

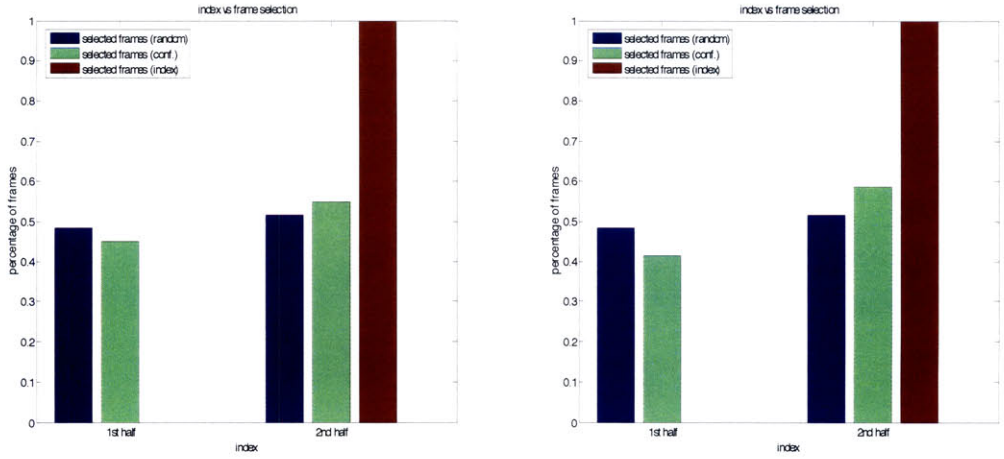


Figure 6-4: Index distribution of selected frames at selection rate 50%, $\sigma = \sigma_0$ (left: Pred-ARD-EP; right: SVM)

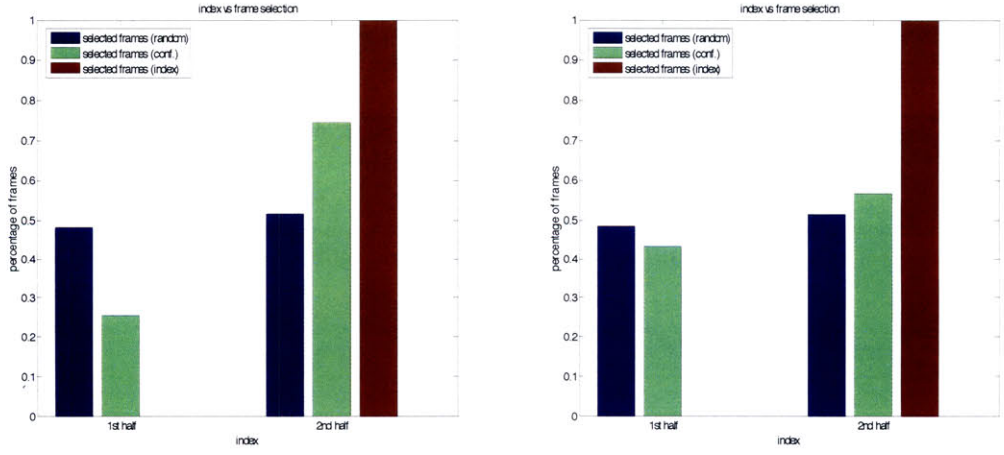


Figure 6-5: Index distribution of selected frames at selection rate 50%, $\sigma = 0.1 \cdot \sigma_0$ (left: Pred-ARD-EP; right: SVM)

6.4 Experiment 2: Gender Classification of Singing Voice

The data set used in this experiment consists of monophonic recordings of 45 male singers and 28 female singers, one sequence for each singer. All the other parameters are the same as the first experiment except that no noise was added to the data, since we here want to analyze whether the algorithmically selected frames are correlated with the vocal portion of the signal.

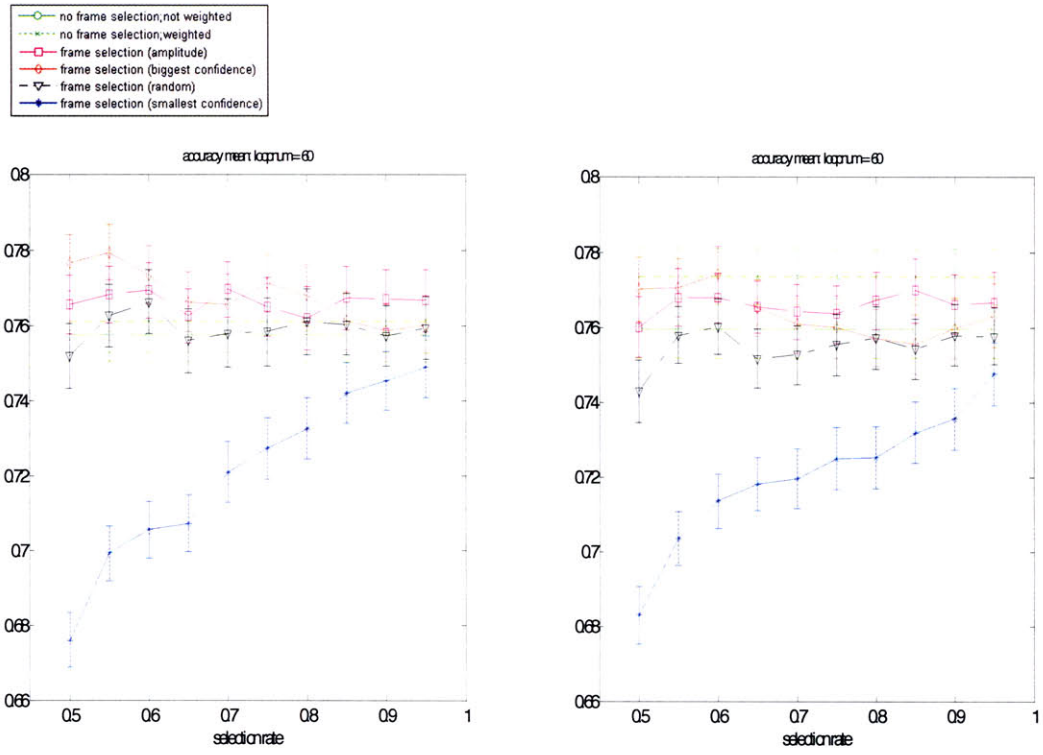


Figure 6-6: Accuracy of Gender Classification of Singing Voice (left: Pred-ARD-EP; right: SVM)

The results from the experiment are summarized in Figure 6-6. Similarly, the two horizontal lines in Figure 6-6 are baselines. The other four curves, *a* through *d* from top to the bottom, correspond to:

a) Selecting frames of the highest confidence,
b) Selecting frames of the highest energy,
c) Selecting randomly
d) Selecting frames of the lowest confidence.

In curve *b*, we used amplitude instead of index (i.e., location of the frame) as the criterion for selecting frames, because the data set consists of monophonic recording of singing voice and amplitude can be a good indicator of whether there is vocal at the time. The order of these four curves can be explained the way similar to the last experiment, except that, selecting frames based on prior knowledge does not seem to outperform selecting frames based on confidence. The reason here might be that amplitude itself cannot completely determine whether the frame contains vocal or not. For example, an environmental noise can also cause high amplitude. It might be better to combine other features, e.g., pitch range, harmonicity, to determine the vocal parts.

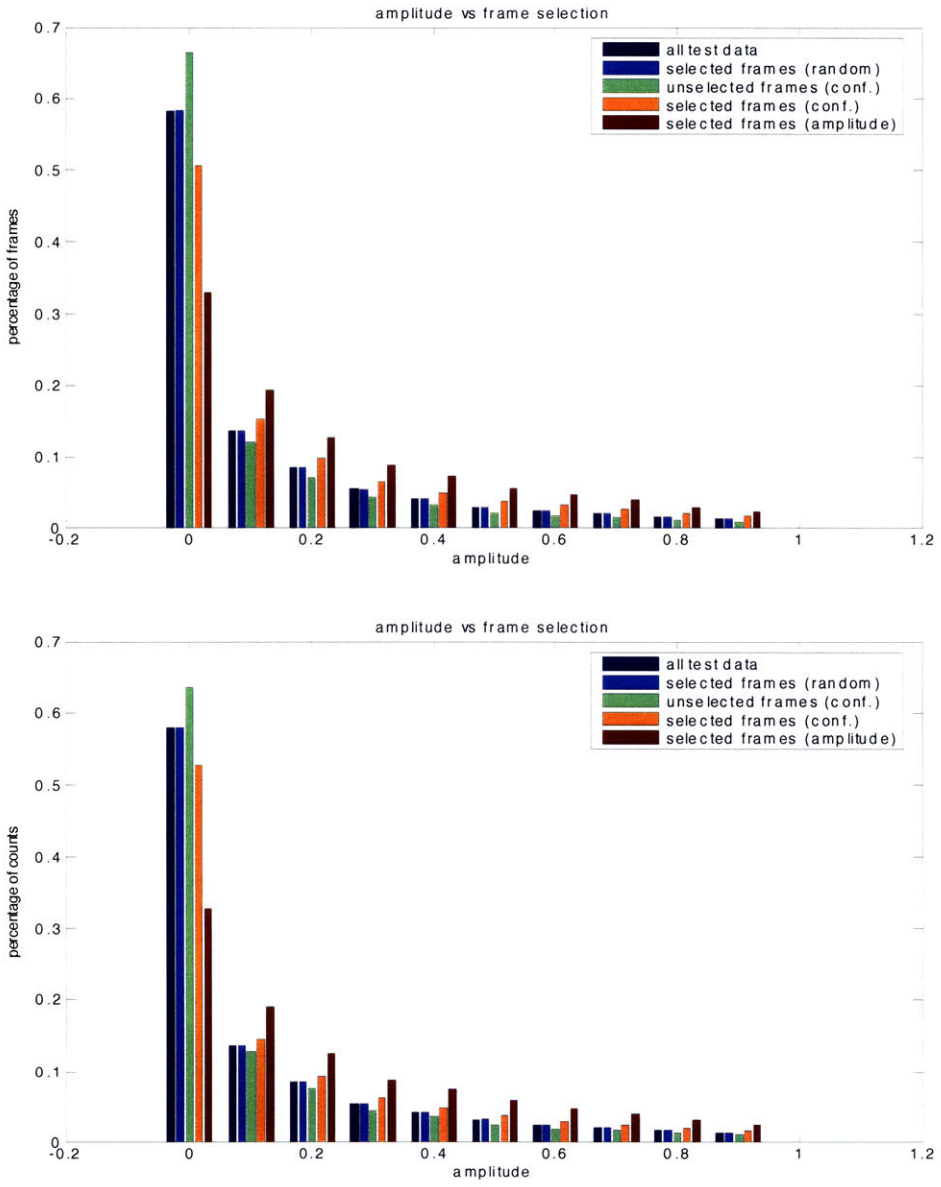


Figure 6-7: Amplitude distribution of selected frames at selection rate 55% (above: Pred-ARD-EP; below: SVM)

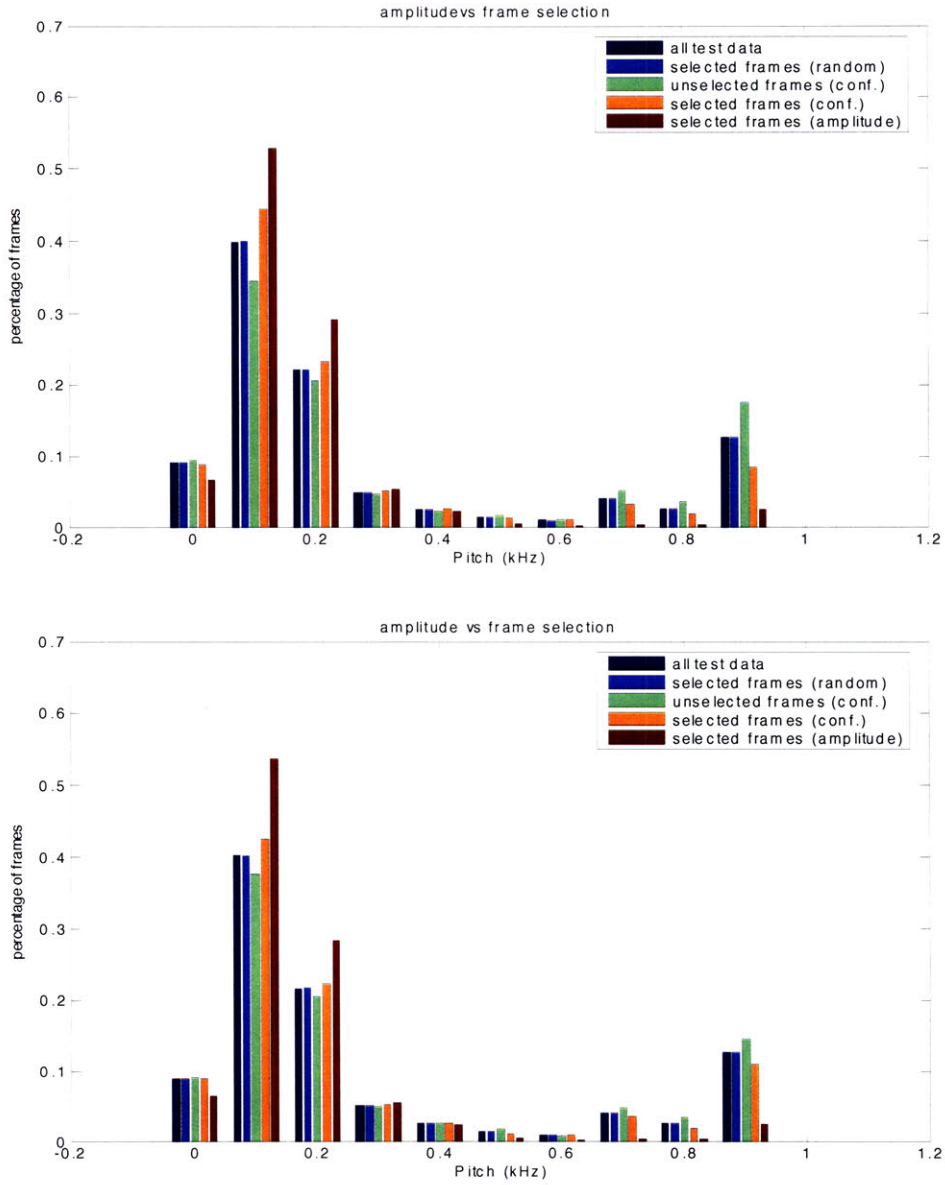


Figure 6-8: Pitch distribution of selected frames at selection rate 55% (above: Pred-ARD-EP; below: SVM)

Figure 6-7 shows the histogram (ten bins divided evenly from 0 to 1) of the amplitude of selected frames at selection rate 55%. The five groups correspond to distributions of all test data, selected frames by random selection, discarded frames by confidence-based selection, selected frames by confidence-based selection and selected frames by amplitude-based selection. As we expected, the frame selection based on confidence does tend to select frames that are not silence.

To show the correlation between confidence selection and another vocal indicator – pitch range, Figure 6-8 shows the histogram (ten bins divided evenly from 0 to 1kHz) of the pitches of selected frames at selection rate 55%. Pitch of each frame was estimated by autocorrelation. It clearly shows that the frame selection based on confidence tends to choose frames that have pitches around 100~400Hz corresponding to a typical pitch range of human speakers. Note that although

the data set used here is singing voice instead of speech, most singers sang in a casual way so that the pitches are not as high as normal professional singing.

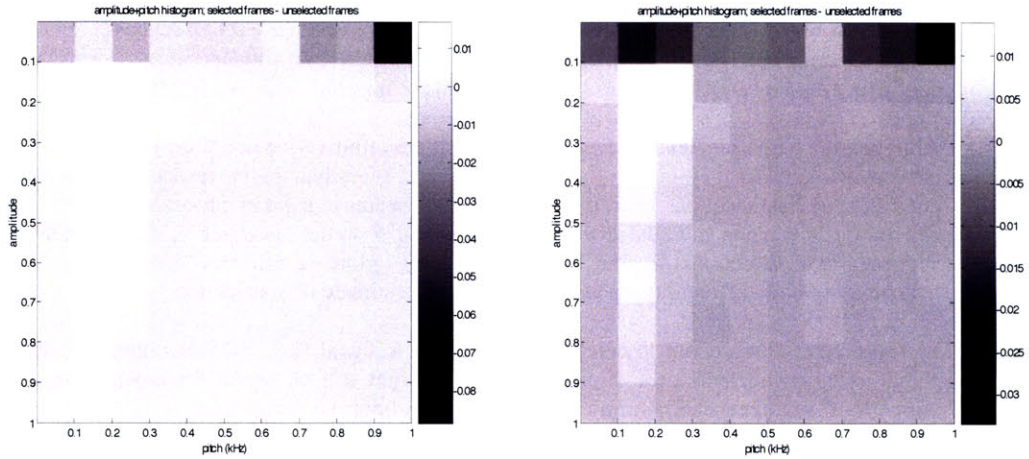


Figure 6-9: Difference of pitch vs amplitude distribution between selected frames and unselected frames at selection rate 55% (left: Pred-ARD-EP; right: SVM)

To show the correlation between confidence selection and the two vocal indicators (amplitude and pitch range) together, Figure 6-9 shows the amplitude-pitch distribution difference between selected frames and unselected frames based on confidence. It clearly shows that the frame selection based on confidence tends to choose frames that have higher amplitude and pitches around 100~400Hz.

6.5 Discussion

The experimental results demonstrate that the confidence measures do, to some extent, capture the importance of data, which is also consistent with the prior knowledge. The performance is at least equally good as the baseline (using all frames), slightly worse than using prior knowledge properly, but significantly better than selecting frames randomly. This is very similar to human perception: for humans to make a similar judgment (e.g., singer identification), given only the signature part should be as good as given the whole piece, and much better than given the trivial parts.

Although this chapter does not aim at comparing Pred-ARD-EP and SVM, for the first data set, SVM slightly outperformed Pred-ARD-EP when the added noise is not separable from the original data, while Pred-ARD-EP outperformed SVM when the added noise is separable from the original data. The second case corresponds to the situation when only part of each musical signal has the "signature" information (or musical salience) of its type, while the other parts of the signals (non-signature parts) may share the same distribution with the non-signature parts of signals from another class and are separable from the distribution of the "signature" parts. This is probably more common in many music applications. For example, for singer identification, the signature parts should be the vocal portion; the non-signature parts should be the instrumental portion. Thus, the non-signature parts of different singers might share the same distribution (assuming same genre) and the instrumental portion should be easily separable from the vocal portion in the feature space, if properly defined.

The result from the first experiment seems also to suggest that SVM is more sensitive to the types of noise added. This is consistent with the conclusion that SVM is in general more sensitive to outliers in the data set because its boundary depends on the support vectors. On the other hand,

Pred-ARD-EP attempts to use Bayesian method to estimate the predictive posterior probability and can better tolerate the impact of different types of noises.

Another interesting phenomenon is, for SVM, the performance curve corresponding to confidence-based selection is always between the two baselines (confidence-weighted or not weighted using all frames), while, for Pred-ARD-EP, the performance curve corresponding to confidence-based selection can go significantly above the confidence-weighted baseline.

Although the classifiers tend to choose frames that are intuitively more "informative", they did not choose as many as they could; the noisy parts and the silent parts, respectively, still seem to contribute to classification. Thus, the "album effect" seems to exist in our cases. This effect should depend on how good the confidence measure is and how the classifier deals with noise, which suggests two directions in the future: exploring more confidence measures and further investigating how different types of noise impact the estimate of confidence.

For classifiers that attempt to catch the dynamic of a signal (e.g., HMM) rather than model the signal as independent frames, similar techniques might still be useful for determining the most "important" segments and ignoring irrelevant parts to improve the accuracy.

6.6 Summary

This chapter presented a framework of music classification with partial selection based on confidence measures. The experiments showed some consistency between algorithmically selected musical parts based on confidence and the salient parts of music that humans would identify, for making a judgment. This will help us understand music listening process; from a practical point of view, using only the musical salience rather than the whole musical signal can help improve the classification accuracy.

Chapter 7 Conclusions

This dissertation has presented a framework for automated analysis of musical structure, including tonality analysis, recurrent structure analysis, hook analysis and salience analysis. The utility of the framework has been demonstrated in the applications of music segmentation, summarization and classification. These results have been validated through quantitative evaluations and perceptual listening experiments using real-world musical stimuli. In this concluding chapter, we will summarize the results in this dissertation with some general principles and thoughts that we had during the experimentation. Following that will be brief discussions about some limitations of the presented framework and research questions that warrant further investigation.

7.1 Reflections

When humans attempt to understand a concept or perceive an object, one important way is to chunk it into smaller parts, investigate the relation between these different parts and the whole, and study how they are organized to fulfill certain functions. I still remember once in Marvin Minsky's class. He proposed that we could find rigorous definitions of concepts only in a mathematical context. "How could you define a chair? You can say it has four legs, but there are certainly some chairs that do not have four legs. You can say it is something that we can sit on, but there are certainly some other things that we can sit on. There is no way that you can find a rigorous definition of chair." However, we cannot say "chair" is not a valid concept that cannot be understood, though finding its definition might be hard. That means - if I do not understand the English word "chair" and somebody tells me "it has four legs and is something that we can sit on." - I will surely get it. This perhaps can demonstrate how the structure of an object and its relation to its functionality are so important for us to understand the object in general. For music, it is especially important.

During my research into music-related applications, I found the representation is always one of the key issues. Chromagram was mainly used in this dissertation for its direct mapping to standard musical notes. However, it does have some drawbacks that make it inappropriate for some applications. For example, as I mentioned in Section 3.1, chromagram might not be good for pieces or genres that are not well tuned. Even for the classical piano music corpus I used for the key detection task, I believe part of the reason that the algorithm got confused between the true key and its dominant or subdominant key is due to the drawback of the chromagram representation. Since chromagram combines frequency components of the spectrogram, it blurs the details of the original signal. Therefore, I suspect it will not work well for timbre-related applications and that is why, for the music classification tasks, I chose MFCCs instead of chromagram.

Recurrent structural analysis is really a hard task given that we have no prior knowledge about the formal structure of the piece and the recurrent patterns. Without explicit modeling of phrases, the accuracy we can get should be limited. However, I found explicitly modeling musical phrases is an even harder task – the current approach as far as I know considers only the amplitude change as the cue for deciding phrase boundaries, which is apparently too simple and far from robust. An ideal approach could be modeling the harmonic structure and incorporating the prior musical knowledge about the phrase structure, besides considering amplitude change. For vocal music, lyrics could be another cue for indicating phrases. Are these the only cues we can use for identifying phases? Are they all necessary? Is there a minimum set of necessary cues for human listeners to decide the phrase boundaries? These are the questions for music psychologists. I was wondering if there were any human experiments to investigate whether a listener can parse the musical phrases correctly on a musical piece from a completely different culture that he/she is not familiar with its harmonic structure and does not understand the lyrics either. I tried to incorporate my result from harmonic analysis and use it to tune the boundaries for recurrent structural analysis, but could not find any significant improvement of segmentation recall or precision.

The hook analysis was originally a by-product of recurrent structural analysis, but it turned out to be an interesting topic itself after some investigation. One question I was confused by for a long time was whether something memorable is equivalent to something catchy. For musical hooks, I found these two concepts often got interchanged. But this actually causes trouble because to be memorable things need to be repeated, while to be catchy things need to have variety. This seems to suggest the most memorable part of music and the catchiest part are not the same, which is not consistent with our intuition. My hypothesis is that good composers can craftily integrate these two rather contradictory sides in their music. Hooks tend to repeat, but they tend to repeat in different ways – different tempi, different lyrics, different instrumentations, at different keys, etc. – and appear at specific locations, say, the beginning of a section to make a contrast to the previous section.

The result from my human experiment on summarization is very interesting to me. I was actually very surprised with the cultural difference result from my experiment. The Chinese song where the biggest difference occurred is a quite representative piece of Chinese pop music. Section B containing the title of this song is no doubt the hook of this song for Chinese listeners – almost all my Chinese friends can hum the tune of this section while few of them can memorize the tune of Section A. However, the experimental result shows that all non-Chinese subjects chose Section A as the hook of this song. Here are the reasons that came up in my mind. First, lyrics do play a role in hook identification. All the Chinese subjects chose Section B as the hook at least partly because it contains the title of the song, while the non-Chinese subjects had no clue about the words in music. Second, due to the lack of exposure to Chinese pop music for these non-Chinese listeners, they could not distinguish clearly the two sections after listening to the music for one or two times during the experiment and thus they simply marked the first section of the piece (Section A) as the hook. This also seems to suggest that maybe hook identification for human listeners is not as easy as we think of; given a new musical piece of unfamiliar type, we might not be able to identify the hook without listening to it enough times.

The musical salience detection problem came from a game that my previous roommate liked to play with me. She wanted me to listen to a musical piece that I never heard before and guess the composer or the singer. As long as it was by some artists whom I was familiar with, I always got it right. She was very surprised and asked me how. I would tell her something like "Didn't you notice that this singer always pronounces this character a little bit wrong?", or "This phrase sounds like something composed by that composer." I realized that my judgment was typically based on some small parts of music rather than the whole piece. When I saw Berenzweig's paper (2002) of locating vocal part for singer identification, it reminded me of the game with my roommate. Can machines automatically learn those characteristics of the artist from his/her works? Using the vocal part for identifying a singer is still too broad; there should be something even smaller and varying from artist to artist. Using confidence for this task is quite natural maybe. Previous research has attempted to incorporate confidence in the classification tasks to see if it can improve the accuracy, but no one attempted to investigate whether musical parts with high confidence by computation are also the characteristic parts in our mind (which I call it salience in this dissertation). My first experiment with this was actually using Neural Networks and its various confidence measures. Disappointingly, there was no significant improvement of accuracy by selecting frames based on confidence over selecting frames randomly; and there seemed no correlation between the confidence and the salience of music. After that, I started to consider the probabilistic approach, using predictive posterior probability as confidence. The result turned out to be much better. The performance based on confidence is significantly better than random; and the result did show some correlation between confidence and salience of music. There is one thing interesting to me but not explored in this dissertation. Salience of music really depends on specific tasks. If you compare singer A and singer B, you might find the main difference is in their high-pitch range; but if you compare singer A and singer C, you might find the main difference is in their low-pitch range. So, one experiment could be conducted: we have four data sets A, B, C and D. For classification task 1, A and B belong to one class while C and D belong to the other class. For classification task 2, A and C belong to one class while B and D belong to the other class. We

then can investigate how the salient parts of music change and the parts with high confidence change based on different tasks and whether the changes are consistent. I hope my work in this dissertation could be a step toward automatically learning the characteristics of artists from corpora of acoustic musical signals.

7.2 Directions for Future Research

Throughout this dissertation, we have focused separately on four main issues related to analysis of musical structure: tonality analysis, recurrent structure analysis, hook analysis and salience analysis. These are actually all correlated problems, solving any of these problems can benefit solving the others. For example, Chapter 4 showed how key detection could be useful for recurrent structure analysis; Chapter 5 showed that hooks of music can be closely correlated with musical form and structural accentuation. The salience analysis problem seems a little independent from the other three. But, depending on what kind of judgment we are interested in, it can also be related to music segmentation, which can help tuning the boundaries for analyzing recurrent structure or other segmentation tasks. Therefore, the framework presented could be extended to involve more correlations of different components of musical structures.

Furthermore, the problems addressed in this dissertation are by no means comprehensive for analysis of musical structure, though they are all very important problems and might help solve other problems in this area. For example, Chapter 3 showed how chord change detection can be fused with other methods for beat tracking.

Only a limited amount of prior musical knowledge has been incorporated into the system. One reason is the purpose to investigate the capability of the system to derive the statistical properties from the data automatically to mimic amateur musicians and to make the system relatively general. Another reason is that the author is not a professional musician; this makes it hard to collect and encode the musical rules. We can imagine, with more musical knowledge, the accuracy of the system should be able to improve significantly.

The greatest advances in the computational understanding of audio have come using techniques that are primarily data driven. Correspondingly, the most broadly successful applications are ones that have been trained with the greatest variance of training data. In particular, speech-related applications have benefited from a tremendous amount of meticulously annotated training data. Such a corpus is currently not available for machine listening applications. The wider dissemination of musical source materials and recordings will hopefully lead to even greater successes in this area of research.

The dissertation has presented some findings and hints for issues in music cognition: musical memory, attentive listening, musical similarity, etc. Similarity between the behaviors of the system and those of humans (e.g., the system made similar errors on key detection as humans) is not enough to prove that the human mind works in the same way as the computer program does. We will need more carefully designed human experiments to answer the questions about human music perception.

7.3 Concluding Remarks

This dissertation was originally motivated by the author's Master's thesis work of building a query-by-humming system, which allows the users to search for music by humming melodies. With four years passed, this kind of system, though potentially very attractive on the market, has not been quite commercialized as we hoped. One main reason is the lack of robust techniques for processing acoustical musical signals to automatically build a melody database based on musical recordings. One requirement of it is automatic segmentation and summarization of musical signals for efficient indexing and searching. These kinds of techniques should be very useful to other music information retrieval systems as well.

Since music information retrieval is an emerging but new field compared to speech, it is still in its early stage where most current techniques were borrowed from speech processing and understanding fields. We anticipate the major breakthroughs will take place when there are good ways to combine those techniques from signal processing and machine learning, and principles on music perception and cognition. Without a deep understanding of the human mind, the music listening systems would never be really intelligent.

Appendix A

Table A-1: 21 classical piano solo pieces from CD "25 piano favorites"

1.Brahms: Rhapsody In G Minor Op. 79, No. 2

2. *Mozart: Piano Sonata No. 15 In C (I. Allegro)

3. *Schubert: Moment Musical No. 2

4. *Dvorak: Humoresque No. 7

5. Debussy: Prelude 'La Fille Aux Cheveux De Lin'

6. Grieg: 'Butterfly' From Lyrical Pieces

7. Chopin: Etude In C Minor, Op. 10 No. 12 'Revolutionary'

8. Scarlatti: Sonata In E

9. Debussy: Clair De Lune

10. *Rubenstein: Melody In F

11. Rachmaninoff: Prelude In C-sharp Minor, Op. 3 No. 2

12. *Paderewski: Menuett

13. Chopin: 'Military' Polonaise

14. Liszt: Liebestraum

15. *Chopin: Etude In E, Op. 10 No. 3 'Tristesse'

16. Scriabin: Etude, Op. 8 No. 12

17. *Beethoven: Minuet In G

18. *Mozart: Sonata No. 11 In A 'Rondo All Turca'

19. Debussy: Reverie

20. *Schumann: From Kinderszenen (1. Von Fremden Landern Und Menschen)

21. *Chopin: Waltz In D-flat, Op. 64 No. 1 'Minute Waltz'

* denotes the 10 pieces used for tonality and harmony analysis.

Table A-2: 26 Beatles songs from CD "the Beatles"

1.Brahms: Rhapsody In G Minor Op. 79, No. 2

2. Mozart: Piano Sonata No. 15 In C (I. Allegro)

3. Schubert: Moment Musical No. 2

4. Dvorak: Humoresque No. 7

5. Debussy: Prelude 'La Fille Aux Cheveux De Lin'

6. Grieg: 'Butterfly' From Lyrical Pieces

7. Chopin: Etude In C Minor, Op. 10 No. 12 'Revolutionary'

8. Scarlatti: Sonata In E

9. Debussy: Clair De Lune

10. Rubenstein: Melody In F

11. Rachmaninoff: Prelude In C-sharp Minor, Op. 3 No. 2

12. Paderewski: Menuett

13. Chopin: 'Military' Polonaise

14. Liszt: Liebestraum

15. Chopin: Etude In E, Op. 10 No. 3 'Tristesse'

16. Scriabin: Etude, Op. 8 No. 12

17. Beethoven: Minuet In G

18. Mozart: Sonata No. 11 In A 'Rondo All Turca'

19. Debussy: Reverie

20. Schumann: From Kinderszenen (1. Von Fremden Landern Und Menschen)

21. Chopin: Waltz In D-flat, Op. 64 No. 1 'Minute Waltz'

Bibliography

Aucouturier, J. J. and Sandler, M. "Segmentation of Musical Signals using Hidden Markov Models," Proceedings of AES 110th Convention, May 2001.

Aucouturier, J.-J. and Pachet, F. "Finding Songs that Sound the Same," Proceedings of IEEE Benelux Workshop on Model based Processing and Coding of Audio, November 2002.

Balaban, M., Ebcioglu, K., and Laske, O. "Understanding Music with AI: Perspectives on Music Cognition," Cambridge: MIT Press; Menlo Park: AAAI Press, 1992.

Bartsch, M. A. and Wakefield, G. H. "To Catch a Chorus: Using Chroma-based Representations for Audio Thumbnailing," Proceedings of Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk Mountain Resort, NY, 21-24 October 2001.

Berenzweig, A. L. and Ellis, D. P. W. "Locating Singing Voice Segments within Music Signals," Proceedings of Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk Mountain Resort, NY, 21-24 October 2001.

Berenzweig, A., Ellis, D., and Lawrence, S. "Using Voice Segments to Improve Artist Classification of Music," Proceedings of International Conference on Virtual, Synthetic and Entertainment Audio, 2002.

Boltz, M. and Jones, M.R. "Does rule recursion make melodies easier to reproduce? If not, what does?" Cognitive Psychology, 18, 389-431, 1986.

Burns, G. "A typology of 'hooks' in popular records," Popular Music, 6/1, pp. 1-20, January 1987.

Chai, Wei and Vercoe, Barry. "Music Thumbnailing Via Structural Analysis," Proceedings of ACM Multimedia Conference, November 2003.

Chai, Wei and Vercoe, Barry. "Structural Analysis Of Musical Signals For Indexing and Thumbnailing," Proceedings of ACM/IEEE Joint Conference on Digital Libraries, May 2003.

Chai, Wei. "Structural Analysis Of Musical Signals Via Pattern Matching," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, April 2003.

Chai, Wei and Vercoe, Barry. "Melody Retrieval On The Web," Proceedings of ACM/SPIE Conference on Multimedia Computing and Networking, Jan. 2002.

Chai, Wei. Melody Retrieval On The Web. Master thesis. MIT 2001.

Chai, Wei and Vercoe, Barry. "Folk Music Classification Using Hidden Markov Models," Proceedings of International Conference on Artificial Intelligence, June 2001.

Chuan, Ching-Hua and Chew, Elaine. "Polyphonic Audio Key-Finding Using the Spiral Array CEG Algorithm," Proceedings of International Conference on Multimedia and Expo, Amsterdam, Netherlands, July 6-8, 2005.

Dannenberg, R, Thom, and Watson, "A Machine Learning Approach to Musical Style Recognition," Proceedings of International Computer Music Conference, International Computer Music Association, pp. 344-347, September 1997.

Dannenberg and Hu, "Pattern Discovery Techniques for Music Audio," Proceedings of *Third International Conference on Music Information Retrieval*, M. Fingerhut, ed., Paris: IRCAM, pp. 63-70, 2002.

Deliege, I. "Grouping conditions in listening to music: An approach to Lerdahl & Jackend-off's Grouping Preference Rules," Music Perception, 4(4), 325-360, 1987.

Deliège, I., M. Melen, D. Stammers and I. Cross. "Musical schemata in real-time listening to a piece of music," Music Perception **14**(2): 117-160, 1996.

Erickson, Robert. Sound Structure in Music. Berkeley: University of California Press, 1975.

Foote, J. T. "An Overview of Audio Information Retrieval," ACM-Springer Multimedia Systems, vol. 7 no. 1, pp. 2-11, ACM Press/Springer-Verlag, January 1999.

Foote, J. and Cooper, M. "Visualizing Musical Structure and Rhythm via Self-Similarity," Proceedings of International Conference on Computer Music, Habana, Cuba, September 2001.

Foote, J. and Uchihashi, S. "The Beat Spectrum: A New Approach to Rhythm Analysis," Proceedings of International Conference on Multimedia and Expo, 2001.

Foote, J., Cooper, M. and Nam, U. "Audio Retrieval by Rhythmic Similarity," Proceedings of International Symposium on Musical Information Retrieval, Paris, 2002.

Ghias, A., Logan, J., Chamberlin, D. and Smith, B. C. "Query by Humming: Musical Information Retrieval in an Audio Database," Proceedings of ACM Multimedia Conference, San Francisco, 1995.

Goto, M. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds," Journal of New Music Research, Vol.30, No.2, pp.159-171, June 2001.

Haitsma, J. A., Kalker, T., Oostveen, J. "Robust Audio Hashing for Content Identification," Proceedings of Second International Workshop on Content Based Multimedia and Indexing, September 19-21, Brescia, Italy, 2001.

Hsu, J.L., Liu, C.C., and Chen, L.P. "Discovering Nontrivial Repeating Patterns in Music Data," IEEE Transactions on Multimedia, Vol. 3, No. 3, pp. 311-325, September 2001.

Huron, D. "What is melodic accent? A computer-based study of the Liber Usualis," Paper presented at the Canadian University Music Society Theory Colloquium (Calgary, Al-berta), 1994.

Huron, David. "Perceptual and cognitive applications in music information retrieval," International Symposium on Music Information Retrieval, October 23-25, 2000.

Jbira, A. and Kondoz, A. "Multiple Frequency Harmonics Analysis and Synthesis of Audio," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000.

Kim, Youngmoo and Brian Whitman. "Singer Identification in Popular Music Recordings Using Voice Coding Features," In Proceedings of the 3rd International Conference on Music Information Retrieval. 13-17 October 2002, Paris, France.

Kruskal, J. B., and Wish. M. Multidimensional Scaling. Sage Publications. Beverly Hills. CA, 1977.

Lerdahl, Fred and Jackendoff, Ray. A Generative Theory of Tonal Music. Cambridge, Mass.: MIT Press, c1983.

Liu, D., Lu, L., and Zhang, H.J. "Automatic mood detection from acoustic music data," Proceedings of the International Conference on Music Information Retrieval. 13-17 October 2003.

Logan, B. and Chu, S. "Music Summarization Using Key Phrases," Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2000.

Kemp, T., Schmidt, M., Westphal, M., and Waibel, A. "Strategies for Automatic Segmentation Audio Data," In Proc. International Conference on Acoustics, Speech and Signal Processing, 2000.

Kim, Youngmoo and Brian Whitman. "Singer Identification in Popular Music Recordings Using Voice Coding Features." In Proceedings of the 3rd International Conference on Music Information Retrieval. 13-17 October 2002, Paris, France.

Klapuri, A., Virtanen, T., and Holm, J.-M. "Robust Multipitch Estimation for the Analysis and Manipulation of Polyphonic Musical Signals," Proceedings of COST-G6 Conference on Digital Audio Effects, Verona, Italy, 2000.

Laroche, J. "Estimating Tempo, Swing and Beat Locations in Audio Recordings," Proceedings of Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk Mountain Resort, NY, 21-24 October 2001.

Lerdahl, Fred and Jackndoff, Ray. A Generative Theory of Tonal Music. Cambridge, Mass.: MIT Press, c1983.

Martin, K. D. Sound-Source Recognition: A Theory and Computational Model. Unpublished Ph.D. thesis, MIT Media Laboratory, 1999.

McNab, R. J., Smith, L. A., Witten, I. H., Henderson, C. L., Cunningham, S. J. "Towards the Digital Music Library: Tune Retrieval from Acoustic Input," Proceedings of the first ACM international conference on Digital libraries, 1996.

Minka, T. A Family of Algorithms for Approximate Bayesian Inference. PhD Thesis, Massachusetts Institute of Technology, 2001.

Minsky, M. "Music, mind, and meaning," In The Music Machine: Selected Readings from Computer Music Journal, C. Roads, ed. Cambridge MA: MIT Press: 639-656, 1989.

Ockelford, A. "On Similarity, Derivation and the Cognition of Musical Structure," Psychology of Music, Vol. 32, No. 1, 23-74, 2004.

Pachet, F., Aucouturier, J.-J., La Burthe, A., Zils, A., and Beurive, A. "The Cuidado Music Browser: an end-to-end Electronic Music Distribution System," Multimedia Tools and Applications, Special Issue on the CBMI03 Conference, 2004.

Peeters, G., Burthe, A.L., and Rodet, X. "Toward Automatic Music Audio Summary Generation from Signal Analysis," In Proc. International Conference on Music Information Retrieval, October 2002.

Pye, D. "Content-Based Methods for the Management of Digital Music," Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2000.

Qi, Yuan, and Picard, Rosalind W. "Context-sensitive Bayesian Classifiers and Application to Mouse Pressure Pattern Classification," Proceedings of International Conference on Pattern Recognition, August 2002, Québec City, Canada.

Qi, Yuan, Minka, Thomas P., Picard, Rosalind W., and Ghahramani, Zoubin. "Predictive Automatic Relevance Determination by Expectation Propagation," To appear in Twenty-first International Conference on Machine Learning, July 4-8, 2004, Banff, Alberta, Canada.

Rabiner, L. R.; Cheng, M. J.; Rosenberg, A. E. and McGonegal, C. A. "A comparative performance study of several pitch detection algorithms," IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP-24, no.5, 1976, 399-418.

Roads, C. The Computer Music Tutorial. Cambridge: MIT Press, 1994.

Scheirer, E. D. "Music Perception Systems," Proposal for PhD dissertation, MIT Media Laboratory, 1998.

Scheirer, E. D. "Tempo and Beat Analysis of Acoustic Musical Signals," J. Acoust. Soc. Am. 103:1, pp 588-601, Jan 1998.

Shalev-Shwartz, S., Dubnov, S., Singer, Y. and Friedman, N. "Robust Temporal and Spectral Modeling for Query by Melody," Proceedings of SIGIR, 2002.

Sheh, A. and Ellis, D. "Chord Segmentation and Recognition using EM-Trained Hidden Markov Models," 4th International Symposium on Music Information Retrieval ISMIR-03, Baltimore, October 2003.

Soltau, H., Schultz, T., and Westphal, M. "Recognition of Music Types," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA. Piscataway, NJ 1998.

Sterian, A. and Wakefield, G. H. "Music Transcription Systems: From Sound to Symbol," Proceedings of the AAAI-2000 Workshop on Artificial Intelligence and Music, 2000.

Tzanetakis, G. Manipulation, Analysis and Retrieval Systems for Audio Signals. PhD Thesis, Computer Science Department, Princeton University, June 2002.

Whitman, Brian, Gary Flake, and Steve Lawrence. "Artist Detection in Music with Minnowmatch," In Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing, pp. 559-568. Falmouth, Massachusetts, 2001.

Wold, E., Blum, T., Keislar, D., and Wheaton, J. "Content-based Classification, Search, and Retrieval of Audio," IEEE Multimedia Vol. 3 No. 3:Fall 1996, pp. 27-36.

Yang, C. "MACS: Music Audio Characteristic Sequence Indexing for similarity Retrieval," Proceedings of Workshop on Applications of Signal Processing to Audio and Acoustics, 2001.