Towards Automatic Music Structural Analysis: Identifying Characteristic Within-Song Excerpts in Popular Music

By

Bee Suan Ong

Submitted in partial fulfillment of the requirements for
the degree of Diploma of Advanced Studies
Doctorate in Computer Science and Digital Communication
Department of Technology

Tutor: Dr. Xavier Serra

Universitat Pompeu Fabra

Barcelona, July 2005

ABSTRACT

Towards Automatic Music Structural Analysis:
Identifying Characteristic Within-Song Excerpts in Popular Music

Bee Suan Ong

Automatic audio content analysis is a general research area in which algorithms are developed to allow computer systems to understand the content of digital audio signals for further exploitations. The main focus therein is on the practical applications for audio files management, like automatic labeling, efficient browsing, or the retrieval of relevant files with little effort from a big database. Automatic music structural analysis is a specific subset of audio content analysis in which the domain of audio content is restricted to the semantically meaningful descriptions of audio in a musical context. The main task of automatic music structural analysis is to discover the structure of music by analyzing audio signals in order to facilitate a better handling of the current explosively expanding amounts of audio data available in digital collections.

In this research work, we focus our investigation on two areas that are part of audio-based music structural analysis. First, we propose a unique framework and method for temporal audio segmentation at the semantic level. The system aims to detect the structural changes in music to provide a way to separate the different "sections" of a piece according to its structural titles (i.e. intro, verse, chorus, bridge, etc). We present a two-phase music segmentation system together with a combined set of low-level audio descriptors to be extracted form the music audio signals. Contrary to existing approaches, we consider the applicability of image processing methods in audio content analysis. A database of 54 audio files (The Beatles' song) is used for the evaluation of the proposed approach on a mainstream popular music collection. The experiment results demonstrate that our proposed algorithm has achieved 71% of

accuracy and 79% of reliability in a practical application for identifying structural boundaries in music audio signals.

Secondly, we present our proposed framework and approach for the identification of representative excerpts from music audio signals. The system aims to extract a short abstract that serves as a ‘hook’ or thumbnail of the music and generates a retrieval cue from the original audio files. Instead of simply pursuing the present literature that mainly accentuates the repetitiveness of audio excerpts in the identification task, we also investigate the potential of audio descriptors in capturing specific characteristics of the representative excerpts. A database of 28 music tracks that comprises popular songs from various artists is used to evaluate the performance of our identification system. By integrating musical knowledge in selecting appropriate audio descriptors for the identification task, preliminary quantitative evaluation results show that the overall performance of the content-based approaches has achieved a higher performance rate compared to repetition-based approaches.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Xavier Serra, for giving me the opportunity and financial support to work in the Music Technology Group. He has introduced me to the field of music content processing. This work would not have been possible without his help.

I would also like to thank my research project manager, Perfecto Herrera, for his limitless patience in discussing this work with me regardless of his many other commitments. During my study, I have learned a lot from his helpful insights and valuable suggestions.

I want to thank all my colleagues from SIMAC research group, Emilia Gomez, Fabien Gouyon, Enric Guaus, Sebastian Streich and Pedro Cano for providing a very rare kind of stimulating and supportive environment. I am very grateful to their excellent technical advice and insight throughout this work. Without their presence, I would still be wandering.

For moral and emotional support, I would like to thank my dear big sister, Dr. Rosa Sala Rose, for her love, encouragement and for creating such a fun and enjoyable experience throughout my stay in Spain.

Finally, I wish to thank my family back in Malaysia for their unlimited support for my study trip to Spain. Particularly, I wish to express my profound gratitude to my parents, whose love, support and teachings made me that way I am.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER I

INTRODUCTION

For real world popular music, it is common that listeners are apt at recognizing a piece of music by just listening to certain excerpts of the music. For instance, when a prior heard song is played halfway through on the radio, listeners are able to recognize the song without having to go through the whole song from the beginning until the end. According to psychology, it is the retrieval cue in the music that stimulates us to recall and retrieve information in our memory. Human by nature own a remarkable object recognition capability. According to [Roediger05], people can often recognize items that they cannot recall. One example would be the experience of not being able to answer a question but then recognizing an answer as correct when someone else supplies it. In music context, even musically untrained people are able to recognize or at least determine whether they have heard a given song before without much difficulty. Amazon's 30 first seconds of sound samples is a practical example of using a short segment from the original audio files to review or recognize a piece of music.

In this work, we undertake a study of analyzing musical structure with the aim to identify characteristics within-song excerpts from the perspective of content-based analysis. Repetitions and transformations of music structure create the uniqueness of identity for each music piece. Thus, we hypothesize that identification of these transformations and the generation of a semantic level of music structural description will significantly contribute to better handling of audio files. We do not attempt to investigate all kinds of music (i.e. classical, jazz, ethnic, to name a few) but only focus in "pop" music. Unlike much previous work in structural analysis, we make no attempt in tackling this matter based on symbolic notated music data (i.e. MIDI) but on the actual raw audio. Hence, we rely on the particular characteristics of audio features in music content to perform structural analysis on music audio signal.

Our work contributes in a number of areas in music audio retrieval. In audio segmentation task, we present a novel two-phase approach to detect the significant structural changes in audio content. In order to extract content descriptions that are significant in describing structural changes in music, we propose a combination set of low-level descriptors computed from audio signals. In addition, we also introduce the application of image processing filtering techniques for facilitating a better segment boundaries detection. Finally, we compare our method's segmentation performance with a previous method described in [Chai03c] to evaluate the efficiency of our proposal and apparently it seems to perform better. In identifying representative audio excerpts of music, we take into consideration on the potential of audio descriptors in capturing the specific features of the 'hook' or 'gist' in music. Thus, we experience in using various low-level audio descriptors to facilitate the selection of representative excerpts of music. Finally, coarse quantitative evaluations are used to measure the efficiency of our proposed method.

1.1. Context

This work has been carried out at the Music Technology Group (MTG) [1]of the Audiovisual Institute (IUA)[2], Universitat Pompeu Fabra (UPF)[3] in Barcelona. The MTG, founded in 1994 by its current director, Dr. Xavier Serra, is a research group specialized in audio processing techniques and its music and multimedia applications. Areas of research and developments projects carried out within the MTG include:

- Audio content analysis
- Audio identification;
- Audio processing and synthesis;
- Interactive systems;
- Music performance;
- Singing voice processing;
- Software tools

[1] http://www.iua.upf.es/mtg
[2] http://www.iua.upf.es
[3] http://www.upf.es

Among the above listed research areas, audio content analysis is currently one of the core research area in the MTG. SIMAC[4] (acronym for Semantic Interaction with Music Audio Contents), a European Commission funded project, is one particular example of a framework that carries out audio content analysis research in the MTG. SIMAC is about doing innovative research and development on digital music distribution services related to automatic extraction and organization of semantic descriptors of music content. Its key feature is the usage and exploitation of automatically extracted semantic descriptors of musical content. So far, three prototypes (i.e. for music exploration, recommendation and retrieval) have been developed in the SIMAC project to step beyond music information retrieval and move towards the realm of music content discovery. Three axes of research and technological development in SIMAC are:

- Semantic descriptors of music
- Similarity in Music
- Musical structure

SIMAC project has covered different facets of the scientific, technical and commercial skills involved in the area of music technologies by merging with internationally recognized research centers and industrial partners. This can be seen from the project documents, scientific publications and other dissemination texts generated in the SIMAC consortium.

1.2. Author's background

Listed below is a brief summary of the author's academic trajectory.

I graduated both my undergraduate (BMus) and postgraduate studies (MSc), majoring in Music Technology, from University Putra Malaysia (Malaysia) in 2000. After graduated from my studies, I joined the Centre for Signal Processing at Nanyang Technological University (Singapore) as a researcher, working on audio signal processing. In 2002, I was recruited by the Agency of Science, Technology and Research of Singapore (A*STAR) as a research engineer, working on the same topic of research, at the Laboratories for Information Technology [5](currently known

[4] http://www.semanticaudio.org

[5] http://www.i2r.a-star.edu.sg

as “Institute of Infocomm Research - I2R”), National University of Singapore. By the end of 2002, I was matriculated in the PhD program in Computer Science and Digital Communication of the Universitat Pompeu Fabra. Since then, I have joined the Music Technology Group (MTG) and am currently participating in the SIMAC project working on research related to music audio content, specifically music structural analysis. Personally, I enjoy listening to music. Throughout these years, I have downloaded quite a big amount of mp3 files in my personal music audio collection. Among these downloaded audio files, there are many which contain improper and inconsistent labeling. Thus, my personal bad experience of having difficulty in searching songs in my own collections has influenced me to choose a research topic related to music content analysis for my PhD study.

1.3. Motivation and Goal

As we enter a new advanced technology era, the explosion of multimedia content in databases, archives and digital libraries has caused some problems in efficient retrieval and management of this data content. Under these circumstances, automatic content analysis and processing of multimedia data becomes more and more important. In fact, content analysis, particularly content understanding and semantic information extraction have been identified as important steps towards a more efficient manipulation and retrieval of multimedia content.

Music is highly structured and the perception and cognition of music rely on inferring structure to the sonic flow heard [Weyde03]. In fact, music structure varies widely from composer to composer, from piece to piece and from listener to listener. Thus, music structural analysis, which aims to compute a representation of the semantic content of a music signal through discovering the structure of music, is believed to be able to provide a powerful way of interacting with audio content (i.e. browsing, summarizing, retrieving and identifying) and facilitates a better handling of music audio data.

In this research, we aim to provide an efficient methodology towards automatic audio-based music structure analysis. Alike many tractable problems in content analysis, focused in identification of salient features of a particular selection, we attempt to identify “singular” within-song excerpts in popular music. We have focused our investigation in two areas that are closely related:

(i) Semantic audio segmentation
(ii) Identification of representative excerpts of music audio

Semantic audio segmentation attempts to detect significant structural changes in music audio signals. It aims to provide a direct access to different “sections” of a popular music track, such as “intro”, “verse”, “chorus” and so on. In contrast, the identification of representative excerpts of music audio aims to recognize the significant audio excerpts, which represent a whole piece of music. The significant excerpts may consist of the most repetitive segments or even the most outstanding or “attention-grabbing” segments that are usually not repeated but are capable of leaving a strong impression in our mind. The goal behind representative excerpts identification is to generate a “hook”, thumbnail or cue abstraction of the music that would give listeners to have an idea of a piece of music without having to listen to a whole piece. This would be very much useful in facilitating time-save browsing and retrieval of music, since it saves a considerable amount of time and thus speeds up the iteration.

It is important to note that in this study, we do not deal with all kinds of music. Here, we are only interested in the structural analysis of pop music. Thus, other music genres, such as classical, jazz, ethnics and so forth, will not be included in our research study area. Music can be represented in two different manners: symbolic representation and acoustic representation. The symbolic representation is based on score-like notation of music. Thus, only a limited set of music material (e.g. pitches, note duration, note start time, note end time, loudness, etc.) involves in music representation. Examples of such representation include MIDI and Humdrum [Selfridge97, Huron99]. The acoustic representation is based on the sound signal itself. Thus, acoustics representations can represent any sound in the natural world. Different storage formats and different lossy compression standards have leaded to different formats for acoustics representations of music. They are such as wav-, au-, or mp3-files to name only a few. The task of automatic music structural analysis can be accomplished using either of these music representations. However, considering the prevalent usage of acoustic representation in representing popular music and the task of converting acoustic representations to symbolic representations of music is currently still an open issue, we concentrate our study dealing with acoustic representations instead of symbolic representation of music.

In the following section, we review the potential of music structural analysis for a variety of application domains.

1.4. Applications

1. One of the primary applications for music structural analysis is the production of music structural descriptors for music content exploitation. It is believed that music structural descriptions, which subsume temporal, harmonic, rhythmic, melodic, polyphonic, motivic and textual information, may improve efficiency and effectiveness in handling huge music audio databases, as repetition and transformation of music structure create a uniqueness identity of music itself. For instance, music can be classified according to the similarity or difference of its structural descriptions.
2. Music structural analysis has also some applications for facilitating time saving browsing of audio files. Being able to provide higher semantic information from audio files would offer users some clues regarding whereabouts the structural changes of audio occur (i.e. from "Intro" →"Verse" →"Chorus", etc.). This would allow users to grasp the audio content through scanning the relevant segments. For example, an audio player with a functionality of allowing users to skip from one section to another section would definitely reduce the browsing time of assessing large amount of retrieved audio [Goto03b].
3. Repetition in music is one of the crucial elements in extracting a short abstract or generating a retrieval cue from the original audio files. Seeing that structural analysis holds a high potential in revealing the repeated patterns in music, it has also extended the applications for music summarization and thumbnailing.
4. Coupling music structural analysis functionality into music annotation tools would offer users with an initial audio annotation to the user. For the case of manual annotation, annotators could take profit of this initial annotation information from the system and make further adjustments or expansions of it. Without doubt, this would enhance the annotation processes.

5. The generation of higher-level semantic information of music audio may also provide an additional measuring or comparing dimension for music recommendation systems in finding music with similar characteristics. The systems can tailor users' preferences based on simplicity or complexity of the music structure in the users' own collections.
6. Besides the usefulness in generating an abstract of the original audio files through music summarization, music structural analysis would also contribute in offering an interactive multimedia presentation that shows "key-frames" of important scenes in music and allow users to interactively modify the summary. For instance, users can create a mega-tune comprising the remix of all the choruses from their favourite artists.
7. Finally, automatic music structural analysis may serve as a valuable tool for musicological analysis of computer music. Music analysis is one of the main subjects in the musicological field. Traditionally, musicologists analyze music with the aid of music notation scores and music theory. However the lack of common musical notation scores of computer music has created a complex issue in musicological analysis of computer music. So far, there exist two approaches in musicological analysis of computer music. One approaches music from the viewpoint of perception and another from the point view of creativity. Both approaches aim to describe the listening in order to understand the structure of music. Thus, automatic music structure analysis, which is based on audio content, may facilitate in justifying the listening for musicological analysis of music,

1.5. Scope

In this work, we examine two tasks of music structural analysis: (i) semantic audio segmentation; (ii) identification of representative excerpts of music audio signal. Two separate systems have been developed to automatically perform each task. Both of them accept music audio as input signal. The segmentation system outputs a text file in ASCII format, which indicates the detected segment boundaries with a temporal resolution of 0.01 sec. The identification system outputs the transcription files in the lab-file format (used by WaveSurfer[6], an Open Source tool for sound visualizing and manipulation, to transcribe sound files). These transcription files comprise the beginning and ending time of the repeated sections together with their

[6] http://www.speech.kth.se/wavesurfer/

given labels marking the repeated sections (ex: A, B, C, etc.). One audio input may yield more than one transcription file since there may occur various repetition patterns in a piece of music. In addition, the system also outputs an audio excerpt, which contains the most representative excerpt of the input music signal.

1.6. Thesis Outline

The remainder of this work is organized in the following manner.

Chapter II reviews related literature in automatic music structural analysis. This chapter begins with a brief overview of multimedia content analysis and proceeds by literature review sections. We include in this chapter a discussion regarding the pros and cons of each approach found in the literature for discovering the structure of music.

Chapter III presents our approach for semantic audio segmentation corresponding to the structural changes in music. It begins by giving an outline of our proposed method and this is followed by its full description. This chapter includes quantitative evaluation of the system's performance based on a test set. All experiments involve the use of polyphonic music from audio recordings of popular songs.

Chapter IV attempts to identify representative excerpts in music audio signals. This chapter first lays down the framework of our approach. It is then followed by a detail description of our approach. We examined the performance of our proposed method based on different assumptions used in identifying representative audio excerpts using a test set of popular songs from various artists. The final section of this chapter includes the discussion of the quantitative evaluation results.

Finally, Chapter V draws conclusions and examines potential directions for future work leading towards PhD level.

CHAPTER II

LITERATURE REVIEW

In this chapter, we present a literature review focused on the topic of this thesis. It starts with a general overview of multimedia content analysis that relates to music content analysis. Then we proceed to a general overview of music content analysis. Following this, we review the research that is directly related to music structural analysis. Current research works in music structural analysis can be classified into two main approaches: the audio-signal approach versus the symbolic representation approach. The audio-signal approach deals with the actual raw audio file whereas the symbolic representation approach deals with music symbolic notation data (i.e. MIDI). Here, we focus our literature review on the audio-signal approach rather than on the symbolic representation. Feature extraction is an indispensable process in music content analysis. Thus, we devote some space to present the different extracted feature attributes considered in the literature. Audio segmentation facilitates truncation of audio signals for further analysis. In fact, it seems to be an indispensable procedure in certain content-based analysis. Here, we review work relevant to segmenting audio signals for further structural analysis. Music structural discovery towards the direction of identifying representative excerpts of music is a key issue in this thesis. Thus, in the last section of the literature review, we will focus on relevant approaches for the identification task and the pros and cons of each approach used in the identification task.

2.1. Multimedia content analysis

With the rapid increase in electronic storage capacity and computing power, the generation and dissemination of digital multimedia content experiences a phenomenal growth. In fact, multimedia is pervasive in all aspects of communication and information exchange even through internet networking. Efficient management and retrieval of multimedia content have become the key issue especially for large distributed digital libraries, databases and archives. Traditional search tools are built upon the success of text search engines, operating on file names or metadata in text format. However these have become useless when meaningful text descriptions are not available [Cheng03]. Apparently, large indexing of multimedia content based on human efforts may lead to incoherent descriptions by different indexers and errors caused by carelessness. This would create a hassle when search on improper indexed multimedia databases using text descriptions. Thus, a truly content-based retrieval system should have the ability to handle those flaws caused by text descriptions. So far, much research has been focusing on finding ways of analysis and processing to effectively handle these enormous amounts of multimedia content. In this context, multimedia content analysis, which aims to compute semantic descriptions of a multimedia document [Wang00], holds a tremendous potential.

The term "media" encompasses all modalities of digital content, such as audio, image, language. Video and so forth, which is used in entertainment, broadcasting, military intelligence, education, publishing and a host of other applications, represents a dynamic form of media. Digital video is a composite of image, audio, language and text modalities [Smith04]. Generally, video contains a complex assortment of audio, visual and textual information. So far, content-based analysis of video has been a fast emerging interdisciplinary research area. Prior video content-based analysis used physical features, such as colour, shape, texture and motion for frame characterization and later on scene recognition using similarity between frame attributes to study its content. Current video content-based analysis makes use of audio information included in video to facilitate better content descriptions. The exploration of the significance of audio characteristics in semantic video content understanding has led audio content to be associated with video scene analysis, such as video segmentation, scene content classification and so forth, to facilitate easy browsing [Adam03]. In fact, audio content-based analyses are important processes in video characterization that aims to preserve and communicate the essential content of the video segments via some visual representation [Smith04]. Most characterization techniques use the visual stream for temporal segmentation and the audio stream is then analyzed for content classification [Nam97, Wang00]. The development of MPEG-7 is an ongoing effort by the Moving Picture Experts Group to standardize such relevant features or metadata available for efficient characterization and descriptions of multimedia content. In this context, MPEG-7 holds a high potential in a variety of application domains: large distributed digital libraries, digital and interactive video, multimedia directory services, broadcast media selection and multimedia authoring.

Recently, much project work in video content-based analysis has been focusing on video content understanding in order to facilitate efficient browsing of video content for better indexing and retrieval of video data. Here, browsing entails viewing a large collection of videos in a short time with minimal loss in information. In this context, it is necessary to represent video in a more abstract or summarized manner. Video abstraction and visualization techniques, which include text titles and single thumbnail images or an ordered set of representative thumbnail images (key frames) simultaneously on a computer scene to present a synopsis of the original video, have been used to represent video in a compact manner for the purpose of time-saved assessing of video content. Video summarization and video skimming, which aim at providing an abstract of a video for shortening the navigation and browsing the original video, are a few examples of video content-based analysis working towards this direction. For video abstract presentation, video summarization uses a continuation of image sequences while video skimming uses still-images [Mulhem03].

2.2. Music Content Analysis

With the advance of compression technology and wide bandwidth of network connectivity, the existence of music downloading services on the internet blossoms. The availability of these services has made it possible for computer users to store many music files that he/she has only once or even never listened to. For instance, Napster, which offers over 700,000 songs, 70,000 albums and 50,000 artists to be downloaded for offline listening, is still adding new music to its database every day with new release from all of the four major music labels in the world, such as Sony/BMG, EMI, Warner Music Group and Universal Music Group. Apparently, the rapid increase of music collections has created difficulties for administrating these audio data. Retrieving a song without knowing its title from one of these huge databases would definitely be a difficult task. From this, we can see that the traditional way of music indexing and retrieval is no longer able to handle these huge databases. Thus, content-based analysis is believed to be able to facilitate efficient handling of these huge amounts of digital audio data. Similar to video content analysis, current music content analysis works focuses on generating semantic descriptions of the music that is contained in an audio file.

Apparently, structure of music is an important aspect of it. Repetitions, transformations and evolutions of music structure create a uniqueness identity of music itself. The uniqueness of music structure can be seen through the use of different musical forms in music compositions. For instance, western classical sonata music composers used structural functions, such as exposition, transition, development or termination, to shape the music. This is very different from the present popular music compositions, which are much shorter in length and use much simpler structural forms. In fact, laying out the structure plan (in mind or on a piece of paper) is a prerequisite for most music composers before starting to compose their own music. Thus, it is believed that music structural descriptions, which subsume temporal, harmonic, rhythmic, melodic, polyphonic, motivic and textual information, may improve efficiency and effectiveness in handling huge music audio databases. Moreover, such structural description can also provide a better quality access and powerful ways of interacting with audio content, such as better quality audio browsing, audio summarizing, audio retrieving, audio fingerprinting etc., which would be very useful and applicable for music commercial and movie industries.

Limitation of human memory makes us incapable to recall every single detail of all incidents that happen in their daily life. As human beings, we may only recall certain events, which have created a “strong” impression in our mind. The same happens with music, we do not recall the music that we hear in its entirety but through a small number of distinctive excerpts that have left an impression on our mind. It is usually the case that we only need to listen to one of those distinctive excerpts in order to recall the title for the musical piece, or, at least, to tell if we have heard this song before. Seeing this, much research in current music structural analysis is focusing on identifying representative musical excerpts of audio signals.

2.3. Research in Automatic Music Structural Analysis

In the following sections, we explore several research works directly related to automatic audio-based music structural analysis in more detail, with a particular focus on discovering structure descriptions. These related automatic structural analysis research works either form as the basis for other studies (i.e. music summarization) or as the subject of study in itself. We begin with a discussion of audio features that are commonly used in music structural analysis literature. It is then followed by the review of audio segmentation approaches aiming at a better truncation of the audio signal for further structural processing. Finally, we discuss a variety of identification techniques to discover the structure of music for further exploitations.

2.3.1. Audio Features

In music content analysis, proper selection of audio feature attributes is crucial to obtain an appropriate musical content description. For music structural analysis, it is important to extract a kind of music representation that is able to reveal the structural information from the audio signal. Extracting symbolic score-like representation from music could be a possible way to complete the task of music structural analysis. However due to the demanding constraints in extracting symbolic score-like representation from polyphonic music, this approach is practically infeasible. Instead, extracting low-level representations from the audio signal for musical content description is found to be an alternative way for completing this task. Lately, low-level audio feature attributes, which describe the musical content of a sound signal, have been widely used in research works closely related to music structural analysis, such as audio segmentation or boundary detection, audio thumbnailing, chorus identification, music summarization, pattern analysis of music, etc. In automatic audio-based music structural analysis related works, feature attributes are often computed on a frame-by-frame basis in order to obtain the short-term descriptions of the sound signal. The music signal is cut into frames. For each of these frames, a feature vector of low-level descriptors is computed. In accordance with the similarities and differences of the generated content descriptions, these feature attributes can be roughly classified into three groups: timbre-related features, melody-related features, and dynamics-related features. Figure 2.1 illustrates the overall taxonomy of features.

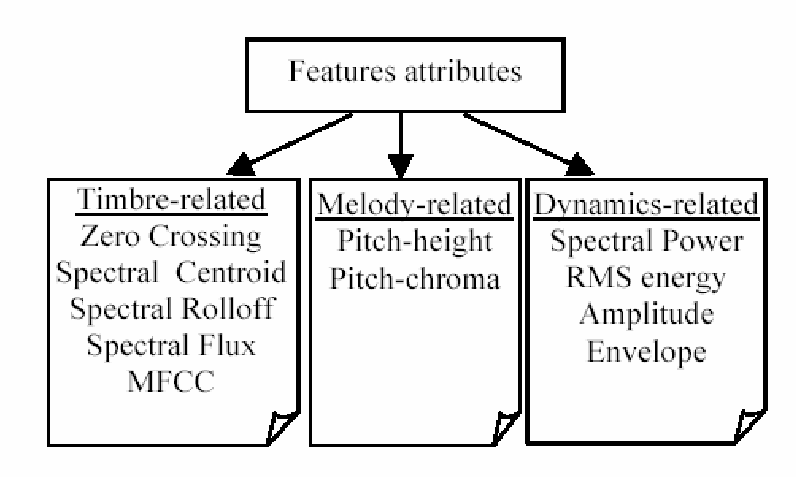


Figure 2.1. Illustration of categories of feature attributes

Timbre-related features

Timbre content descriptions are of general importance in describing audio. Most of the existing research work uses timbre content descriptions in order to differentiate music and speech besides music classification applications. Hence, a lot of timbre-related features have been proposed in this research area [Tzanetakis99]. In fact, timbre-related features are the most widely used among the three groups mentioned above. So far, the most employed timbre-related features are:

Zero Crossings: A measure of the number of time-domain zero crossings within a signal. It gives an approximate measure of the signal's noisiness.

$$Z_t = \frac{1}{2} \sum_{n=1}^{N} | sign(x[n] - sign(x[n-1])) | \tag{2.1}$$

where *sign* function is 1 for positive *x[n]* and –1 for negative *x[n]* while *t* denotes the frame number.

Spectral Centroid: A representation of the balancing point of the spectral power distribution within a frame that is computed as follows:

$$SC = \frac{\sum_k kX[k]}{\sum_k X[k]} \tag{2.2}$$

where *k* is a correspond index to a frequency bin, within the overall measure spectrum, and *X[k]* is the amplitude of the corresponding frequency bin.

Spectral Rolloff: A measure of the frequency, below which 95 percentile of the spectral energy are accumulated. It is a measure of the "skewness" of the spectral shape – the value is higher for right-skewed distributions

$$SR = K, \text{where} \qquad \sum_{k<K} X[k] = 0.95 \sum_{k} X[k] \qquad (2.3)$$

Spectral Flux (also known as Delta Spectrum Magnitude): A measure of spectral difference, thus it characterizes the shape changes of the spectrum. It is a 2-norm of the frame-to-frame spectral magnitude difference vector

$$SF = || X[k] - X[k-1] || \qquad (2.4)$$

where *X[k]* is the complete spectral magnitude of a frame.

MFCC, also called Mel-Frequency Cepstral Coefficients [Rabiner93]: A compact representation of an audio spectrum that takes into account the non-linear human perception of pitch, as described by the Mel scale. It is the most widely used feature in speech recognition. Currently, much research has focused in using MFCC to automatically discover the structure of music. [Aucouturier02, Xu02, Steelant02, Logan00, Peeters02, Foote99, Cooper02]. MFCC is particularly useful for analyzing complex music due to its low-dimensionality, smooth version of the log spectrum, the ability to discriminate between different spectral contents [Steelant02] and to somehow discard differences due to pitch evolution. MFCC calculation can be done through the following steps [Rabiner93]:

1. Convert signal into short frames
2. Compute discrete Fourier transform of each frame
3. The spectrum is converted to the log scale
4. Mel scaling and smoothing the log scale spectrum

5. Discrete cosine transform is calculated (to reduce the spectrum to 40 coefficients)

Harmonic or Melody-related features

Melody, together with harmony, rhythm, timbre and spatial location makes up the main dimension for sound descriptions [Gómez03]. With the implicit information that it carries, melody plays an important role in music perception and music understanding. According to Selfridge-Field [Selfridge98], it is the melody that makes music memorable and enables us to distinguish one work from another. Current research in music content processing such as music transcription, melody similarity, melodic search, melodic classification and query-by-humming, works closely with melodic information. So far, there are several ways of defining and describing a melody. Solomon [Solomon97] and Goto [Goto99, Goto00] define melody as a pitch sequence. While some others define music as a set of attributes that characterize the melodic properties of sound, a set of musical sounds in a pleasant order and arrangement etc. [Gómez03]. Among those viewpoints, melody as a pitch sequence would be the most appropriate for finding repetitions of music with the aim to discover music structure.

In pitch perception, humans recognize pitch as having two dimensions, which refer to pitch height and pitch chroma, respectively. Pitch chroma embodies the perceptual phenomenon of octave equivalence, by which two sounds separated by an octave (and thus relatively distant in term of pitch height) are nonetheless perceived as being somehow equivalent. Therefore, pitch chroma provides a basis for presenting acoustic patterns (melodies) that do not depend on the particular sound source. In contrast, pitch height varies directly with frequency over the range of audible frequencies. Hence, it provides a basis for segregation of notes into streams from separated sound sources. The function of these two pitch dimensions is illustrated when the same melody is sung by a male or a female voice [Warren03].

In the music structural analysis and processing domain, melody-related features have been widely used in identifying repetitive patterns or representative excerpts of music. According to the dimension they focus on, we can consider two approaches in extracting melody-related features. The first one focuses on the pitch-height dimension. This approach uses features that carry pitch-height information to find repetitive patterns of music. Dannenberg and Hu, [Dannenberg02b] use this approach to estimate pitch and identify the note boundaries of monophonic music. The authors compute the correlation between the signal and a time-shifted version of it. Finally, the fundamental pitch is selected based on several heuristics rules. This approach is only applicable for single pitch monophonic music. However, for real-world polyphonic music with a complex mixture of pitches, extracting the predominant one is highly complicated and practically infeasible with current methods. Sound source separation, which aims to separate a sound mixture, could be a possible way to facilitate in extracting predominant pitch of music. However due to the immaturity of our present sound source separation technologies, extracted pitch height information from polyphonic music would still be very unreliable.

The second approach focuses on the pitch-chroma dimension and thus uses features that carry pitch-chroma information. Pitch-chroma holds the information related to the harmony or the melodic content of music and it captures the overall pitch class distribution of music [Goto03a], the description it yields can be similar even if accompaniment or melody lines are changed to some degree. With this unique characteristic of pitch-chroma, there is no constraint of using this approach to analyze polyphonic music. In fact, the application of harmonic or melodic content-related information in music content processing is not a novel strategy. The pitch histogram proposed by Tzanetakis [Tzanetakis02] for measuring similarity between songs would be an example. Tzanetakis's pitch histogram is composed of a set of global statistical features related to the harmonic content. This set presents the most common pitch class used in the piece, the occurrence frequency of the main pitch class, and the octave range of the pitches of a song.

In their research on the identification of representative musical excerpts research, several authors [Goto03a, Dannenberg02a, Birmingham01, Bartsch01] have employed chroma-based vectors to find the repetitive patterns of music. A chroma-based vector is basically an abstraction of the time varying spectrum of audio. It is computed mainly through restructuring a sound frequency spectrum into a chroma spectrum. Octave information is discarded through folding frequency components in order to fall into twelve distinct chroma bins which correspond to the twelve pitch classes [Dannenberg02a]. Bartsch and Wakefield [Bartsch01] perform autocorrelation to the chroma-based vector in order to identify the song extract, which holds the most repeated "harmonic structure". With a different formulation, Goto's [Goto03a] RefraiD method employs a 12-element chroma-based vector similar to the one that is used in [Bartsch01], in order to analyze relationships between various repeated sections, and finally detecting all the chorus parts in a song and estimating their boundaries.

Dynamics-related features

In human auditory perception, loudness contrast captures listeners' ears. The musical term "dynamics", which refers to relative loudness or quietness measurement of the sound, holds a significant role in expressive musical structure formation. In music composition and music performance, artists use dynamics to emphasize and shape the structure of music. Current research studies in music expressive performance analyze dynamics behaviour to evaluate the expressiveness of the performance. A real-time expressive music performance visualizing system, based on tempo and loudness spaces, has been built to help studying performance expressiveness. It depicts the dynamics and tempo behaviour of each performance done by different interpreters on the same piece of music [Widmer03]. Considering the significance of music dynamics in marking the occurrence of new music events, dynamics-related features have become unique and useful in music segmentation. When finding repetitions in music, proper identification of dynamics-based repetition boundaries is highly significant. So far, three dynamics-related features frequently appear in the existing work: Spectral Power, RMS and amplitude envelope.

Spectral power: For a music signal *s(n)*, each frame is weighted with a window. [Xu02] weights each frame signal with a Hanning window that is defined as *h(n)*:

$$h(n) = \frac{\sqrt{8/3}}{2}[1-\cos(2\pi\frac{n}{N})] \tag{2.5}$$

where N is the number of the samples of each frame.

$$SP = 10\log_{10}\left[\frac{1}{N}\left\|\sum_{n=0}^{N-1} s(n)h(n)\exp(-j2\pi\frac{n}{N})\right\|^2\right] \qquad (2.6)$$

RMS energy [Tzanetakis99, Steelant02]: A measure of physical loudness of the sound frame

$$RMS = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1} x[k]^2} \qquad (2.7)$$

where N is the number of samples in each frame.

Amplitude Envelope: A description of the signal's energy change in the time domain. [Xu02] computes the signal envelope with a frame-by-frame root mean square (RMS) and a low 3rd order Butterworth lowpass filter [Ellis94] with empirically determined cutoff frequencies.

2.3.2. Feature Extraction Approach

So far, there exist two approaches in using the above mentioned low-level feature attributes to obtain useful descriptions for music structure discovery: the static one and the dynamic one. The static approach computes low-level descriptions directly from the sound signal to represent the signal around a given time. Hence, in order to detect repetitive patterns in music, it is essential to find feature sequences with identical evolution. The dynamic approach, proposed by Peeters et al. [Peeters02], uses features that model directly the temporal evolution of the spectral shape over a fixed time duration. The difference between the two approaches is that the earlier one uses features that do not model any temporal evolution and only provide instantaneous representations around a given time window (i.e. only the successive sequence of the features models the temporal evolution of the descriptions).

Following the static approach, Steelant et al. [Steelant02] propose the use of statistical information of low-level features, instead of the features themselves, to find the repetitive patterns of music. These statistics are mainly the average and the variance of the instantaneous features over the whole signal. According to Steelant et al., global representations of the low-level features, which consist of their statistical information, can overcome the problem of very similar passages having different extracted coefficients, due to the large frame-step during feature extraction process. In their research to find the repetitive patterns of music, they use feature sets, which contain mean and standard deviation of MFCCs. Their algorithm, tested on a database of only 10 songs, showed a slight improvement when using the statistical information of the low-level features instead of using the frame by frame features.

On the dynamic approach side, Peeters et al. [Peeters02] compute dynamic-features by passing the audio signal, $x(t)$ through a bank of N Mel filters. Short-Time Fourier Transform (STFT) with window size L is then used to analyze the temporal evolution of each output signal $x_n(t)$ of the $n \in N$ filters. The transformed output, $X_{n,t}(w)$, models directly the temporal evolution of the spectral shape over a fixed time duration. According to Peeters et al., the window size that is used for STFT analysis determines the kind of music structure (i.e. *short-term* or *long-term*) that can be derived from the signal analysis. Even though this approach may greatly reduce the amount of used data, the advantage is only noticeable when one deals with a high dimensionality of feature attributes.

2.3.3. Audio Segmentation

Music structural discovery from audio signals was first inspired by the works on signal segmentation first developed in speech applications, such as "SpeechSkimmer" [Arons93], and were later adapted for musical applications. Thus, signal segmentation is closely associated with music structural discovery. In fact, signal segmentation, which facilitates partitioning audio streams into short regions for further analysis, is an indispensable process in music structure discovery. Finding appropriate boundary truncations is crucial for certain content-based applications, such as audio summarization, audio notation and etc. In this section, we will discuss different methods implemented for segmenting audio signals for later structural identification. In addition, we have grouped the methods according to their similarities and differences regarding implementation (i.e. model-free segmentation versus model-based segmentation).

In discovering structure of music, we can distinguish between two segmentation processes: pre-analysis segmentation and post-analysis segmentation. As shown by its name, pre-analysis segmentation (sometimes also called frame segmentation) takes place before the content analysis process. In fact, pre-analysis segmentation is a crucial primary step for content analysis description. Normally, it partitions audio streams into fixed-length short regions for later content analysis. These short regions may sometimes partially overlap. As arbitrary fixed resolution segmentation of audio streams may cause unnatural partitions, high-level audio descriptions (such as, beat or note-onset information) could be useful in finding natural segmentation points and improve the overall pre-analysis segmentation performance [Bartsch01].

On the other hand, post-analysis segmentation appears subsequent to the content analysis process. The aim of this segmentation process is to identify appropriate boundaries for partitioning the audio streams into sections. These sections comprise a non-fixed number of successive short regions being the output from earlier segmentation processes (as shown in Figure 2.2), based on their feature changes. Hence, the partitions we obtain using post-analysis have a longer duration than those from pre-analysis segmentation. Post-analysis segmentation assumes that the boundaries between two consecutive partitions should consist of abrupt changes in their features' contents. Meanwhile, the feature values of the signal inside each partition are supposed to vary little or slowly (i.e. are homogenous). Since appropriate boundary truncations are rather significant for music structure, this segmentation process holds an important role in automatic music structural analysis.

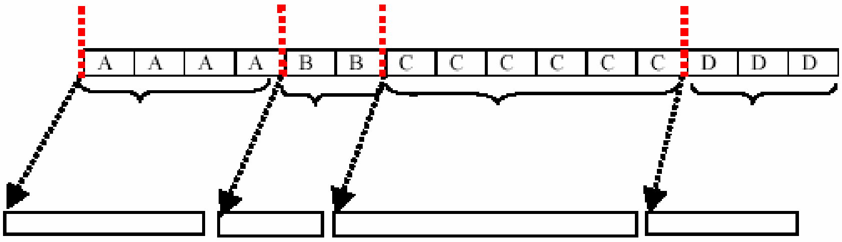


Figure 2.2. Illustration of post-analysis segmentation

Post-analysis segmentation strategies can be categorized into two groups, according to the similarities and differences in their implementations. Hence, we speak of model-free segmentation and of model-based segmentation. Model-free segmentation algorithms partition signals without requiring any training phase. An example of model-free post-analysis segmentation method used in automatic music structure analysis are similarity measures [Bartsch01, Steelant02, Cooper02, Cooper03, Goto03a, Lu04, Bartsch05]. In the case of model-based segmentation, a training phase is necessary in order to learn the models for segmenting. The model is built, by using a collection of examples, which correspond to the desired output from the segmentation algorithm, as training samples. Hidden Markov Models (HMM) [Logan00, Aucouturier02] are an example of the model-based post-analysis segmentation method used in music structure analysis.

Model-free Segmentation

A widely used model-free segmentation technique takes advantage of (dis)similarity measures [Foote00, Bartsch01, Steelant02, Cooper02, Peeters02, Cooper03, Goto03a, Lu04, Bartsch05]. Foote [Foote99] first proposed the use of local self-similarity in spotting musically significant changes in music. It is done by measuring the distance between feature vectors using Euclidean distance or the cosine angle between the parameter vectors. The similarity matrix is a two-dimensional representation that contains all the distance measures for all the possibilities of frame combinations. As every frame will be maximally similar to itself, the similarity matrix will have a maximum value along its diagonal. In addition, if the distance measure is symmetric, the similarity matrix will be symmetric as well. With the use of a cosine metric, similar regions will be close to 1 while dissimilar regions will be closer to –1. According to Foote, by correlating a similarity matrix, *S*, with a checkerboard kernel, which is composed of self-similar values on either side of the centre points and of cross-similarity values between the two regions, along the diagonal of the similarity matrix, it yields the time instant of audio novelty $N(i)$, which is useful for identifying the immediate changes of audio structure. A simple 2x2 unit kernel, *C,* that can be decomposed into "coherence" and "anticoherence" kernels is shown in equation 2.8 below.

$$C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.8}$$

Audio novelty can be represented by

$$N(i) = \sum_{-L/2}^{L/2} \sum_{-L/2}^{L/2} C(m,n)S(i+m,i+n) \tag{2.9}$$

where *i* denotes the frame number and *L* represents the width of the kernel, which is centered on 0,0. A visual rendering of a similarity matrix (bottom) (with a given grey scale value proportional to the distance measure) together with its corresponding novelty score (top) give a clear image display of the occurrences of different sections in audio, as shown in figure 2.3.

Given that novelty detection is based on the correlation process, the width of the kernel affects the resolution of the detection outcome. A small kernel, which detects novelty on a short time scale, is capable of identifying detailed changes in the audio structure such as the individual note events. On the other hand, a large kernel, which takes a broader view of the audio structure, compensates its coarse detection with a better identification for longer structural changes, such as music transitions, key modulations, etc. A large kernel can be constructed by forming the Kronecker product of **C** with a matrix of one and applying a window to smoothen the edge effects. Finally, segment boundaries are extracted by detecting peaks where the novelty score exceeds a local or global threshold. A binary tree structure is then constructed to organize the index points of the segment boundaries by the novelty score. Our semantic audio segmentation approach is mainly based on (dis)similarity measurement approach. Further details regarding our proposed semantic audio segmentation method will be presented in the next chapter.

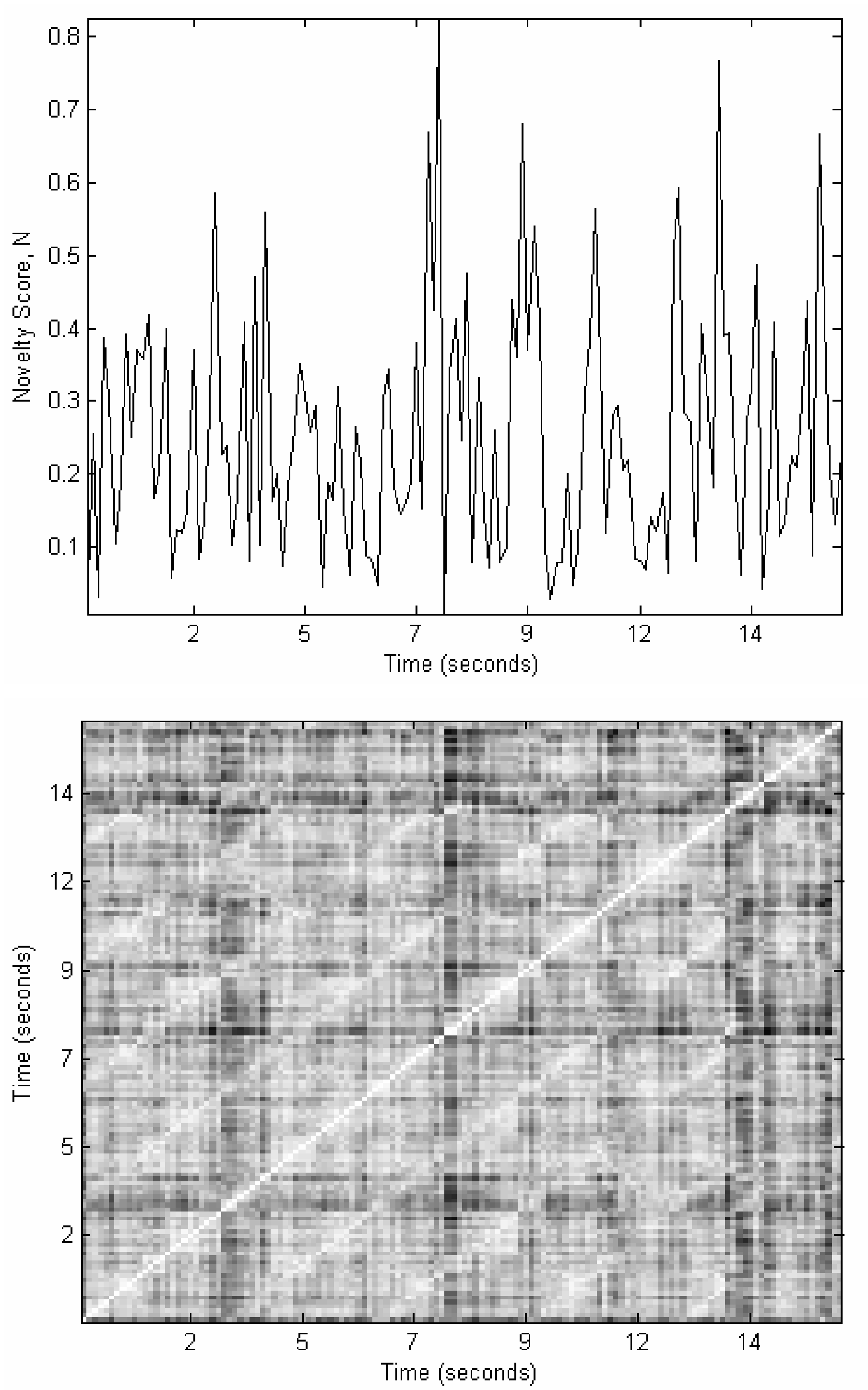


Figure 2.3. (Top) Novelty score and (bottom) similarity matrix computed from an audio excerpt from the soundtrack of "Beauty and the Beast". The MFCC derivatives were used as low-level features

Model-based Segmentation

Hidden Markov Models (HMM) [Rabiner86], a well-known technique in pattern discovery and speech processing, is an example of model-based segmentation used in the research aiming to identify representative musical excerpts. Aucouturier and Sandler [Aucouturier01] train a 4-state ergodic HMM with all possible transitions to discover different regions in music based on the presence of steady statistical texture features. In their experiments, they use the classic Baum-Welsh algorithm to train the HMM. The algorithm optimizes the Gaussian mixture distribution parameters and the transition probabilities for each state of the model for the given training data. Finally, segmentation is deduced by interpreting the results from the Viterbi decoding algorithm for the sequence of feature vectors for the song. One of the two approaches used in Logan and Chu [Logan00] is another example of applying Hidden Markov Models in a post-analysis segmentation task. Since the segmentation and the identification processes are closely related, HMM is capable of integrating the segmentation and identification process into a unified process. In other words, it completes both tasks by using a single algorithm. The application of HMMs for solving identification tasks will be discussed in the following section.

2.3.4. Music Structure Discovery

Structural analysis seeks to derive or discover structural descriptions of music and provide a higher-level interactive way of dealing with audio files. Structural analysis research work such as, semantic audio segmentation [Chai03c], music thumbnailing [Bartsch05, Chai03a, Chai03b, Aucouturier02], music summarization [Cooper03, Xu02], chorus detection [Goto03a, Bartsch01] and repeating patterns identification [Lu04], although carrying different titles, all shares the same goal of facilitating an efficient browsing and searching of music audio files. In fact, they are all built upon the identification of significant audio excerpts that are sufficient to represent a whole piece of music. Hence, identifying the representative musical excerpts from music structure is the key issue here. There are different approaches, including those which are commonly used in pattern recognition and image processing. Here, we organize these approaches into four main groups: Self-similarity Analysis, Dynamic Programming, Clustering, and Hidden Markov Modeling. In the forthcoming subsections we discuss these approaches, including pros and cons of their specific algorithms.

Self-Similarity Analysis

The occurrence of repetitive sections in the structure of music audio has caused researchers to relate music audio structure with fractal geometry phenomena in mathematics. A few methods based on self-similarity have been employed for identifying representative musical excerpts. One of them is the two-dimensional self-similarity matrix [Foote00]. Seeing that self-similarity measurement is capable of expressing local similarity in audio structure, Bartsch and Wakefield [Bartsch01] use a restructured time-lag matrix to store the filtering results that are obtained through applying a uniform moving average filter along the diagonals of the similarity matrix, for the aim of computing similarity between extended regions of the song. Finally, they select the chorus section of music by locating the maximum element of a time-lag matrix based on two defined restrictions: (1) the time position index of the selection must have a lag greater than one-tenth of the song; (2) it appears after less than three-fourths of the way through the song.

Goto's [Goto03a] RefraiD method is another example of using time-lag similarity analysis in identifying representative musical excerpts from audio music. Goto also uses 2-dimensional plot representations having time-lag as their ordinate, in order to represent the similarity and the possibility of containing line segments at the time lag. With an automatic threshold selection method, which is based on a discriminant criterion measure [Otsu79], time-lags with high possibility of containing line-segments are selected. These selected time lags are then used to search on the horizontal time axis on the one-dimensional function for line segments using the same concept of the previous threshold selection method. After that, groups are used to organize those line segments, with each group consisting of the integration of the line segments having common repeated sections. The author then recovers the omitted line segments from previous line segment detection process through searching again the time-lag matrix using the line segment information of each group. Finally groups, which share a same section, are integrated into a singular group. The group integration process works by adding all the line segments belonged to the groups and adjusting the lag values. With the use of the corresponding relation between circular-shifts of the chroma vector and performance modulation, Goto further improves the overall detection performance by tackling the problem in identifying modulated repetition. According to Goto, when an original performance is modulated by *tr* semitones upwards, its modulated chroma vectors satisfy,

$$\vec{v}(t) \doteqdot S^{tr}\vec{v}(t)' \qquad (2.10)$$

where

$\vec{v}(t)$ = chroma vectors of modulated performance,

$\vec{v}(t)'$ = chroma vectors of original performance,

S^{tr} = shift matrix

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & 0 & \ddots & \ddots & \vdots \\ 0 & \ldots & \cdots & \cdots & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix} \qquad (2.11)$$

By using this strategy, Goto computes twelve kinds of extended similarities using the shift matrix and chroma vectors of original performances in order to represent the modulation of twelves semitones upwards. Twelve kinds of extended similarity of each *tr* is defined as:

$$r_{tr}(t,l) = 1 - \frac{\left| \frac{S^{tr}\vec{v}(t)}{\max_c v_c(t)} - \frac{\vec{v}(t-l)}{\max_c v_c(t-l)} \right|}{\sqrt{12}} \qquad (2.12)$$

For each kind of extended similarity, the above-mentioned process of listing and integrating the repeated sections is performed, with the exception that the threshold adjusted for the original performance vectors is used for modulation vectors as well. Goto later unfolds each line segment in each group to obtain unfolded repeated sections and λ_{ij} (its possibility of being chorus sections). Before the possibility λ_{ij} . of each repeated section is used for later calculation for V_i (a total possibility of each group for being a chorus section), it is adjusted based on three heuristic assumptions,

i. The length of the chorus section has an approximate range. If the length is out of range. λ_{ij} is set to 0.

ii. Long repeated sections may correspond to a long-term repetition (e.g. the verseA, verseB and chorus) and it is likely that a chorus section is located near its end. Hence, if there exists a repeated section whose end is close to the end of another long repeated section (longer than 50 sec), its λ_{ij} is doubled.

iii. Because a chorus section tends to have two half-length repeated sub-sections within its section, a section that has those sub-sections is likely to be the chorus section. If there is a repeated section that has those sub-sections in another group, half of the mean of the probability of those two sub-sections is added to its $\lambda_{ij.}$

Finally, the group with the highest possibility value of V_i is selected as the chorus section. Pseudo code in Figure 2.4 depicts the chorus detection procedure by Goto's RefraiD method. Our approach for the identification of representative excerpts from music audio is highly inspired by [Goto03a]. Full detail regarding our approach will be presented in later Chapter IV.

Dynamic Programming

Dynamic programming is another various approaches used to discover musical structure for later music thumbnailing. Chai [Chai03b, Chai03d] uses dynamic programming to perform music pattern matching for finding repetitions in music and later discovering the structure of music. The structural analysis results determine actual alignment at section transitions, which is also similar to music segmentation. After the pre-analysis segmentation process, the author computes the distance, *c*, between each two feature vectors. The computed distances are kept for later usage in a matrix scoring scheme. Two distances have been defined according to different dimensionalities of the used features. The distance between two one-dimensional pitch features v_1 and v_2 is defined as

$$d_p(v_1, v_2) = \frac{|v_1 - v_2|}{normalization\ factor} \tag{2.13}$$

```
% for detecting non-modulated repetition
Compute time-lag matrix, $r(t,l)$
Normalize $r(t,l)$
Compute $R_{all}(t,l)$ with normalized $r(t,l)$
Set threshold, $Th_R$ based on discriminant criterion measure
Execute **function**(detect_repetition)
Recover omitted line segments based on line segment information of each group
Integrate groups that share a same segment into a singular group

% for detecting the modulated repetition
For each semitone
        Compute modulated chroma vector, $\vec{v}(t)$, based on Equation (2.10)
        Compute $r_{tr}(t,l)$ based on Equation (2.12)
        Normalize $r_{tr}(t,l)$
        Compute $R_{all}(t,l)$ with normalized $r(t,l)$
        Execute **function**(detect_repetition)
        Recover omitted line segments based on line segment information of
        each group
        Integrate groups that share a same segment into a singular group
End

For each group
        Define $\lambda_{ij}$ based on three heuristics rules
        Compute total possibility $V_i$,
End

If $m = \underset{i \to group}{\operatorname{argmax}} V_i$
        Select $m$ as chorus sections
End


**function** (detect_repetition)
{
        Let $high\_peak = R_{all}(t,l)$ that is above $Th_R$
        Let $L_{high\_peak}$ = lag information of each $high\_peak$,
        For each $high\_peak$
                Search on the horizontal time axis of $r(\tau,l)$ ($L_{high\_peak} < \tau < t$) at the
        lag $L_{high\_peak}$
                Set threshold, $Th_{Lag}$, based on discriminant criterion measure
                Search smoothed $r(\tau,L_{high\_peak})$ that is above $Th_{Lag}$,
        End

        Group line segments that have almost the same section into a group
}
```

Figure 2.4. Pseudo code depicts the chorus detection procedure by Goto's RefraiD method.

The distance between two multi-dimensional feature vectors (FFT or chroma) $\vec{v_1}$ and $\vec{v_2}$ is defined as

$$d_f(\vec{v_1},\vec{v_2}) = 0.5 - 0.5 \cdot \frac{\left|\vec{v_1} \cdot \vec{v_2}\right|}{\left|\vec{v_1}\right|\left|\vec{v_2}\right|} \tag{2.14}$$

In both cases, the distance ranges between 0 and 1.

The computed feature vector sequence, $V[1,n] = \{v_j \mid j = 1,2,..,n\}$, is segmented into segments of fixed length *l*. Each segment (i.e., $s_i = V[j, j+l-1]$) is then matched with the feature vector sequence starting from this segment (i.e., $V[j,n]$) by using dynamic programming. The author first creates a $(n+1)$-by-$(l+1)$ scoring matrix *M_i*, (as shown in figure 2.5.) and then fills up the matrix based on a scoring scheme shown in Equation 2.15.

$$M(p,q) = \min \begin{cases} M[p-1,q]+e & (i \geq 1) \\ M[p,q-1]+e & (j \geq 1) \\ M[p-1,q-1]+c & (i,j \geq 1) \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

where e is the insertion or deletion cost, and c is the distance between the two corresponding feature vectors mentioned above.

	V_j	$V_{(j+1)}$	$V_{(j+2)}$	$V_{(j+3)}$	⋮	⋮	⋮	⋮	⋮	V_n
V_j	0	0	0	0	…	…	…	…	…	…
$V_{(j+1)}$	e									
…	2e									
…	…									
…	…									
$V_{(j+l-1)}$	le									

Figure 2.5. Dynamic Programming Scoring matrix, M_i.

After the matrix fill step, the author performs a traceback step to determine the actual matching alignments that result in the minimum score. A repetition detection process is then performed by finding the local minima of the traceback results, $d_i[r]$, based on a predefined parameter *h*. The algorithm then merges consecutive segments that have the same repetitive properties into sections and generates pairs of similar section in terms of tuples $< j_1, j_2, shift >$, which indicates the starting and ending location of each segment together with the lag information of its repetition. With the summarized repetition information, the music structure is inferred and labeled based on heuristic rules. Finally, the structure of music is revealed together with its sections boundaries information. With the use of structural analysis results, Chai summarizes the thumbnails of music by choosing the beginning or the end of the most repeated section based on criteria proposed by Logan and Chu [Logan00].

Clustering

Clustering is a grouping technique that has been extensively used in image processing, machine learning and data mining. Clustering organizes a set of objects into groups, such that all objects inside each group are somehow similar to each other. Logan and Chu [Logan00] use a clustering technique to discover the key phrase of music. The authors divide the sequence of features for the whole song into fixed-length contiguous segments, as a starting point. Then an iterative algorithm proceeds according the following step:

1. Compute mean and covariance for each cluster with the assumption that each cluster has a Gaussian distribution.
2. Compute and store the distortion between each pair of cluster using a modified Kullback-Leibler distance measure [Siegler97]. The purposed of using Kullback-Leibler distance measure is to determine how close the two probability distributions are.
3. Select the pair of clusters with the lowest distortion between them.
4. If it is below a predefine threshold, combine these two clusters and go to step1, else continue with step 5.
5. If not, quit
6. Each distinct cluster is assigned a label (such as ‘0’ and ‘1’), with all the frames inside this clusters are given this label.

7. Determine the most frequent label that occurs in the song.

By using this approach, Logan and Chu select the longest section (which consists of the most frequent label that appears in the first half of the song) as the key phrase of the song. Results from the evaluation test show that the clustering approach performed the best when compare to Hidden Markov Modeling and random selection with an average score of 2.4 against 2.1 and 1.9. Nevertheless, the selected key phrase through clustering approach contains an unnatural starting and ending point, which is caused by a limited resolution in the segmentation process.

Other than using K-means, Foote and Cooper [Foote03] propose using Singular Value Decomposition (SVD). SVD is a dimension-reduction technique originally developed for still image segmentation, which can also be used for completing the task of segment clustering. SVD works by performing decomposition on a similarity matrix. In other words, it finds the repeated or substantially similar groups of segments through factoring a segment-indexed similarity matrix. Since it is not clear which clustering methods perform better, it would be worthwhile to make some objective back-to-back comparison between these two techniques.

Hidden Markov Modeling

Hidden Markov Modeling (HMM) [Rabiner89] is another approach used in determining representative excerpts of music signal. HMM has a good capability in grasping the temporal statistical property of stochastic process. A Hidden Markov Model consists of a set of *n* finite number of states interconnected through probabilistic transitions, and is completely defined by the triplet, $\lambda = \{A, B, \pi\}$, where *A* is the state transition probability. *B* is the state observation probability, and π is the initial state distribution. At each time, HMM stays in one specific state. The state at time *t* is directly influenced by the sate at time *t*-1. After each translation from one state to another, an output observation is generated based on an observation probability distribution associated with the current states. State variable are "hidden" and are not directly observable and thus, only output is observable. Figure 2.6 below shows a 4-state ergodic hidden Markov model.

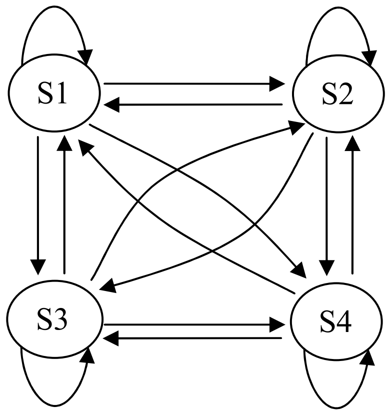


Figure 2.6. A 4-state ergodic hidden Markov Model

With the application of HMM, the segmentation process and the identification process are integrated into a single unified process [Logan00]. Hence, it is not necessary to perform any segmentation prior from using the HMM technique as the system learns the segmentation from the data itself. Unsupervised Baum-Welsh is used to train the HMM given the sequence of feature attributes of the songs. In HMM, each state corresponds to a group of similar frames in the song. With Viterbi decoding, the most likely state sequence for the sequence of frames is determined, where each states is given a label. Finally, continuous segments are constructed by concatenating consecutive frames that have the same given label. Logan and Chu [Logan00] chose the key phrase based on the duration and frequency of occurrence of these segments. In their studies, HMM overcame the problem of unnatural keyphrase beginning that was observed using the clustering approach, even though the HMM did not achieve a satisfactory performance in the evaluation test. Nevertheless, using a fixed number of states in HMMs may not be an optimal solution since in real world music, the number of sections in music may vary significantly from one title to another one.

One underlying issue when using HMM in music structural analysis, lies in finding the appropriate number of states for initialization. An insufficient number of states results in poor representations of the data, whilst an excessive amount of states causes too detailed representations. Besides, using a fixed preset number of states for the HMM model would also limit its potential in structure discovery. Hence, previous knowledge of these parameters will definitely improve the overall performance. Considering this factor, Peeters et al [Peeters02] propose a multi-pass approach combining segmentation and HMM, which does not require the a priori fixing of the number of states, for automatic dynamic generation of audio summaries. Its first-pass performs a post-analysis segmentation through similarity measurements between feature vectors in order to allow the definition of a set of templates (classes) of music. Here, the author intends to make use of the restructured information boundaries from post-analysis segmentation for achieving a better estimation of the number of classes and their potential states for a K-means clustering algorithm [MacQueen67]. With the constituted templates of the music, the second-pass organizes nearly identical (similarity≥ 0.99) templates into groups and uses the reduced number of groups as "initial" states to initialize K-means clustering algorithm. The output from the K-means clustering is then used to initialize an ergodic HMM learning model. Similar to Logan and Chu's [Logan00] approach, the classic Baum-Welch algorithm is used to train the model. The outputs of the training are the state observation probabilities, the state transition probabilities and the initial state distribution. Finally, decoding using Viterbi algorithm with the given HMM and the signal features vectors, they obtain the state sequence corresponding to the piece of music. Through this unsupervised learning process, each time frame is given a state number. The authors suggest that the generation of the audio summary from this state representation can be done in several ways (with a given structure example: AABABCAAB):

- Providing audio example of class transition (A→B, B→A, B→C, C→A)
- Providing a unique audio example of each of the states (A, B, C)
- Reproducing the class successions by providing an audio example for each class apparition (A, B, A, B, C, A, B)
- Providing only an audio example of the most important class in terms of global length or repetitiveness of the class) (A)

Peeters et al. research work has shown that an integrative approach by means of segmentation and an unsupervised learning method (K-means and Hidden Markov Models) can overcome the quick state-jump between states and produce a better and smoother state sequence. Thus, it overall improves the performance of using HMM in music structural discovery. However the authors do not provide any evaluation data to verify this observation.

2.4. Discussion

In this section, we discuss the pros and cons on each approach used in identifying representative musical excerpts of audio music. Self-similarity analysis approach has the advantage of providing a clear and intelligible view of audio structure. Nevertheless, it is not efficient for spotting repetitions with a certain degree of tempo change. A fixed resolution in its feature representation may give a different representation view on the tempo-changed repeated sections compared with its original section. Another problem with this approach is its threshold dependency in reducing noise for line segment detection. Threshold setting may vary from one song to another. Hence, a general setting threshold may not be valid for a wide range of audio.

The dynamic programming approach [Chai03c] has an advantage in offering a better accuracy in boundary detection. However this approach requires the comparison of all possible alignments between two sequences. The number of operations grows quadratically with the total number of frames. Thus, it suffers a lack of scalability.

The clustering approach manages to overcome the problem of the sensitivity to tempo changes that the previously mentioned approaches suffer, as long as boundary truncations are appropriate. However, one has to take notice that clustering, which organizes objects into groups based on their similarity, may produce complex representations of audio structure when a large range of similarity values exists among its feature contents. Hence, this approach is not appropriate for music that has non-homogenous feature contents, such as electronic music.

HMM approach with its transition statistical parameters is capable of handling the problem caused by non-homogenous segments that we have to face with music content analysis. Other advantages of HMMs are their efficiency in handling non-fixed length input and their independency in completing both the segmentation task and identification task without any external support. Nevertheless, this approach has a disadvantage in its expensive computation. In addition, HMM's performance efficiency highly depends on the number of states and on a good initialization. An insufficient number of states causes a poor representation of the data, whilst excessive state numbers cause too detailed representations. As the number of states in HMM can roughly correspond to the amount of different sections in the song, using a fixed number of states in HMMs may cause unsatisfactory outcomes.

So far, much research works focusing on finding significant excerpts to represent a piece of music mainly depend on the repetitiveness of a music segment in the identification task. Apparently, no other assumptions have been proposed. In fact, how does one define the "significant" of an audio excerpt? From the musical point of view, it could be a "chorus" section of the pop music. While from the perceptual or cognitive point of view, it could be the most outstanding or attention grabbing or "strange" or "unexpected" excerpts that are usually not repeated but are capable in leaving a strong impression on our mind. Thus, repetitiveness may not be the only factor in criteria of defining the "significant" of an audio excerpt.

Finally, by reviewing the current developments in this area, we observed a few limitations in the aspect of algorithm evaluations in present literature works. First limitation is the lack of generality of the test databases. Databases consisting of a few hundreds of songs with different diversity and complexity will not be able to reflect the real-world music. Hence, by using such a database, it is quite impossible to obtain an objective evaluation on the algorithm efficiency for most of the existing music. Another limitation is the method in weighting the importance of extracted music sections. The significance of the musical excerpts in audio signal highly depends on human perception.

2.5. Summary

In this chapter, we have covered a substantial range of background information in music structural analysis. We have presented various audio feature classes and extraction approach used for music structural analysis, audio segmentation techniques for better truncations of audio signal and related identification approaches for discovering the structure of music.

In the next chapter, we begin our study of segmenting audio semantically in term of the structural changes in the music signal. Chapter III begins with a presentation of the overview of our proposed framework for semantic audio segmentation. The chapter proceeds by presenting the descriptions of our proposed approach in more detail. Finally, we present quantitative evaluation results of the performance of our proposed method based on a set of test database.

CHAPTER III

SEMANTIC AUDIO SEGMENTATION

In this chapter, we begin our study of automatic music structural analysis with the aim to detect significant structural changes in music signals. It is to provide a way to separate the different "sections" of a piece, such as "intro", "verse", "chorus", etc.

The work in this chapter has two aspects. First, we present our semantic audio segmentation proposed method. In our approach, we divide the segment boundaries detection task into a two-phase process with each having different functionalities. Unlike traditional audio segmentation approaches, we employ image processing techniques to enhance the significant segment boundaries in audio signals. In order to obtain appropriate structural boundaries, we propose a combination of low-level descriptors to be extracted from music audio signal. Section 3.1 comprises all the descriptions in this aspect. First, we start by giving an overview of the proposed framework. Later, we extend the description of each procedure in detail at each subsection according to the processing sequence.

The second aspect, presented in Section 3.2, is a set of experiments to evaluate the efficiency of our system by using various combinations of low-level descriptors and descriptive statistics measures. We use some basic measures in evaluating search strategies to achieve an objective evaluation for each performed experiment. By using our test dataset, which consists of 54 songs from first four CD's of The Beatles' (1962 - 1965), the experiment results show that our approach has achieved 71% of accuracy and 79% of reliability in identifying structural boundaries in music audio signals.

3.1. Approach

Audio segmentation facilitates partitioning audio streams into short regions. It seems an indispensable process in certain content-based applications, such as audio notation, audio summarization, audio content analysis, etc. Seeing this reason, research in this area has receiving an increasing attention in recent years. A number of different approaches have been proposed [Aucouturier01, Foote00, Tzanetakis99, Ong04].

In this chapter, we proposed a novel approach for the detection of structural changes in audio signal by dividing segment detection process in two phases (see diagram in figure 3.1). Each phase is given a different goal: Phase 1 focuses in detecting boundaries, which may contain structural changes from the audio signal; Phase 2 focuses in refining detected boundaries obtained from phase 1 by aggregating contiguous segments while keeping those which really mark structural changes in music audio. Our proposed method consists of 9 steps as follows:

Phase 1 – Rough Segmentation

(1) Segment input signal into overlapped frames of fixed length and compute audio descriptors for each frame (see section 3.1.1);

(2) Compute between-frames cosine distance to obtain several similarity matrices [Foote00] for each one of the used features (see section 3.1.2);

(3) Apply morphological filter operations (see section 3.1.2) to similarity matrices for enhancing the intelligibility of the visualization;

(4) Compute novelty measures by applying kernel correlation [Foote00] along the diagonal of the post-processed similarity matrices (see section 3.1.2);

(5) Detect segments by finding the first 40 highest local maxima from novelty measure plot (see section 3.1.2);

(6) Combine the detected peaks to yield boundary candidates of segment changes of music audio (see section 3.1.2);

Phase 2 – Segment Boundaries Refinement

(7) Assign frames according to detected segments obtained from phase 1 and compute the average for all the used features (see table 3.1) in each segment;

(8) Compute between-segments distances using the mean value of each features in each segment (see table 3.1);

(9) Select significant segments based on distance metrics (see table 3.1).

The following sections explain each step in detail.

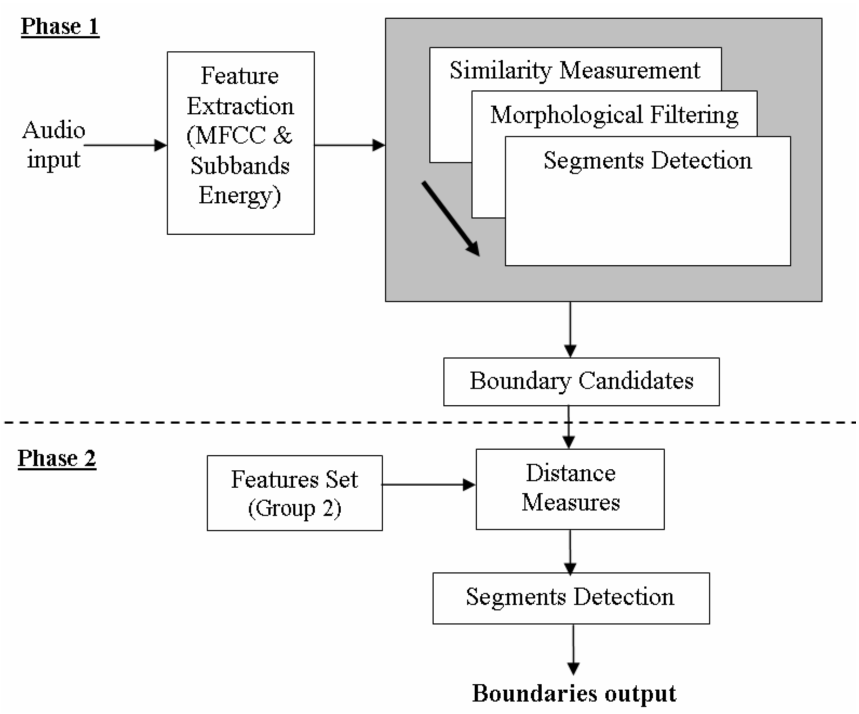


Figure 3.1. Overview framework of our approach.

3.1.1. Feature Extraction

We begin our segmentation task by extracting feature representation of the audio content. As mentioned earlier, detecting significant structural changes in music signal is a key issue of our research objective in this chapter. Thus proper selection of feature attributes is crucial to obtain appropriate musical content descriptors that grant a proper boundaries detection process. Nevertheless, effective description of musical content not only depends on the best feature attributes, but sometimes also on the use of different features in a combined manner. Therefore, the application of musical knowledge into the selection process would further improve the quality of musical content description.

As regards to obtain the short-term descriptions of the audio sound signal, we partition the input signal into overlapped frames (4096-samples window length) with the hop size of 512 samples. We then follow by extracting feature descriptions of each of these frames with a Hamming window.

To estimate the content descriptions of music audio signal, we consider different timbre and dynamics related features: MFCC, sub-bands energy, spectral centroid, spectral rolloff, spectral flux, zero crossings, spectral flatness, low bass energy, high-medium energy and RMS energy. The following gives a brief description for each of the used content descriptor. Please refer to section 2.3.1 for details explanation of these descriptors.

MFCC, also called Mel-Frequency Cepstral Coefficients: A compact representation of an audio spectrum that takes into account the non-linear human perceptual of pitch, as described by Mel-scale.

Sub-bands energy: A measure of power spectrum in each sub-band. We divide the power spectrum into 9 non-overlapping frequency bands as describe in [Maddage04].

Spectral Centroid: A representation of the balancing point of the spectral power distribution within a frame.

Spectral Rolloff: A measure of frequency, which is below 95 percentile of the power spectral distribution. It is a measure of "skewness" of the spectral.

Spectral Flux: The 2-norm of the frame-to-frame spectral magnitude difference vector. It measures spectral difference, thus it characterizes the shape changes of the spectrum.

Zero Crossings: A time-domain measure that gives an approximation of the signal's noisiness

Spectral Flatness: A measure of the flatness properties of spectrum within a number frequency bands. High deviation from a flat shape might indicate the presence of tonal component.

High-medium energy: A ratio of spectrum content within the frequency range of 1.6 kHz and 4 kHz to the total content. This frequency range comprises all the important harmonics, especially for sung music.

Low Bass energy: A ration of low frequency component (up to 90 Hz) to the total spectrum energy content. This frequency range includes the greatest perceptible changes in "bass responses".

RMS energy: A measure of loudness of the sound in frame.

In our approach, we use two different groups of descriptors, each one for one of the different phases.

Phase 1	**Phase 2**
MFCC Sub-bands Energy	Zero Crossings rate, Spectral Centroid, Spectral Flatness, Spectral Rolloff, Spectral Flux, RMS, Low Bass Energy, High-medium Energy

Table 3.1. The list of audio descriptors for Phase 1 and Phase 2.

3.1.2. Phase 1 – Rough Segmentation

After computing feature vectors for each frame, we group every 10 frames (116ms) and calculate the mean value for every feature. In this phase of segment detection process, we only work with MFCC and subband energies. We treat those features separately in order to combine both results in the final stage of detection process in phase 1. In order to find the structural changes in the audio data, we measure the distance between each feature vectors, $V_n = \{v_{n,1}, v_{n,2}, ..., v_{n,m}\}$, and their neighbouring vectors, $V_{n+i} = \{v_{n+i,1}, v_{n+i,2}, ..., v_{n+i,m}\}$, using cosine angle distance [Foote00] given by the expression:

$$SD_{\text{cosine}}(V_n, V_{n+i}) = \frac{V_n \bullet V_{n+i}}{\|V_n\| \|V_{n+i}\|} \tag{3.1}$$

$$= \frac{\sum_{j=1}^{m} (v_{n,j} * v_{n+i,j})}{\sqrt{\sum_{j=1}^{m} (v_{n,j})^2 * \sum_{j=1}^{m} (v_{n+i,j})^2}} \tag{3.2}$$

where *m* denotes *m*-dimensional of the feature vectors.

Figure 3.2 illustrate the two-dimensional cosine similarity plot computed using MFCC features. As shown in figure 3.2, some structural changes can be perceived in the similarity plot. To enhance such information, we need to further improve the intelligibility of the vague visualization given by the similarity plot. For this purpose, we apply morphological filters [Burgeth04], a widely used filtering techniques applied to image processing, on the computed distance matrix representations. The idea behind this operation is to increase the intelligibility of the structural changes and facilitate the enhancement of the segment boundaries. The reason for selecting morphological filter lies on its advantages in preserving edge information and computationally efficiency over other techniques. Since morphological filtering techniques are relatively unknown in music analysis, we dedicate a few paragraphs below to give a brief introduction about its operations' functionalities and implementations procedure.

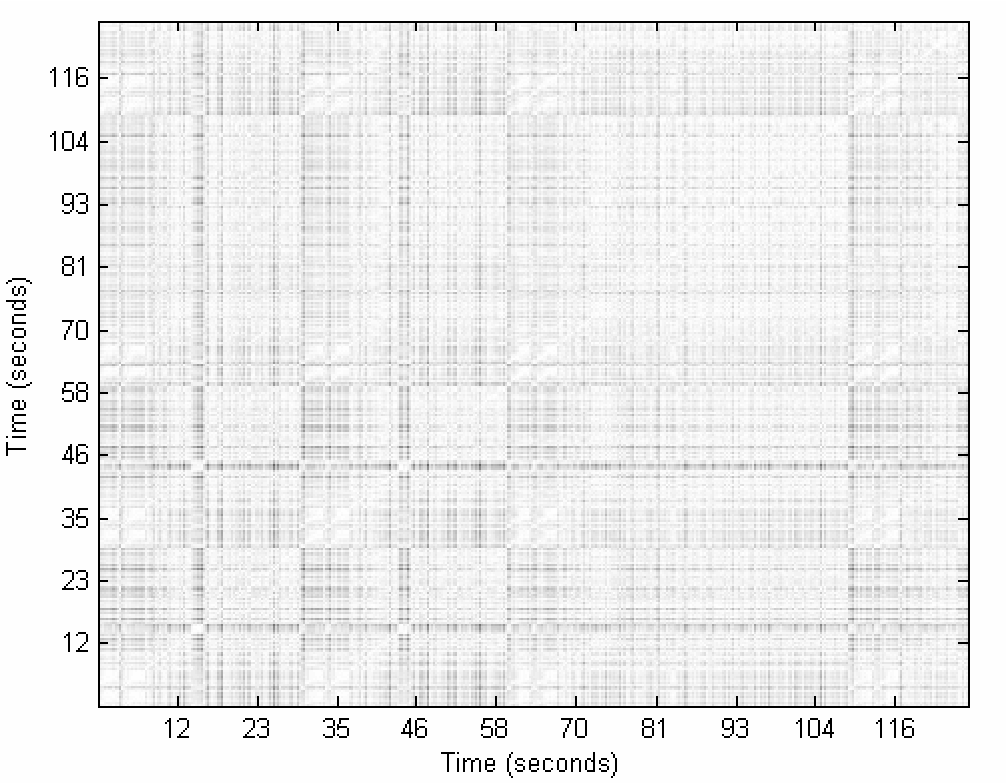


Figure 3.2. Two-dimensional cosine similarity plot computed from the song entitled “When I Get Home” using MFCC features.

Morphology filtering is an analysis process of signal in terms of shape. Basically, it uses set theory as the foundation for many of its operations [Young02]. Its simplest operations are dilation and erosion. In general, dilation causes objects to dilate or grow in size while erosion causes objects to shrink. The amount of changes (grows or shrinks) depends on the choice of the structuring element. The following paragraph explains how dilation and erosion work in detail. Dilation, also known as ‘Minkowski Addition’, works by moving structuring element over input signal and the intersection of structuring element reflected and translated with input signal is found. In another words, the output is set to one unless the input is the inverse of the structuring element. For instance, ‘000’ would cause the output to be zero and placed at the origin of the structuring element, B, for the given example in figure 3.3.a. Similar to dilation, erosion, also known as ‘Minkowski Subtraction’, works by moving structuring element over input signal. The erosion of input signal, A, and structure element, B, is the set of points x such that B translated by x is contained in A. In contrast with dilation operation, the output is set to zero unless the input is identical with the structuring element. Figure 3.3.b shows how erosion opens up the zeros and removes runs of ones that are shorter than the structuring element in one-dimensional binary signal. [Young02].

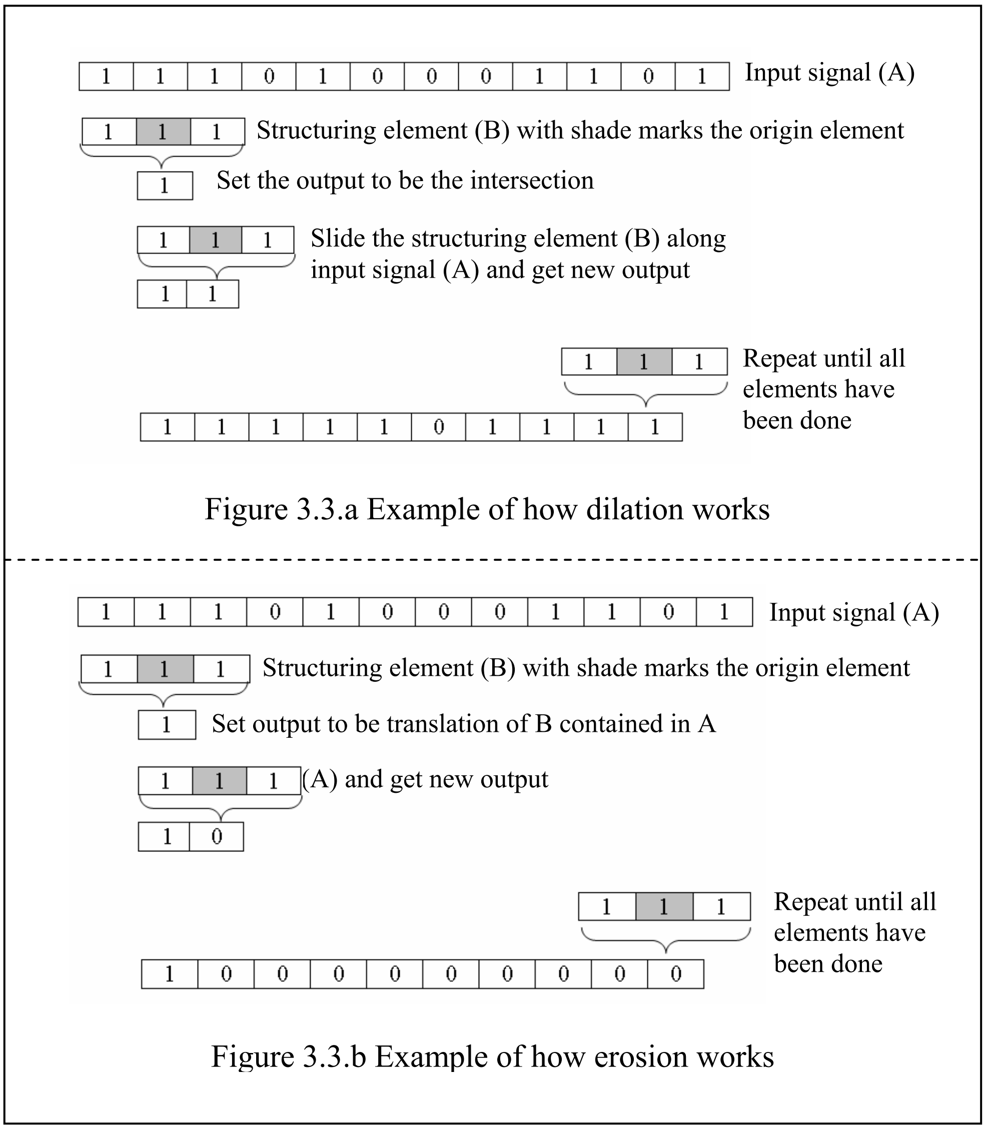

1 1 1 0 1 0 0 0 1 1 0 1
Input signal (A)
1 1 1
Structuring element (B) with shade marks the origin element
1
Set the output to be the intersection
1 1 1
Slide the structuring element (B) along input signal (A) and get new output
1 1
1 1 1
Repeat until all elements have been done
1 1 1 1 1 0 1 1 1 1

Figure 3.3.a Example of how dilation works

1 1 1 0 1 0 0 0 1 1 0 1
Input signal (A)
1 1 1
Structuring element (B) with shade marks the origin element
1
Set output to be translation of B contained in A
1 1 1
(A) and get new output
1 0
1 1 1
Repeat until all elements have been done
1 0 0 0 0 0 0 0 0 0

Figure 3.3.b Example of how erosion works

Figure 3.3. Examples of how dilation and erosion work with the shaded structuring elements show the origin element.

So far, the above mentioned dilation and erosion operations are associated with one-dimensional binary signal. For non-binary signal, dilation (erosion) operation works the same as taking the maximum (minimum) value of the signal, which lies within the 1's of the structuring element. Thus, dilation and erosion operations for non-binary signal can be redefined as

$$Dilation = \max_{i \in B}(A_x) \qquad \text{where,} \quad |i| \leq \frac{n-1}{2} \;\cap\; i \in \mathbb{Z} \qquad (3.3)$$

$$Erosion = \min_{i \in B}(A_x) \qquad \text{where,} \quad |i| \leq \frac{n-1}{2} \;\cap\; i \in \mathbb{Z} \qquad (3.4)$$

Figure 3.4 illustrates the properties of the input signal, A_x, with its structuring element, B_i, as defined in expression 1 and 2.

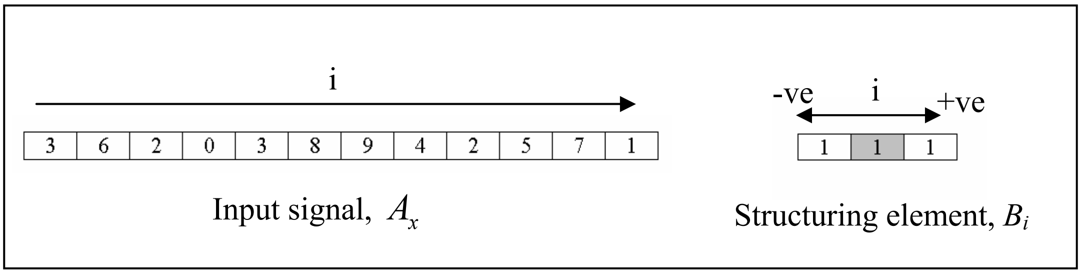


Figure 3.4. The properties of one-dimensional signal, A_x, with its structuring element, B_i, in defined in expressions 1 and 2.

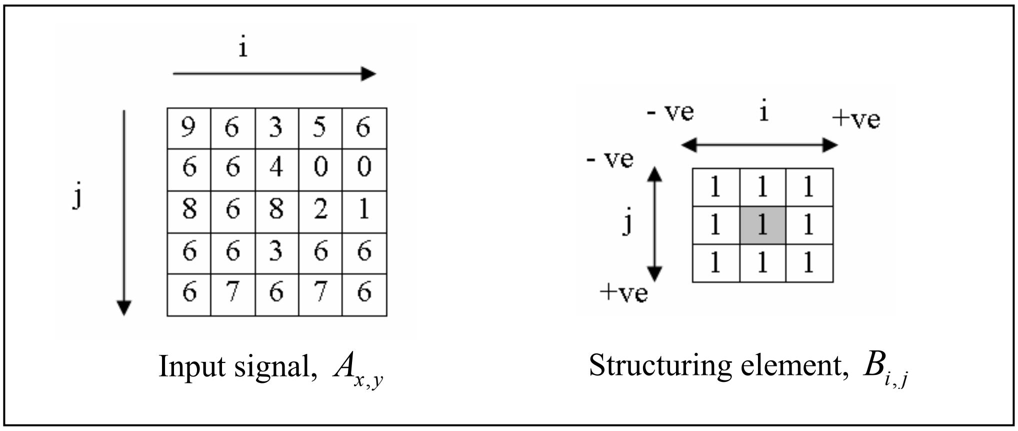
For two-dimensional input signal, erosion and dilation operations still work in exactly the same way as in one-dimensional but with a two-dimensional structuring element instead. Hence, dilation and erosion operations for two-dimensional signal can be expressed as:

$$Dilation = \max_{(i,j) \in B}(A_{x+i,y+j}), \quad \text{where} \; |i|,|j| \leq \frac{n-1}{2} \;\cap\; i,j \in \mathbb{Z} \qquad (3.5)$$

$$Erosion = \min_{(i,j) \in B}(A_{x+i,y+j}), \quad \text{where} \; |i|,|j| \leq \frac{n-1}{2} \;\cap\; i,j \in \mathbb{Z} \qquad (3.6)$$

Figure 3.5 shows the two-dimensional properties of input signal, $A_{x,y}$, with its *n*-by-*n* structuring element, $B_{i,j}$, as defined in expression 3 and 4.

'Opening' and 'Closing' operations are two morphological filter operations, which contain the properties of both dilation and erosion operations. The 'Closing' operation works by dilating the signal and followed by eroding the results. In contrast, the 'Opening' operation works by eroding the signal followed by dilating the results. Figure 3.6 demonstrate both 'Closing' and 'Opening' operations of morphological filter on one-dimensional binary signal. From the figure, we can clearly see the distinct properties of these two operations. The ‘Closing’ operation (as shown in figure 3.6.a) closes the gaps that lie within the length of structuring element, whereas the ‘Opening’ operation (as shown in figure 3.6.b) opens the gaps and removes runs of ones that are shorter than structuring element of the signal. Otherwise, the signal is left unchanged.



9	6	3	5	6
6	6	4	0	0
8	6	8	2	1
6	6	3	6	6
6	7	6	7	6

Input signal, $A_{x,y}$

1	1	1
1	1	1
1	1	1

Structuring element, $B_{i,j}$

Figure 3.5. The two-dimensional properties of input signal, $A_{x,y}$, with its *n*-by-*n* structuring element, $B_{i,j}$, as defined in expression 3 and 4.

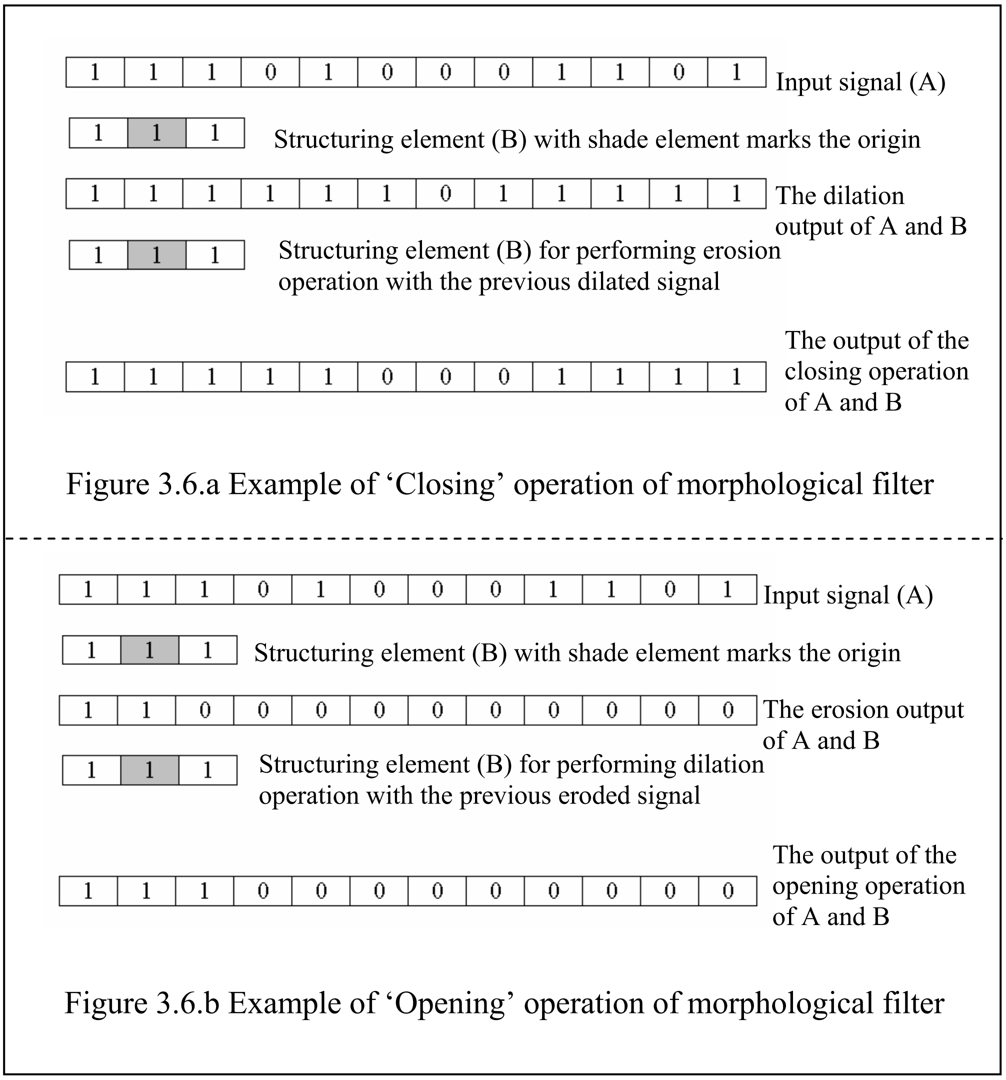
In our work, we utilize ‘Open-Close’ and ‘Close-Open’ operations of morphological filter. There two operations are the combination products of ‘Opening’ and ‘Closing’ operations in order to merge their properties into one filter operation. ‘Open-Close’ operation is implemented by first opening the signal and then closing the opened signal. In contrast with ‘Open-Close’ operation, ‘Close-Open’ is implemented by first closing the signal and then opening the closed signal. Both ‘Open-Close’ and ‘Close-Open’ can be expressed as:

$$\text{Open-Close}(A,B) = close(open(A,B),B) \tag{3.7}$$

$$\text{Close-Open}(A,B) = open(close(A,B),B) \tag{3.8}$$

where *A* denotes the input signal and *B* is the structuring element.

Figure 3.7 shows how these filter operation work on one-dimensional binary signal. Comparing the operations outputs showed in figure 3.6 with those in figure 3.7, we can see that Open-Close' ('Close-Open') operations produce a similar output as 'Opening' ('Closing') operations when dealing with one-dimensional binary signal. However it is not such a case when applying to two-dimensional non-binary signal. 'Opening' operation will remove high intensity points whilst keeping the rest of the signal intact. 'Closing' operation will discard low valued points whilst keeping the rest of the signal intact. While 'Open-Close' and 'Close-Open' operations will remove both high and low valued points while keeping the rest of the signal intact.



1 1 1 0 1 0 0 0 1 1 0 1 Input signal (A)

1 1 1 Structuring element (B) with shade element marks the origin

1 1 1 1 1 1 0 1 1 1 1 1 The dilation output of A and B

1 1 1 Structuring element (B) for performing erosion operation with the previous dilated signal

1 1 1 1 1 0 0 0 1 1 1 1 The output of the closing operation of A and B

Figure 3.6.a Example of 'Closing' operation of morphological filter

1 1 1 0 1 0 0 0 1 1 0 1 Input signal (A)

1 1 1 Structuring element (B) with shade element marks the origin

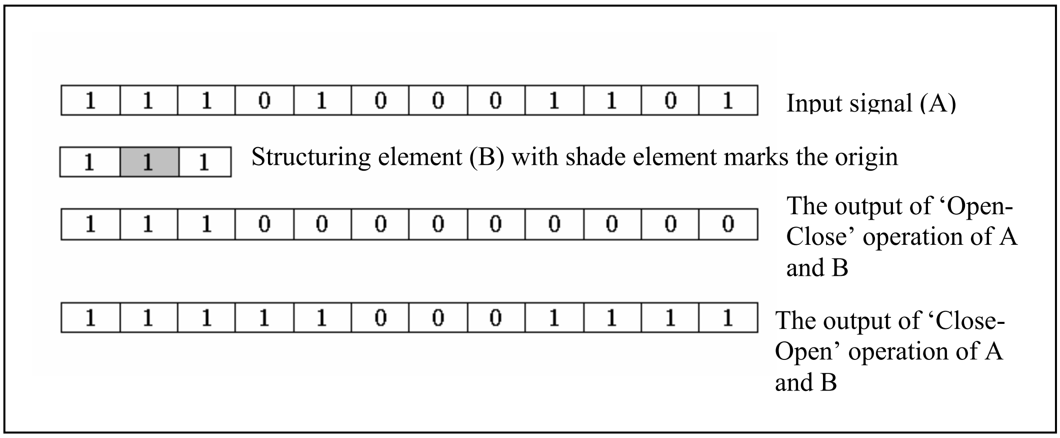
1 1 0 0 0 0 0 0 0 0 0 0 The erosion output of A and B

1 1 1 Structuring element (B) for performing dilation operation with the previous eroded signal

1 1 1 0 0 0 0 0 0 0 0 0 The output of the opening operation of A and B

Figure 3.6.b Example of 'Opening' operation of morphological filter

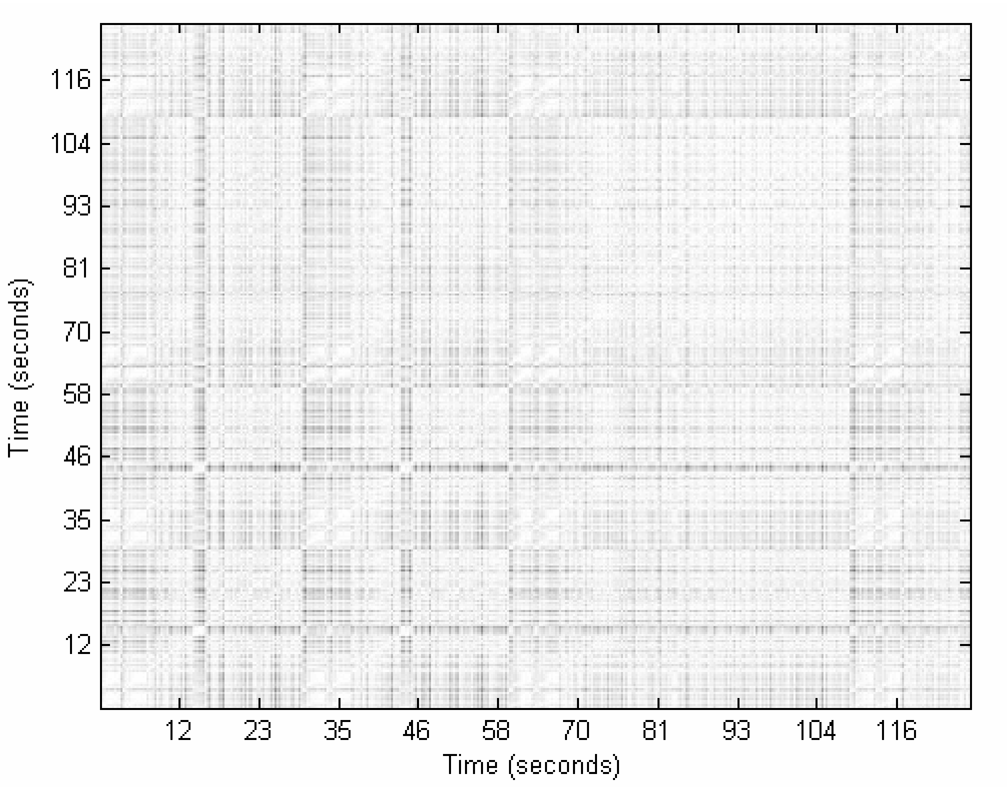
Figure 3.6. The opening operation of morphological filter on one-dimensional binary signal.



1	1	1	0	1	0	0	0	1	1	0	1	Input signal (A)
1	1	1										Structuring element (B) with shade element marks the origin
1	1	1	0	0	0	0	0	0	0	0	0	The output of 'Open-Close' operation of A and B
1	1	1	1	1	0	0	0	1	1	1	1	The output of 'Close-Open' operation of A and B

Figure 3.7. The 'Open-Close' and 'Close-Open' operations of morphological filter on one-dimensional binary signal.

Since our computed distance matrix consists of two-dimensional non-binary signals, the applications of 'Open-Close' and 'Close-Open' operations would disregard high and low valued points in our distance matrix and produce (dis)similarity representations with an enhanced intelligibility. Figure 3.8 and Figure 3.9 illustrate the post-processed distance matrixes obtained from applying 'Close-Open' and 'Open-Close' operations independently on the distance matrix as shown in Figure 3.2. Compared to figure 3.2, the appearances of structural patterns in the distance representation plots have been amplified after morphological filtering processes. From the figures, we can see that although both two operations have the same characteristics in removing intensity points from signal, they do not produce the same filter results. It is due to different sequential of erosions and dilations in implementing both operations. In addition to the outputs of both morphological operations, we also utilize additional distance matrixes yielded from the multiplication and subtraction between 'Close-Open' and 'Open-Close' operation outputs to facilitate the identification of relevant structural changes of music. Figure 3.10 illustrate the distance matrix representation obtained from the multiplication between 'Open-Close' and 'Close-Open' filter results.



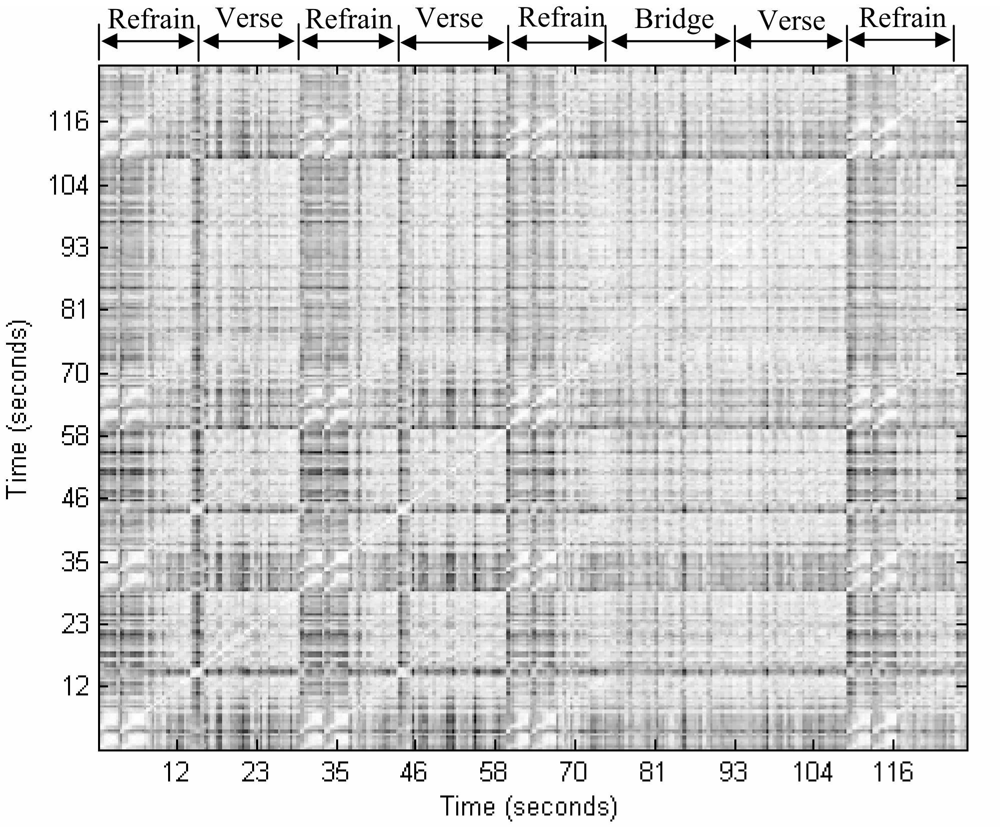


Figure 3.8. Similarity representation before morphological operation (top) versus similarity representation after 'Close-Open' operation.

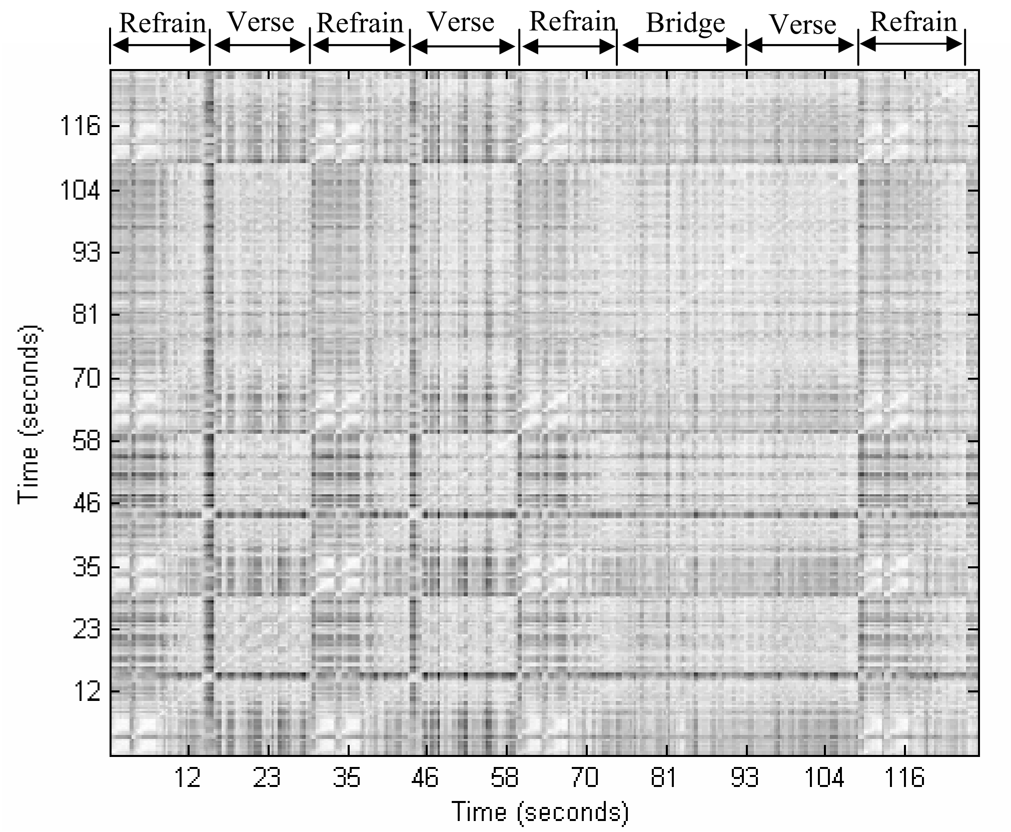


Figure 3.9. Similarity representation after ‘Open-Close’ operation

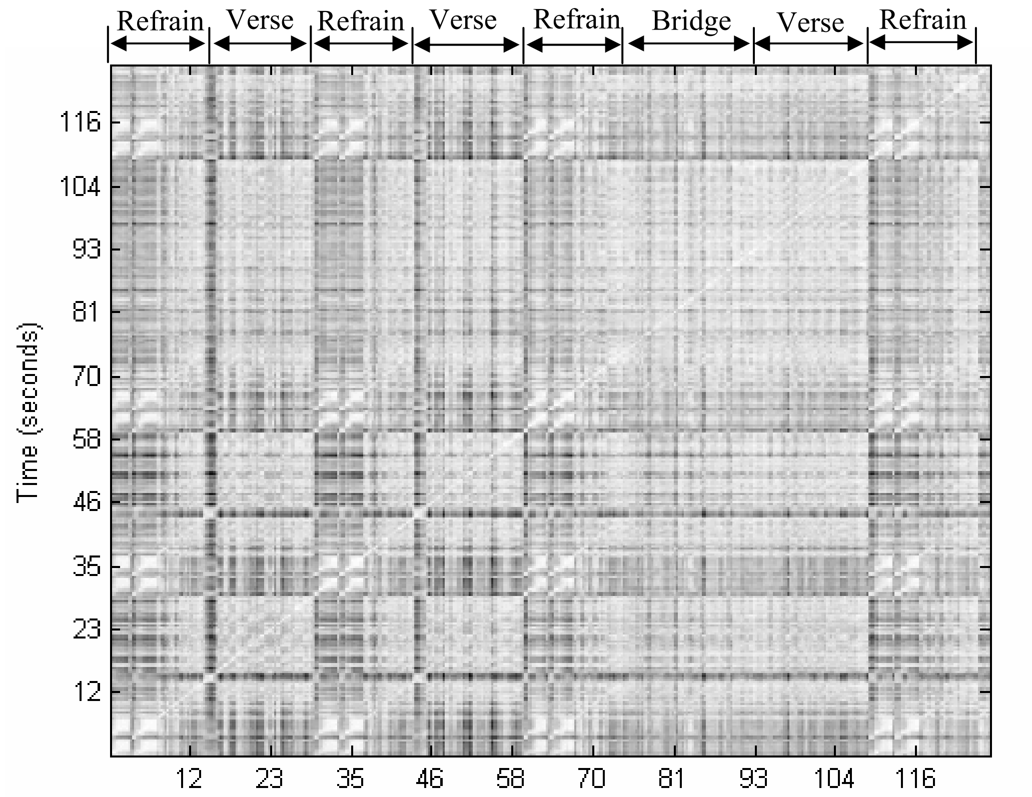


Figure 3.10. Distance matrix representation obtained from the multiplication between ‘Open-Close’ and ‘Close-Open’ filter results.

With the post-processed similarity matrixes obtained from morphological filtering operations, we then apply a kernel correlation [Foote00], with the width of 10, along the diagonal of each post-processed similarity matrix to measure the audio novelty. This is to observe any significant changes of the related audio contents for every 1 second approximately. In order to accumulate all information from the post-processed similarity matrixes, we aggregate all the computed novelty measures and normalize it with its maximum value. This produces an overall novelty measure with its values lie within the range of 0 to 1. Based on the overall novelty measure, the first 40 highest local maxima are selected based on a constraint that each selected local maximum must be at least *m* seconds apart from its neighouring selected local maximum. Three preset *m* parameters have been considered: 2.3 sec, 2.9 sec, and 3.5 sec. Finally, we accumulate all the detected peaks detected based on these three *m* parameters and select the first 40 highest local maxima amongst all local maxima as the segment boundaries candidates from the employed features.

As mentioned in the earlier section, in this phase of the segment detection process, we are working with MFCC and subband energies. Hence the whole process of similarity measurement, morphological filtering and segment candidates' selection processes is repeated for subband energies. Finally, we combine all segment boundaries candidates detected from both features (MFCC and subband energy) and select the first 40 highest amongst them to be considered as boundaries candidates of segment changes of music audio. The selection is based on the criterion that each segment candidate must be at least 2.3 seconds (the lowest considered value for *m* parameters) apart from each other. Figure 3.11 illustrates the detected boundaries candidates yielded by segment detection process in phase 1.

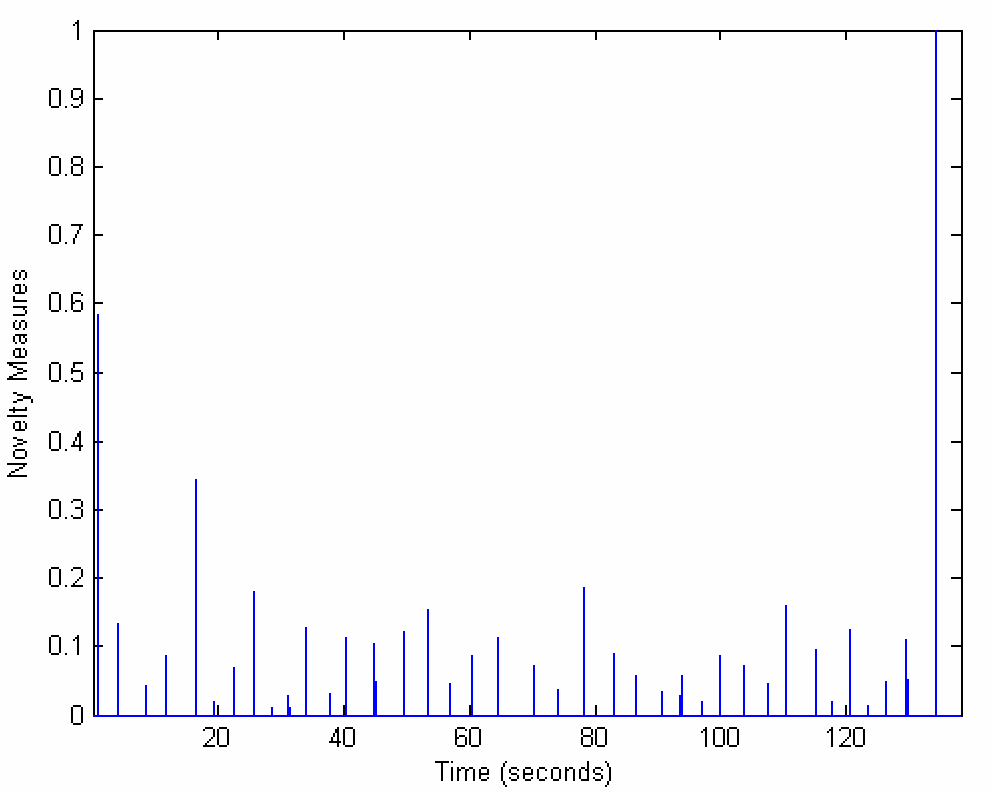


Figure 3.11. Detected boundaries candidates yielded by segment detection process in phase 1.

3.1.3. Phase 2 – Segment Boundaries Refinement

In the second phase of the detection process, we are only using time position from the candidates extracted in the previous phase. Here, we consider the values for all the attributes used in phase 2 (see table 3.1) that are within the detected segment boundaries and compute the average of each of them. Hence, each detected segment now comprises only a set of feature vectors representing the mean value of the attribute in that segment. It has to be mentioned that in our used attributes, there exists a different range of feature values. Apparently, attributes whose values are larger than the other would have more influence in determining the similarity of any two sequences. Hence, in order to avoid such effect and to have an equal importance weight among the used attributes, we normalize all attributes so that its feature values are within the range of 0 and 1. We then compute (dis)similarity between each segment and its neighbouring segments by measuring the Euclidean distance between their feature vectors. Euclidean distance between vectors, $V_n = \{v_{n,1}, v_{n,2}, ..., v_{n,m}\}$ and $V_{n+i} = \{v_{n+i,1}, v_{n+i,2}, ..., v_{n+i,m}\}$ is given by the expression:

$$\left|V_n - V_{n+i}\right| = \sqrt{\sum\nolimits_{j=1}^{m} (v_{n,j} - v_{n+i,j})^2} \tag{3.9}$$

where *m* denotes *m*-dimensional of the feature vectors. Theoretically, Euclidean distance and cosine angle distance used in section 3.1.2 give a same distance measure when two compared feature vectors have same variance value [Gang02]. In fact cosine angle distance is very sensitive to the variance of compared feature vectors. Thus, it is very useful in finding very similar items. Since our feature vectors in this phase are obtained based on the detected boundaries information from phase 1, we hypothesize that Euclidean distance will be more suitable to compute the distance measures of these feature vectors. Similar to the previous steps in computing novelty measures from the similarity representations, we apply a kernel correlation, along the diagonal of (dis)similarity representation of segments to yield the novelty measures, *N*, between each segment and its next sequential segment. Figure 3.12 and Figure 3.13 illustrate the (dis)similarity representations and novelty measures computed from the (dis)similarity representations between segments. Finally we select significant segment boundaries from the computed novelty measures, $N = \{n_s \mid s = 1, 2, ..., l\}$ (where l is the number of segment boundaries candidates) based on the following steps:

1. Select all the peaks that lie above a predefined threshold, *Pt*, based on their computed novelty measures, N_s, and organize them into a group, which is represented as $P = \{p_i \mid i = 1, 2, ..., M\}$ (*M* is the number of selected peaks). Whereas those peaks that lie below the predefined threshold, *Pt*,, are organized into another group denoted by $E = \{e_j \mid j = 1, 2, ..., N\}$ (*N* is the number of unselected peaks).
2. Organize all peaks in *E* in ascending order according to their distance measures.
3. Select the highest peak in *E* for further evaluation.
4. Based on temporal information, if the evaluated peak is located at least 4 sec apart from any peaks in *P*, insert it in group *P* and reorganize all peaks in group *P* in ascending order based on the segment index number; otherwise delete it from *E*. This is based on the assumption that each section in music (ex. verse, chorus,

etc.) should at least hold 4 sec (1 bar for songs with quadruple meter with 60 bpm tempo) in length before moving to the next section.

5. Go to step 3.

The whole iterative peak selection process ends when there is no more peak in *E*. Finally, segment boundaries in *P* are considered as significant segment boundaries that mark structural changes in music audio signals.

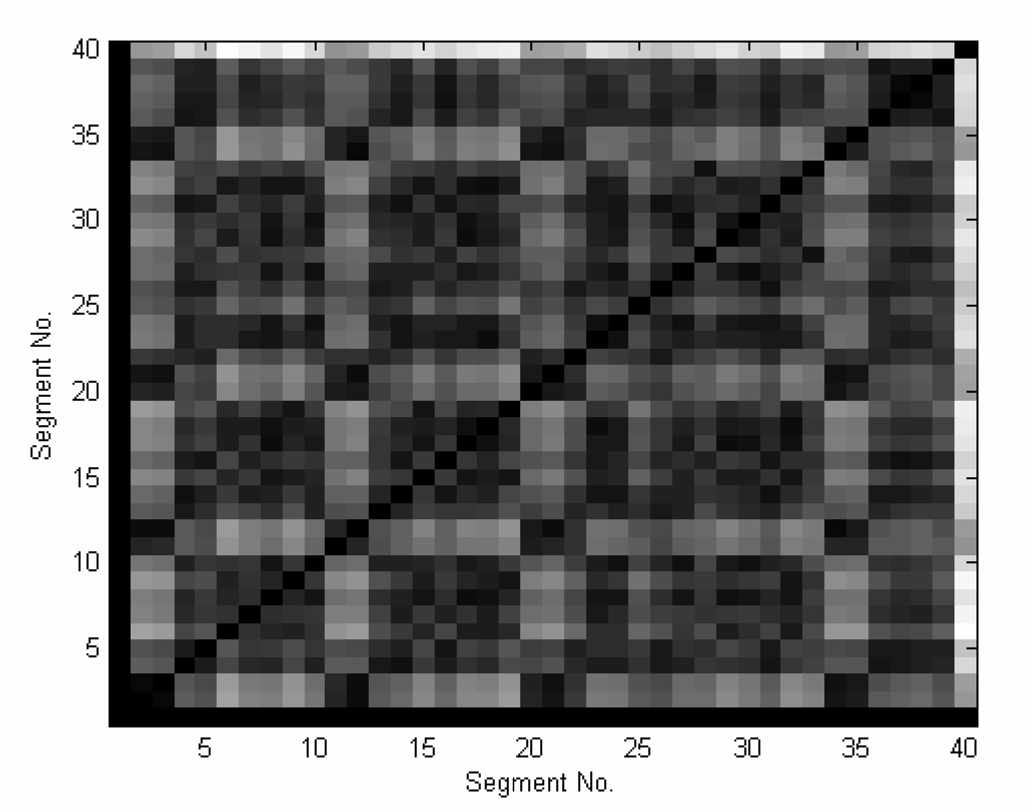


Figure 3.12. The (dis)similarity representations between segments detected in phase 1.

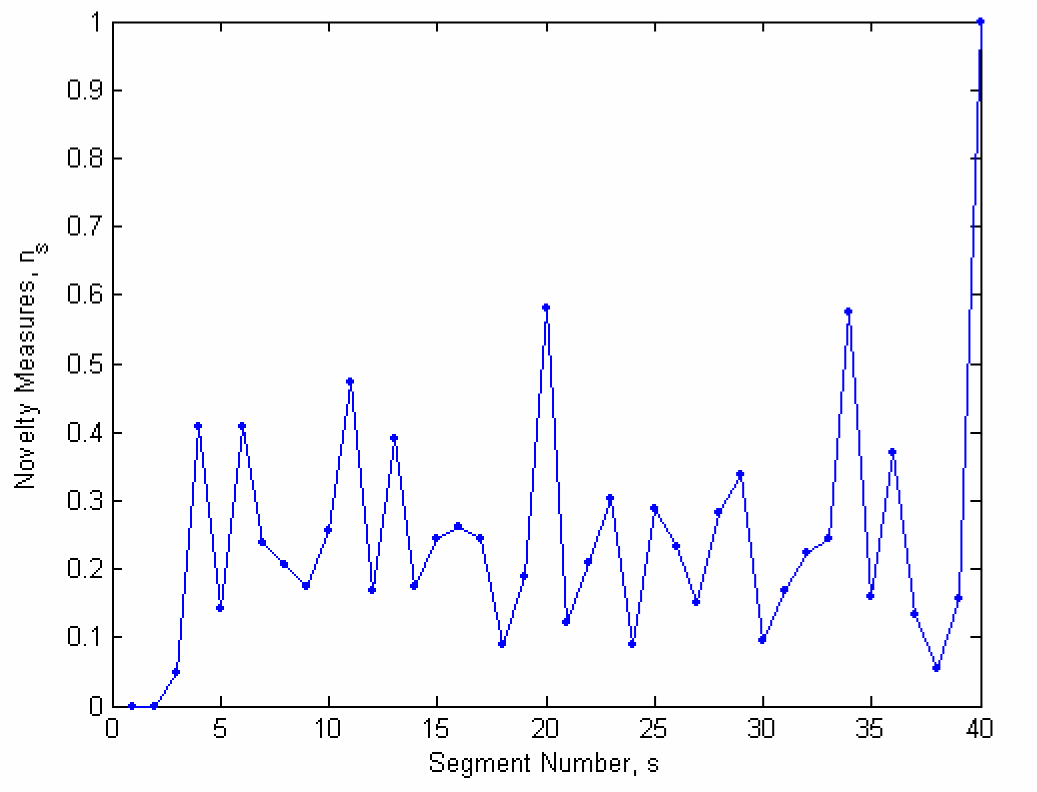


Figure 3.13. The novelty measures computed from the (dis)similarity representations between segments.

3.2. Evaluation

In this section we present an evaluation of our proposed semantic audio segmentation method in detecting structural changes of music audio signal. We first begin by discussing in detail our dataset. It is then followed by presenting the measures used for evaluation the performance of our proposed method.

3.2.1. Datasets

In our experiment, we create an audio database, which consists of 54 songs from first four CD's of The Beatles (1962 – 1965) as a test set. Each song is sampled at 44.1 kHz, 16-bit mono. In the objective evaluation, we have generated a ground truth truth by manually labelling all the sections (i.e. intro, verse, chorus, bridge, verse, outro, etc.) of all the songs in the test set, according to the information provided by Allan W. Pollack's "Notes On" Series website on song analyses of Beatles' twelve recording project[7]. A music composer supervised the labelling process and results.

3.2.2. Procedure

To quantitatively evaluate the detected segments from the proposed algorithm, the detected segment boundaries are compared with the ground truth in term of precision and recall. The precision and recall are defined as follow:

$$\text{Precision} = \frac{D \cap G}{D} \tag{3.10}$$

$$\text{Recall} = \frac{D \cap G}{G} \tag{3.11}$$

where D denotes detected segments, G denotes relevant ground truth segments and $D \cap G$ signifies detected segments that place within the region of relevant ground truth segments' with its tolerance deviation. In evaluating the identified segments, we

[7] The Twelve Recording Projects of the Beatles web page: http://www.icce.rug.nl/~soundscapes/DATABASES/AWP/awp-beatles_ projects. html

allow a tolerance deviation of ± 3 seconds (approximately 1 bar for a song of quadruple meter with 80 bpm in tempo) from the manually labeled boundaries.

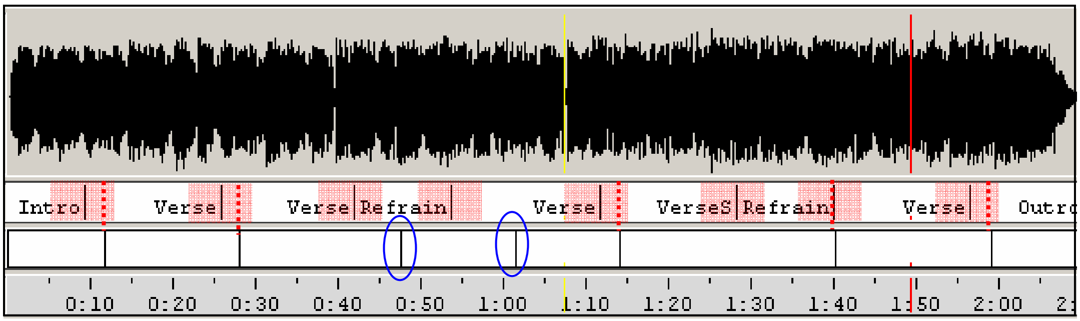


Figure 3.14. An example of measuring segmentation performance with a tolerance deviation presented as shaded area (top transcription: ground truth segments; bottom transcription: detected segment). Circled segments mark the outliers of the correctly detected segments.

Figure 3.14 shows an example of measuring segmentation performance with a tolerance deviation. In the example shown in the figure 3.14, segments with marked circle do not fall within the region of ground truth segment boundaries with its tolerance deviation (shaded area). Hence these two mark circled segments will not be considered as $D \cap G$. The top transcription, which represents the ground truth results, comprises 8 segment boundaries whereas the bottom transcription, which denotes detected results, comprises 7 segment boundaries. Thus, the precision and recall in this example are 5/7 (≈ 0.71) and 5/8 (≈ 0.63).

Precision and recall measures are mainly used to evaluate the accuracy and reliability of the proposed algorithm. In addition, we use F-measure to evaluate overall effectiveness of segment detection by combine recall and precision with an equal weight:

$$\text{F-measure} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision+Recall}} \tag{3.12}$$

3.2.3. Results and Discussion

In this section, we summarize the key findings from the quantitative evaluation results of our proposed segmentation method discussed in this chapter.

From our test dataset, our proposed structural changes detection approach has achieved accuracy higher than 71% and a reliability of 79% using the ground truth set. The overall F-measure has reached 75%. In another words, with 10 detected segment boundaries, 7 of them are correctly detected compared to the ground truth data. Whilst about 2 out of 10 manually labelled boundaries are missed by our automatic boundaries detector. The distribution of precision scores has a standard deviation of 0.11 and the range of precision values spans across 0.41-0.94. Bar graphs in Figure 3.3 shows the precision and recall scores obtained for all songs in our test database. From our results, we observe that the best performance in the case of SongID-35 with its recall and precision score of 100% and 94% respectively. Whereas the worst performance is observed in the case of SongID-47, which only reaches the recall rate of 38% and precision rate of 56%. Figure 3.4 illustrates the detected segment boundaries by our proposed algorithm with manually labelled segment boundaries for SongID-35. We have compared our approach with a previous method described in [Chai03c] with our test data set. It was reported that with an allowed tolerance deviation of 3.7 sec (higher than ours), the author reported less than 60% for both recall and precision measures respectively whereas we achieved over 70% for these both measures.

3.3. Summary

In this chapter, we have presented a novel approach for detecting structural changes in music audio using a two-phase procedure and different descriptors for each phase. A combination set of audio descriptors has also been shown useful in detecting music structural changes. Evaluation results have shown the validity and the performance of our proposed approach.

Even though a system with a good semantic segmentation is highly significant for allocating structural changes in music, it may not be practically useful when dealing with search or retrieval of huge amount of music audio files. In the next chapter, we present our approach in identifying and extracting representative excerpts of music audio. The system simultaneously identifies repetition patterns appear in music and provide a visual representation of music structure. The following chapter includes the objective evaluation on the performance of our proposed representative excerpts identification method with the used of a mixture of polyphonic music recordings.

CHAPTER IV

IDENTIFYING REPRESENTATIVE EXCERPTS FROM MUSIC AUDIO SIGNALS

In Chapter 3, we described a method for segmenting music audio signal. However, a system that can provide music structural changes may not be sufficient for practical use when dealing with search or retrieval of huge amount of music audio files. Humans assimilate information at a semantic level with remarkable ease. Same with the aspect of audio music, we do not recall the music that we hear in its entirely but through a small number of distinctive excerpts (e.g. chorus, verse, intro, etc.) that have left an impression on our mind. It is usually the case that we only need to listen to one of those distinctive excerpts in order to recall the title for the musical piece, or, at least, to tell if we have heard this song before. Thus, we hypothesize that identification of representative excerpts from audio signal would be a primary step going towards of generating music structure metadata and contribute to better and efficient retrieval of current massive amount of digital audio data. In addition, it will serve as indispensable processing towards music summarization applications that aim to generate abstracts from music audio similar to those trailer or preview of movies. In this chapter, we present our approach towards computing a representative audio excerpt from music audio.

We have implemented our own system to perform the identification task of finding representative excerpt in music audio signal via automatic structure analysis. Our system takes a polyphonic music audio signal as input and it detects repetitions that appear in the signal by comparing the pitch chroma information of the music according to frame-to-frame information. After some processing of these detected repetitions, which will be described in details in the following sections of this chapter, the system will produce an audio excerpt that serves as the representative excerpt of the music audio signal. In addition, it will also output transcriptions files which comprise the beginning and ending time of the repetition occur in music together with their given labels. The system is designed to achieve two goals:

(i) To detect the representative audio excerpt in music; (In particular, audio excerpt that can serve as a retrieval cue of music with which when one listens to this particular audio except, he or she would be able to tell weather this is the music that he or she is looking for).

(ii) To visualize all the repetitions that appear in a piece of music. This is to give a visual presentation of music and show key frames of important scenes occur in music.

Current literatures mainly focus on the significant of repetitions in music in identifying representative excerpts of music audio. Usually, the most repetitive segments are considered as the most significant excerpts to represent a piece of music. In our work, we explore the potential of audio content descriptors in capturing the specific features of the representative excerpt or 'hook' of the music. We acknowledge the significance of repetitiveness of music in human perception and cognition. However we hypothesize that besides this factor, audio content descriptors might also be useful in picking up audio excerpt that would give a stronger impression to listeners. Thus, in our study, we use two approaches in identifying representative excerpts of music. First approach emphasizes more on the significant of the most repetitive excerpts in music. Second approach considers all repetitions are equivalent. Thus, audio content descriptors are used to pick the best suitable audio excerpts to represent a piece of music.

This chapter introduces our proposed system in detail and presents an evaluation of the system's performance for each approach used in identification task. In Section 4.1, we present an overview of the system. In its subsections, we present in detail the descriptions of all processes carried out by the system. Section 4.2 presents a set of experiments performed on the system and their evaluation results.

4.1. Approach

Given that our goal is to identify the representative excerpts from audio signal, once may ask, "What is the criterion required for a music section to be acknowledged as a representative musical excerpt of an audio music?" As mentioned in a previous chapter, the repetition, transformation, simplification, elaboration and evolution of music structures has created the uniqueness of the music itself. Hence, many researches in this area have assumed that the most representative sections of music are frequently repeated within a song. In this study, we experiment in using different criteria to select the most representative excerpts from music audio signals and we evaluate objectively the performance for each selection criteria. To make this study possible, we have implemented our own system to carry out the identification task. Figure 4.1 below illustrates the overview framework of our system used for identifying representative excerpts from music audio. As shown in figure 4.1, our system involves 6 main processes with each undertake a different task. These main processes are as follows:

(1) Feature extraction: segment input signal into overlapped frame of fixed length and compute a set of audio features to describe audio content for each frame;
(2) Similarity measurement: compute similarity distance between each frame using selected audio features to measure the (dis)similarity between one frame and its neighbouring frames;
(3) Pre-processing: remove redundancies and enhance information supplied by the similarity representation for later processing;
(4) Repetitions detection: identify all repeated line segments that appear in music according to the similarity representation;
(5) Line segments integration: organise all the repeated line segments and recover undetected repeated segments in previous detection processes;
(6) Audio excerpt extraction: select and extract the representative excerpt l according to certain selection criteria;

The following subsections explain each process in detail.

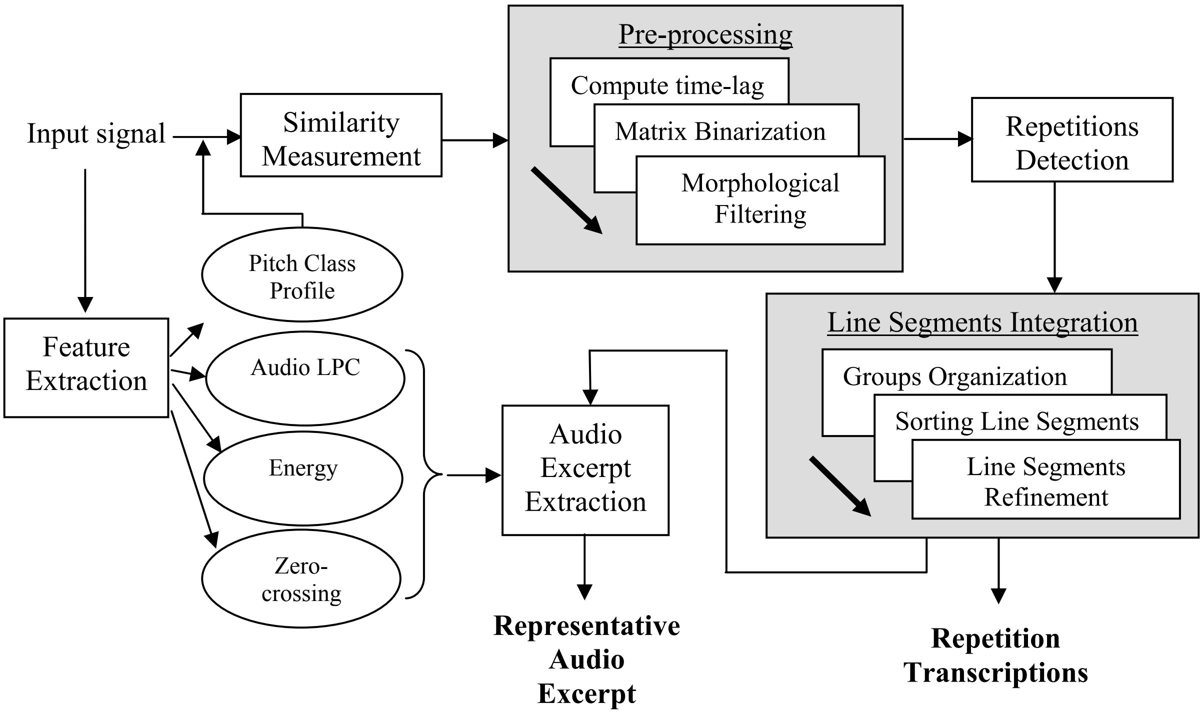


Figure 4.1. Overview framework of our approach in identifying representative excerpt of music audio signal

4.1.1. Features Extraction

Discovering music structure is a key issue in structural analysis research. Hence, extracting a kind of music representation from the audio signal is crucial in discovering the structure of music. Extracting symbolic score-like representation in music could be a possible way to complete the task. However due to the immaturity and high constraint of present sound source separation technologies, extracting symbolic representation of polyphonic music from raw audio signal is practically infeasible at present time. Otherwise, extracting low-level representations of audio signals for musical content description is found to be an alternative way for completing this task.

As mentioned earlier, melody has played an important role in music perception and music understanding with the implicit information that it carries. In fact, perceptual research studies [Dowling78, Edworthy85, Croonen94] have confirmed that contour can serve as a salience cue to melody recognition. Thus, in our approach towards identification of representative excerpt in music audio signal, we exploit melody-related features to first find the repeated patterns appear in music signal. Our audio input signals consist of polyphonic popular music with the presence of simultaneous pitches from instruments plus voices. Thus, we hypothesize that extracting melody-related features focused on pitch-chroma dimension (i.e. Harmonic Pitch Class Profile) would be an appropriate manner to deal with our input signals and apprehend a significant musical content.

Similar to any content based analysis, our system first requires the short-term descriptions of the input audio signal. The input signal is a complete full-length of music audio signal. Because of computational limitations, we usually limit the input signal to maximum 7 minutes in length. We first segment input signal overlapped frames (4096-samples window length) with the hop size of 512 samples. It is then followed by extracting feature descriptions of each of these frames. In our approach, we compute two groups of audio descriptors to be used for different tasks (i.e. repetitions detection; audio excerpt extraction). Table 4.1 below shows the grouping of the computed audio descriptors. .

Repetitions Detection	**Audio Excerpt Extraction**
Harmonic Pitch Class Profile (HPCP)	Linear Prediction Coefficient Zero crossing Energy

Table 4.1. The list of audio descriptors used for different tasks: (right) repetitions detection; (left) audio excerpts extraction.

HPCP, also called Harmonic Pitch Class Profile [Gómez04]: A 36-dimensional pitch-chroma related feature vector, based on the Pitch Class Profile introduced by Fujishima [Fujishima99], which measures the intensity of each twelve semitone pitch classes by mapping each frequency bin to each given class.

LPC, also called Linear Prediction Coefficients: An estimation of the spectral envelope of the signal. It is used to analyse the speech signal by modelling vocal tract transfer function by an all-pole filter with transfer function [Rabiner93]

$$H(z) = \frac{1}{\sum_{i=0}^{p} a_i z^{-i}} \tag{4.1}$$

where p is the number of poles and $a_0 = 1$. The filter coefficients, a_i, are chosen to minimize the mean square filter prediction error summed over the analysis window. Recently it has also been used as audio features to characterize between pure music and vocal music [Xu04].

Zero Crossings: A time-domain measure that gives an approximation of the signal's noisiness (see Section 2.3.1 for detail descriptions).

RMS energy: A measure of loudness of the sound frame (see Section 2.3.1 for detail descriptions).

4.1.2. Similarity Measurement

The second process in our system is similarity measurement. First, the system use the computed HCPC feature vectors as input and select a set of candidates for later processing. The candidate set consists of the first frame feature vectors from every 10 frames features with each represents the pitch class distributions of approximately every 116 millisecond of the original input signal. Here, we only consider the first frame feature vectors instead of all 10 frames features. There are two reasons for us to do so. First, it is to prevent our system from having high computational cost by processing the complete HPCP features of the input audio signal. Second, we assume that there are not much significant changes in terms of music context within such a short interval. Thus, taking the first frame features of every 116 milliseconds interval would be sufficient to seize significant changes of the music. From these candidates, we measure the (dis) similarity distance between each candidate, *v(n),* to its neighbouring candidates using cosine similarity function. The cosine similarity function calculates the dot product of the features vectors and normalized by their magnitudes. It produces distance measures within the range of 0 to 1. The similarity scores tend towards 1 when there is a strong similarity between two candidates. In reverse, the similarity scores tend towards 0 when there is less similarity between two candidates. The cosine of the angle between candidates' 36-dimensional vectors is given by the expression:

$$SD(X,Y) = \frac{X \bullet Y}{\|X\|\|Y\|} \tag{4.2}$$

We then embed the computed (dis)similarity distance values in a two-dimensional representation plot to reveal the repeated patterns occur in the musical structure of input signal.

4.1.3. Pre-processing

In order to ease the process of identifying repetitive segments in music, we compute the time-lag matrix of the similarity representation, was computed from the previous processing, by orientating the diagonal of the computed similarity matrix towards vertical axis. The rotated time lag matrix, $L(l,t)$ between chroma vector *v(t)* and *v(t-l)* is defined as

$$L(l,t) = SD(v_{t,} v_{t-l}) \tag{4.3}$$

Figure 4.2 illustrates the converted time-lag matrix with x-axis refers to the lag and y-axis refers to the time. The vertical lines, which appear to be parallel to the y-axis in time-lag matrix plot indicate the repeated segments appear in music. For instance a vertical line from $L(15,t_{begin})$ to $L(15,t_{end})$ in time-lag matrix denotes that audio section between t_{begin} and t_{end} seconds is a repetition of the earlier section from time (t_{begin}-15) sec to (t_{end}-15) sec. The length of each line segment appears in time-lag matrix plot indicates the duration of each repeated segment in music. In other word, line segments with long vertical line signify long repetition of music segments and vice versa. Hence by detecting the vertical lines appear in time-lag matrix, we would be able to obtain all the repetitions appear in music signal.

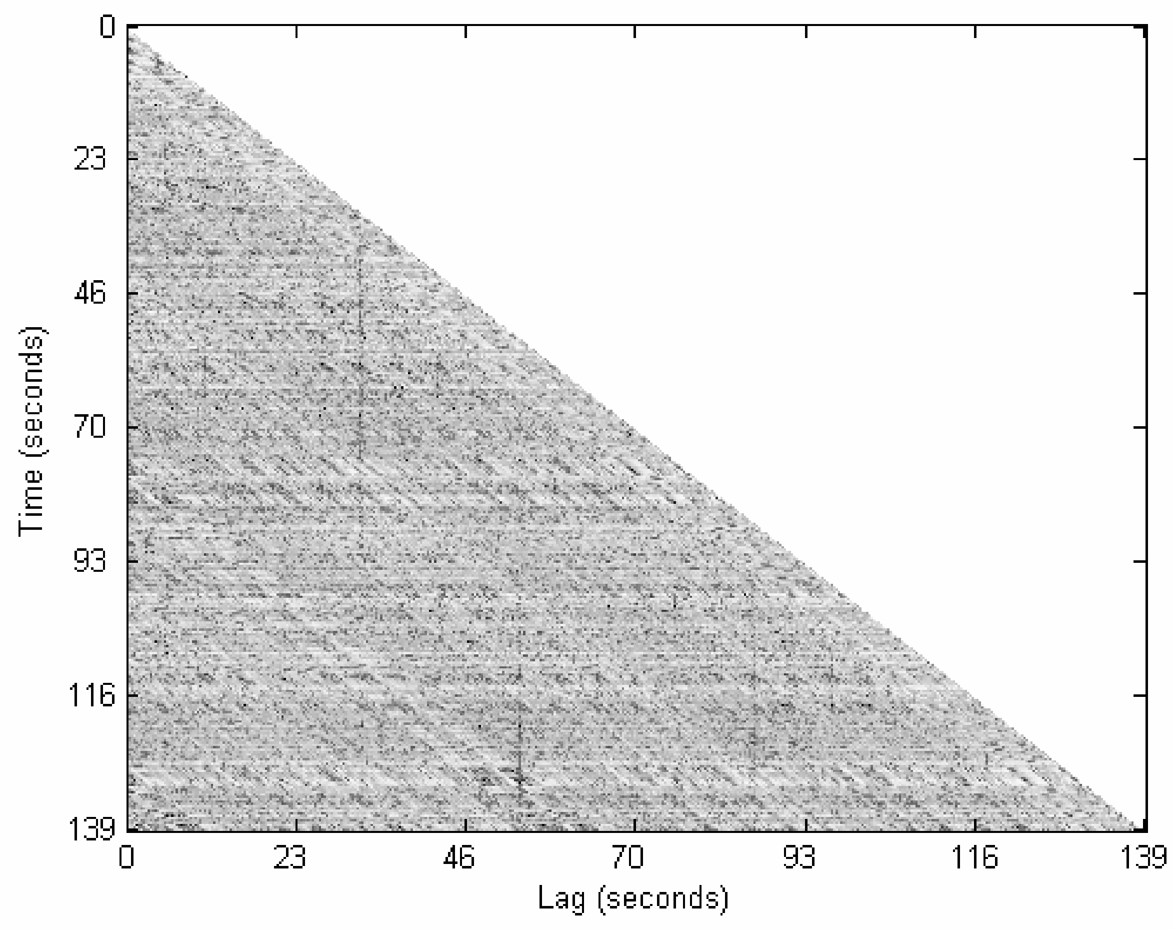


Figure 4.2 illustrates the time-lag matrix, *L*, for songs "I'm a loser" by The Beatles with its x-axis refers to the lag and y-axis refers to the time.

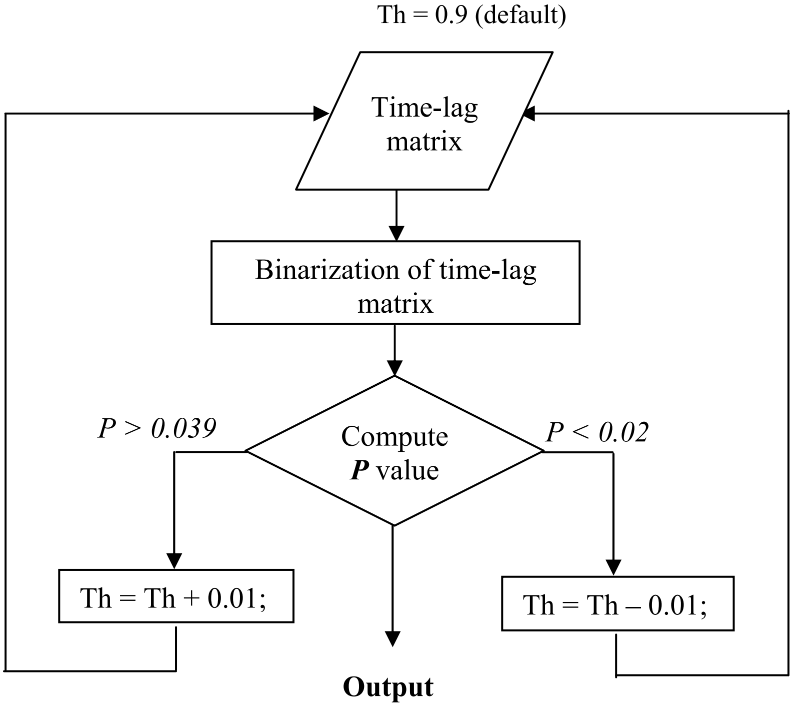


Figure 4.3. Flow chart illustrates the iterative binarization process.

As shown in figure 4.2, there exists much noise in time-lag matrix. Hence, in order to detect repetitions or vertical lines in the time-lag matrix, we would like to consider only a certain degree of similarities. Using a fix degree of similarities is not practical when dealing with broad categories of audio input signal, which may have different quality of recording and etc. Thus, in our approach, we perform a binarization process on the time-lag matrix based on an adaptive threshold to remove redundancies. The threshold, *Th,* used for binarization process will decide the degree of similarity measures to be retained from the matrix for later processing. The implementation of the binarization process is based on an iterative procedure as shown in Figure 4.3.

For initialization, our adaptive threshold holds a default value of 0.9. It means that we would only consider similarity measures with values more or equal to 0.9. We first binarize the similarity measures in time-lag matrix using the default threshold value. In another words, those similarity measures which are less than the threshold value are set to 0 whereas those higher or equal are set to 1. Then we compute a *P* value from the binarized matrix to evaluate the sufficiency of information it retained. The *P* value is defined as:

$$P = \frac{\text{total number of 1 in time-lag matrix}}{0.5 \times Area(\text{time-lag matrix})} \tag{4.4}$$

Based on the computed *P* value, we consider three cases as listed below:

(i) If $P>0.039$, increase the threshold by 0.01 and return to the beginning of the procedure;

(ii) Else if $P< 0.02$, reduce the threshold by 0.01 and return to the beginning of the procedure;

(iii) Else, quit the iterative process and output the binarized time-lag matrix.

For the first two cases, the iterative procedures continue with alteration to the threshold value. The whole iterative procedures only terminates when the third case is fulfilled. In such case, the system will output the binarized time-lag matrix and move on to the last operation in pre-processing section. It is to be mentioned that the values used in evaluating different cases in *P* are empirical results obtained through observing various input signals. When *P*>0.039, it denotes superfluous information in the binarized time-lag matrix. Thus, by increasing the level of threshold, *P,* it removes the redundancies. When *P*< 0.02, it denotes insufficient information contained in the binarized time-lag matrix. Thus, it is necessary to reduce the level of threshold, *P,* in order to yield more information for further processing.

The last operation of pre-processing section consists of applying opening operation of morphological filter, a widely used filter operation in image processing, to the binarized time-lag matrix. The functionalities of opening operation in our application are:

(i) to separate vertical line segments, which contain large gaps in between to several short segments;

(ii) to remove line segments, which are too short to contain any significant repetitions of music.

As showed in Figure 3.5.b, we can clearly see how 'Opening' operation open the gaps in the signal while removing success ones that are shorter than the structuring element in one-dimensional binary signal. As mentioned in Section 3. 1.2, Opening operation of morphological filter is based on erosion and dilation operation [Filonov05]. In general, dilation causes objects to dilate or grow in size while erosion causes objects to shrink. Opening operation works by eroding the signal followed by dilating the results. The amount of changes (grows or shrinks) depends on the choice of the structuring element. The following paragraph explains briefly how dilation and erosion work in detail.

As mentioned in Section 3.1.2, dilation works by moving structuring element over input signal and the intersection of structuring element reflected and translated with input signal is found [Young02]. Figure 3.3.a shows how dilation adding ones to runs of zeros that are shorter than the structuring element. While dilation works by moving structuring element over input signal. The erosion of input signal, A, and structure element, B, is the set of points x such that B translated by x is contained in A [Young02]. In contrast with dilation operation, the output is set to zero unless the input is identical with the structuring element. Figure 3.3.b shows how erosion removes runs of ones that are shorter than the structuring element.

Here, we treat our binarized time-lag matrix as one-dimensional non-binary signal. As mentioned before, opening operation works by eroding the signal followed by dilating the results (as illustrated in Figure 3.4b). Alternatively, we implement erosion operation, $Er(i,j)$, by first applying a zero-phase rectangular window, *w(n)*, along the perpendicular of the binarized time-lag matrix, *x(i,j)*, and compute the minimum value within each windowed signal. That is,

$$Er(i,j) = \min\{x(i,j+n)w(n)\}, \qquad -(N-1)/2 \le n \le (N-1)/2 \qquad (4.5)$$

where *x(i,j)* refers to the lag and j refers to the time of the binarized time-lag matrix. *w(n)* is the zero-phase rectangular window function, which used to define the minimum length of relevant line segments such that line segments shorter than this minimum length are to be removed from the binarized time-lag matrix, and is defined as

$$w(n) = \begin{cases} 1, & |n| \le \frac{N-1}{2} \\ 0, & \text{otherwise} \end{cases} \qquad (4.6)$$

We then perform a dilation operation to the eroded signal, $Er(i,j)$, by applying previously used rectangular window, *w(n)*, to $Er(i,j)$ and followed by computing the maximum value within each windowed signal. That is,

$$Op(i,j) = \max\{Er(i,j+n)w(n)\}, \qquad -(N-1)/2 \le n \le (N-1)/2 \qquad (4.7)$$

Finally, we yield a binarized time-lag matrix with removed vertical line segments that are less than the size of the window used in the morphological filtering operations. In our study, we have experimented in using different window lengths in order to find the optimal one for our proposed method. The experimental results are reported in later Section 4.2.2.

4.1.4. Repetition Detection (Listing the repeated sections)

The main goal of this process is to detect repetitive segments. This process requires the output data from post-processing process, $L_p(l,t)$, as an input signal. As mentioned earlier, vertical line segments in the time-lag matrix represent the occurrence of repetition in music. Thus, for finding the possibility of each lag for containing line segments, $P_r(l,t)$, we sum up each column of the time-lag matrix according to the lag. Since we only consider element below the diagonal of $L_p(l,t)$ and the number of element decreases corresponding to the increased of lag, we normalized the summation results with the total number of element in each lag. The calculation for the possibility of containing line segments, $P_r(l,t)$, of each lag is defined as:

$$P_r(l,t) = \int_l^t \frac{L_p(l,\tau)}{t-l} d\tau \qquad (4.8)$$

Figure 4.4 illustrates the possibility of containing line segments, $P_r(l,t)$, corresponding to each lag. High $P_r(l,t)$ marks frequent repetitions whereas low $P_r(l,t)$ marks that infrequent repetitions occur in lag, *l*.

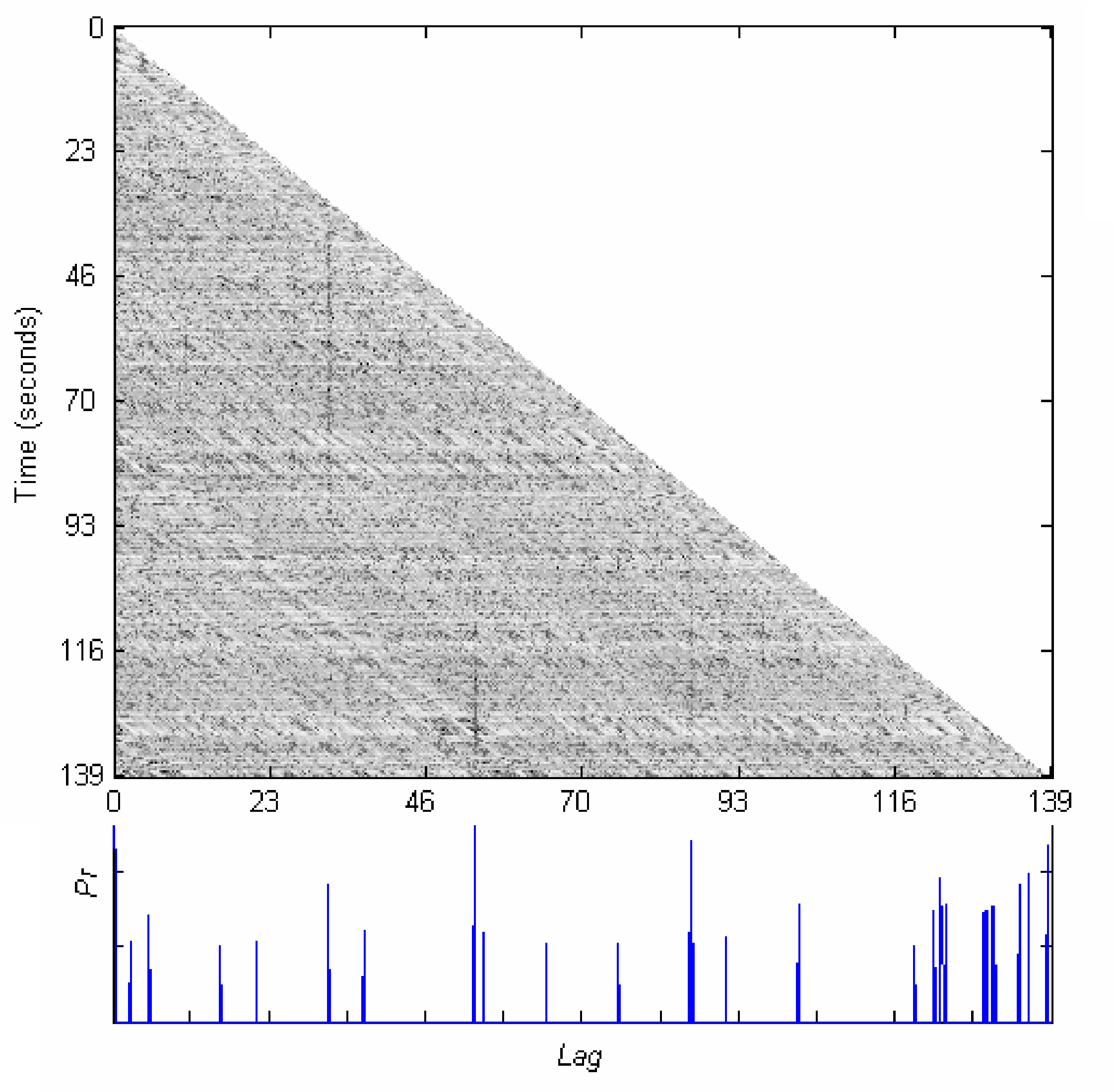


Figure 4.4. The possibility of containing repetition, $P_r(l,t)$, corresponds to each lag.

For searching line segments, we select all peaks appear in $P_r(l,t)$ and store their lag information in a descending order as $l_{PeakSort}$. We then evaluate the occurrence of line segments in $L_p(l,t)$ alternately for each element in $l_{PeakSort}$. We compute $L_p(l_{PeakSort},t)$ for each $l_{PeakSort}$ and search for the occurrence of vertical line segments. Here, we hypothesize that repetitions, which hold less than 4 seconds (or less than 2 bars length for a music piece with a tempo of 120 beats-per-minute in 4/4 time signature), do not carry much significant musical information. Thus, for detecting repetitive segments in music, we only consider those line segments with duration longer than 4 seconds. For each detected line segment, we store the beginning and ending time of the repeat segment together with its repeated segment based on $l_{PeakSort}$ information.

For instance, when a line segment between $L_p(l_{PeakSort} = l_p, T_1)$ and $L_p(l_{PeakSort} = l_p, T_2)$ is located, it means that segment between time T_1 and T_2 is repeating the earlier segment at time $T_1 - l_p$ until $T_2 - l_p$. Hence, by the end of an iterative detection process, we yield a set of repetition pairs. Pseudo code shown in figure 4.5 outlines the above mentioned line segments searching algorithm.

```
Select peaks from $P_r(l,t)$
Let $l_{PeakSort}$ = peaks' lag information in $P_r(l,t)$
Sort $l_{PeakSort}$ by descending order

FOR each of the $l_{PeakSort}$
Search line segments appear in $L_p(l_{PeakSort}, t)$

    FOR each of the obtained line segments
        IF length of line segment less than 4 seconds
            Delete line segment
        ELSE
            Store starting time and ending times of line segment
            Store starting time and ending times of repeated line
            segments
        END
    END
END
```

Figure 4.5. Pseudo code outlines the line segments search algorithm.

4.1.5. Integrating the repeated sections

In this section, we organize the detected repetition pairs obtained from previous steps into groups. Apparently, line segments that share a common line segment are the repetitions of one another. Thus, if these segments are to be labelled, they should be given the same labelling. Based on this observation, we integrate those line segments, which share a common line, into a same labelled group. From this, we yield a set of repetition groups with different labels marking the different repetitive segments appear in music piece. That is

$$Group_{repetitions} = \{Group_1, Group_2, ..., Group_n\} \tag{4.9}$$

where n is the number of repetition groups. In each repetition group, we sort the repeated line segment in an ascending order based on their time information and is represented as

$$Group_A = \{[Tbegin_1, Tend_1];[Tbegin_2, Tend_2];...;[Tbegin_m, Tend_m]\} \tag{4.10}$$

$$where \quad Tbegin_1 < Tbegin_2 < ... < Tbegin_m$$

Tbegin and *Tend* denote the beginning time and ending time of the repetitive segments whereas *m* is the number of repetitive segments in $Group_A$.

For the refinement of line segments, we select the fist line segment from each group in $Group_n$, and correlate it alternately with the pre-processed features, *v(n)*, as mentioned in the earlier section 4.1.2. This is for the purpose of recovering undetected repetitions that we have missed in the previous detection process. We slide the compared segment's feature representations along *v(n)* and compute the Euclidean distance measure between features to find the repetitions. The computed distance measures are within the range of 0 and 1 with low distance measures indicate strong correlations with the compared segment and vice versa. In fact, when there exists 0 in the computed distance measures, it marks the correlation of the compared segment to itself. Figure 4.6 illustrates the correlation between compared segment with pre-processed features, *v(n)* corresponds to time. As shown in figure 4.6, the self-correlated compared segment occurs after 31 seconds of the starting point of the song, marking the actual time position of the compared segment in the input music signal.

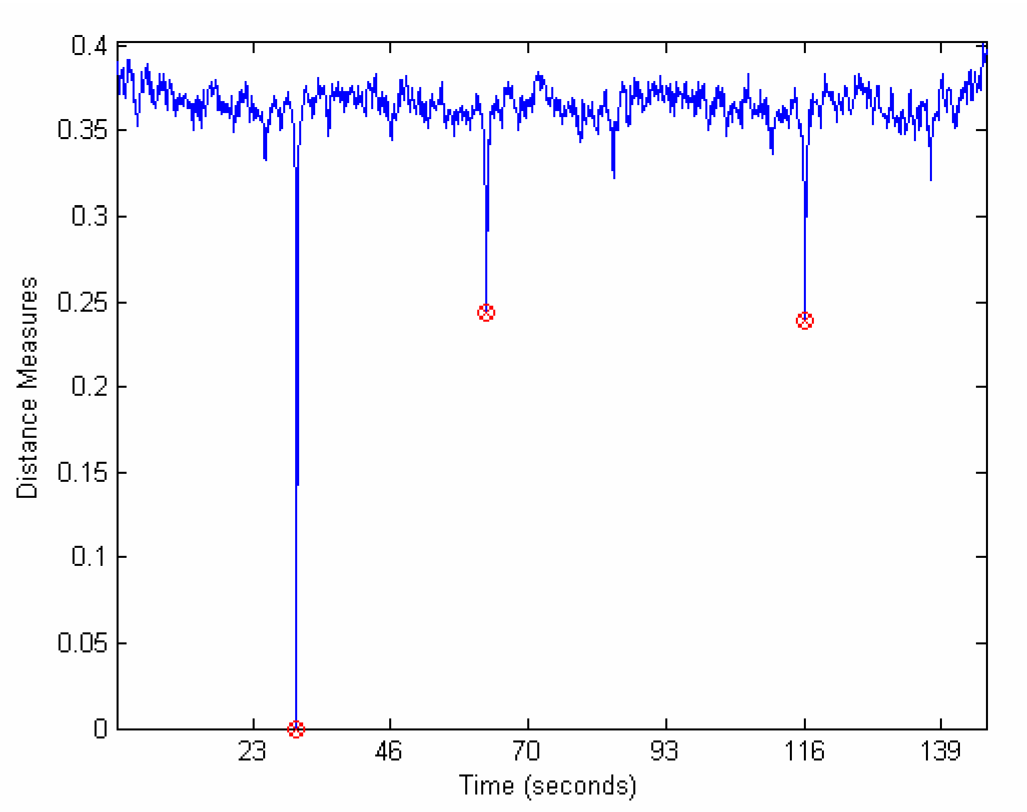


Figure 4.6 illustrates the correlation between selected segment with pre-processed HPCP features, *v(n)*. Cross-circled marks the selected local minima based on adaptive threshold.

To detect significant repetitions appearing in music, we use an adaptive threshold based on the computed distances. Excluding the distance of the compared segment to itself (always zero), we select the lowest occurring distance value. To obtain the adapted threshold, we add a tolerance margin of 0.02 to this value. Then, all local minima falling below the threshold are considered to be relevant to the occurrence of repetitions. We sort the considered local minima based on the distance measure in a descending order. With the length of the compared segment, we estimate and store the corresponding beginning time and ending time for each considered local minimum and form a set of candidate segments. We hypothesize that repetitions of a segment do not overlap with each other. Hence, we disregard those candidate segments that overlap with any of the line segments in the group that hold the same label as the compared segment. The remaining ones are labeled and included in the correct group as omitted repetitions from the earlier detection process. We then reorganize line segments in the group with an ascending order based on their time information. It is similar to the earlier sorting processes of the line segments for each group in *$Group_n$*. Pseudo code shown in Figure 4.7 gives a rough outline of the above mentioned refinement algorithm in recovering omitted repetitions from the earlier detection process.

```
FOR each repetition group in $Group_n$
    Let segments_inGroup = line segments in the repetition group
    Find lowest distance value besides zero

    Let lowest_distance = lowest distance value
    Select local minima within lowest_distance ± 0.02
    Sort selected local minima by descending order based on distance value

    Let Z = length of a line segment in segments_inGroup
    FOR each selected local minima
        Compute starting time and ending time of local minimum based on Z
        Store starting time and ending time of local minimum
        IF overlap with any segments_inGroup
            Remove selected local minimum
        ELSE
            Label and insert selected local minimum in segments_inGroup
            Sort segments_inGroup in ascending order based on its time
        END
    END
END
```

Figure 4.7. Pseudo code outlines the line segment refinement algorithm.

4.1.6. Audio excerpt identification and extraction

The last process in our system seeks to identify the relevant repetition group and generate a representative audio excerpt amongst the repetition segments. In another word, we would like to generate an audio excerpt, which captures the retrieval cue or the gist of the music input signal. There are two ways to approach this matter. One way would be to consider the frequencies that a segment is repeated. This is based on the assumption that the most repetitive segments are the most representative segment in music. So far, this has been the most adopted assumption for generating the most representative excerpt or thumbnail of the music in audio research. The other possible way would be to consider the potential of audio descriptors in capturing the unique features of the ‘gist’ or ‘hook’ of the music. In popular music, we can assent to the viewpoint that “chorus” or “refrain” sections often capture the hook or essence of the music. In musical terms, “chorus” or “refrain” section is defined as the part of a song where a soloist is joined by a group of singers. Others define “chorus” as a part of the song, which often sharply contrast the *verse* melodically, rhythmically and harmonically and often involve higher level of dynamics and activity with added instrumentation. Thus, based on this definition, we hypothesize that certain audio descriptors would be capable of capturing these unique characteristics and facilitate the identification of “chorus” or “refrain” section.

Here, we pursue both assumptions for the task of identifying the most representative excerpt in music. In our study, we extract audio excerpt in two strategies by alternately putting a higher priority on each undertaken assumption. First one gives more priority to the repetitiveness of audio segments while second one focuses more on content descriptions of the unique characteristics of music sections. In the following two paragraphs, we discuss both strategies identifying the representative excerpt in music in detail.

As mentioned above, first manner gives more priority to the repetitiveness of audio segment. Since repetitiveness of each repetition group is defined by number of line segments it encompassed, we begin the identification process with calculating the number of line segments included in each repetition group. Repetition groups with the highest amount of line segments are selected as group candidates for having the possibility of comprising the most representative segments. The idea of extracting representative excerpts comes from applications such as music recognition, audio browsing, audio thumbnailing and so forth. Thus, for selecting representative excerpt of music for such applications, we would prefer to extract audio segment with singing voice instead of purely instrumental as we believe that segment with singing voice are more recognizable than those with purely instrumental. We exploit the uniqueness of zero-crossing rate in distinguishing between pure instrumental and vocal music to avoid extracting purely instrumental segment from the selected group candidates. Figure 4.8 below plots the zero-crossing rate (ZCR) of a pure instrumental segment and a vocal presence segment of a music piece. Comparing both, pure instrument segment mostly stays within a relatively small range of amplitude. We compute the average zero-crossing rate for each segment in the selected candidates group and select those with the highest value as the most representative segment. In addition to zero-crossing rate, we also study the utility of energy feature in making segment selection since we believe that segment with higher dynamics level would give a stronger and more durable impression to listeners instead of a lower one.

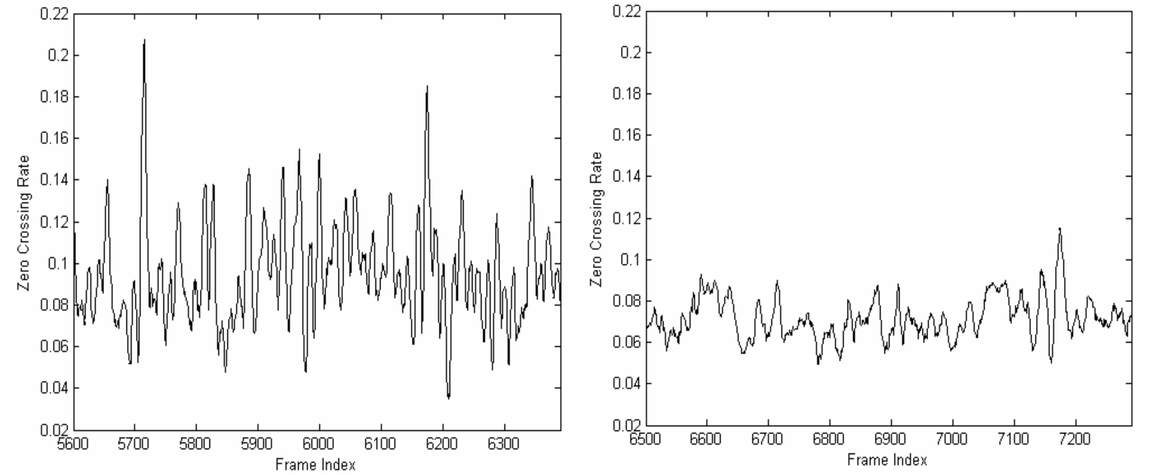


Figure 4.8. An example that illustrates the zero-crossing rate (ZCR) of pure instrumental segment and vocal presence segment taken from different segments of The Beatles' song entitled "I'm a Loser".

In the second manner, we focus more on the content descriptions of the unique characteristics of music sections instead. Here, we disregard the repetitiveness of audio segments but consider all repetition groups are equally important in making group candidates selection. Contrary to the first means, we put more attention on the potential of audio descriptors in grasping the specific features of the 'hook' or 'gist' in music. We undertake a few audio descriptors used in music identification (i.e. zero-crossing rate [Zhang03] and linear prediction coefficients [Xu05, Xu04]) to study their effectiveness in identifying the most representative excerpt of music piece. In addition, we also consider the energy features of the segments since we assume that the presence of joint performance activity in "chorus" or "refrain" section would render a distinctive deviation with other musical sections. We compute the mean value of audio features for all audio segments in each repetition group. Then we select the maximum mean value among the segments in each group and compare with other groups. The group, which has the highest feature value, will be considered as the group candidate and the segment that has the maximum feature value would be extracted as the most representative excerpt. As the linear prediction coefficients (LPC) exploited here form a 12-dimensional vector of audio representation, the way of using it to select group candidates is somewhat different with other features. Here, we do not exploit all LPC coefficients but only consider a few selective ones such as, the 2nd, 4th and 6th, based on their sensitivity in capturing frequency changes within vocal band. Figure 4.9 illustrates an example of the sensitiveness for section changes of the LPC coefficient. As shown in figure 4.9, chorus sections in music appear to have a higher 2nd LPC coefficient compared to other sections in music.

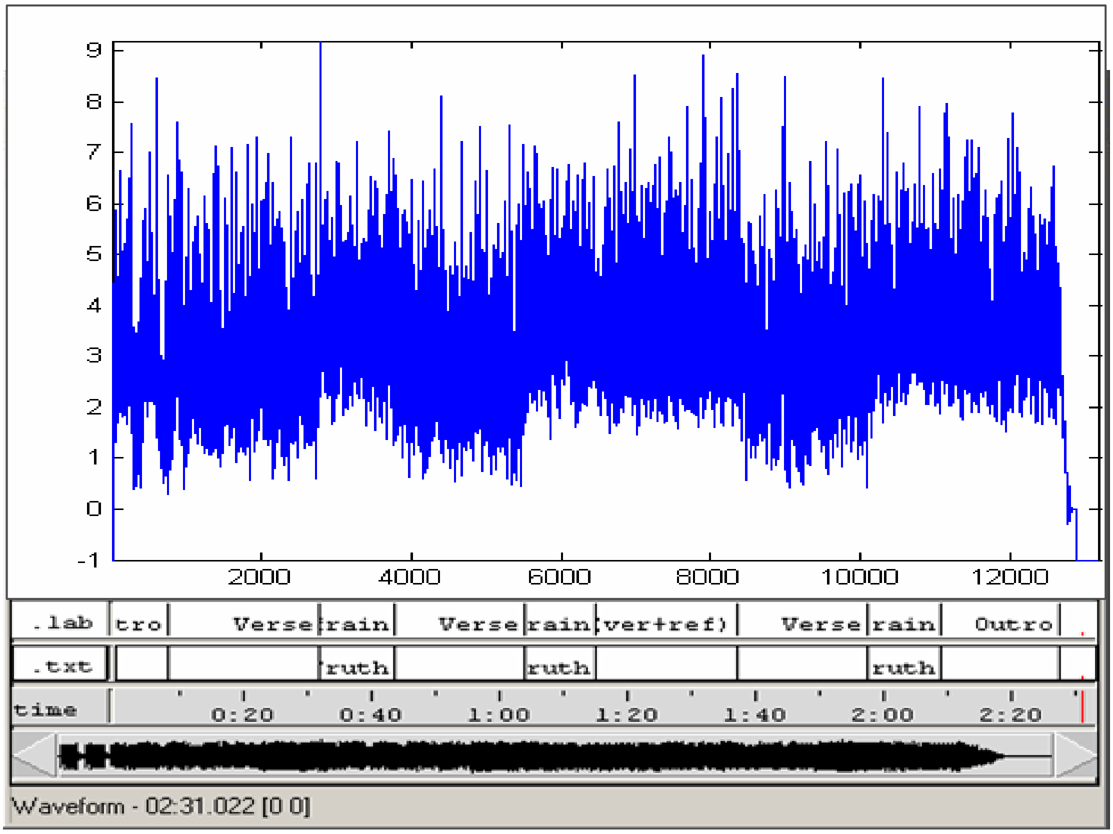


Figure 4.9. An example that illustrates the sensitivity of LPC coefficient corresponds to section changes.

First we compute the mean of the selected LPC coefficients for every segment in each repetition group. Then, we select the segment with the highest mean value among all segments of all repetition groups for each considered LPC coefficient (2nd, 4th and 6th) successively. After this process, we have identified up to three different segments that contain the highest mean values for the considered coefficients. We then take the majority vote to find the repetition group candidate. Since we are considering three LPC coefficients (odd number), we set a criterion such that the repetition group, which comprises segments holding more than one of these three highest mean values in total, would be regarded as the group candidate having the most representative excerpt amongst its segments. If none of the repetition groups comprises segments holding more than one of the highest mean values in total, the repetition group holding segment with the highest mean value of 2nd LPC coefficient would be considered as the group candidate. In order to select a segment from the group candidate, we follow previously defined criterion for group candidate selection procedure. Segment that holds more than one of the highest mean values would be regarded as the most representative segment. If none of the segment fulfills the criterion, the segment with the highest mean value of 2nd coefficient would be considered as the most representative segment.

Finally, we extract the select segment and output the excerpt in a WAV format as the most representative excerpt of the music input signal. In addition, we also output the transcriptions of all the repetition segments in the repetition groups with .lab format, which comprise the beginning time, ending time together with the segments given label. This is to provide a visual cue of different repetition patterns that occur in music input signal.

4.2. Evaluation

In this section we present an evaluation of our system's performance when identifying representative excerpts on a dataset. We first begin by presenting in detail our test data set and the labelling procedure. It is then followed by explaining three evaluation measures used to assess the performance of our proposed method. Then, we show the quantitative evaluation results of our system's performance based on having different priority to the undertaken assumptions (as mentioned in the earlier section) in extracting representative excerpt. Finally, we present our system's performance results based on applying different window size of morphological filtering as described in section 4.1.3.

4.2.1. Data set

In our experiments, we use a dataset, which consists of 28 popular songs, which comprise a chorus section, from various artists. Each song is sampled at 44.1 kHz, 16-bit mono. For evaluation purposes, we have generated a ground truth by manually labelling the “chorus” section for all song in the dataset. Only one person (the author) labelled the data.

4.2.2. Quantitative performance

In quantitatively evaluating our results, we compare the detected excerpts with its manually labelled ground truth results. We use standard measures in information retrieval to discuss the detection results from our algorithm. Since it is quite common that there exist more one “chorus” section in a music, the manually annotated ground truth results may contain a set of n truth intervals of the music, such that $X_{truth} = \{x_1, x_2, ..., x_n\}$. Thus, we use one of the truth intervals, the one that overlaps the most with our detected excerpt, $x_{Detected}$, to measure the precision and recall rate. The precision and recall measures are defined as

$$\text{Precision, Pr} = \frac{|x_{Detected} \cap x_i|}{|x_{Detected}|}, \qquad \text{where } i = \arg\max_{j=\{1..n\}} |x \cap x_j| \tag{4.11}$$

$$\text{Recall, Re} = \frac{|x_{Detected} \cap x_i|}{|x_i|}, \qquad \text{where } i = \arg\max_{j=\{1..n\}} |x \cap x_j| \tag{4.12}$$

For evaluating the overall efficiency of the system, we measure the F-measure, which is defined as

$$\text{F-measure} = \frac{2 \times (\text{Precison} \times \text{Recall})}{(\text{Precison+Recall})} \tag{4.13}$$

As mentioned in section 4.1.6, we have employed two approaches in identifying representative excerpts from music signals. First approach puts higher priority on the reappearances of a segment in music, whereas the second approach focuses more on the content descriptions of a particular section in music, which are "chorus" sections for most of the cases of popular music. Table 4.2 shows the objective evaluation results based on these two approaches with a filter length of approximately 0.3 seconds. Basically, "Repetitive_ZCR" ("Repetitive_Energy") and "ZCR" ("Energy") showed in table 4.2 used the same audio descriptor (zero-crossing rate or energy) in making segments selection from the repetition group candidates. The difference between them is the criterion in deciding repetition group candidates. In selecting the repetition group candidates, "Repetitive_ZCR" ("Repetitive_Energy") based on the reappearance frequency of segments in each repetition group, whereas "ZCR" ("Energy") based on the content descriptions of all segments in each repetitive group. The objective evaluation results (in table 4.2) demonstrate a lower precision and recall scores for repetition-based approaches (i.e. "Repetitive_ZCR" and "Repetitive_Energy") compared with content-based approaches ("ZCR" and "Enegy") in identifying representative excerpts of music.

Based on content-based approach, we further study the significance of various descriptors in identifying representative excerpts of audio with our system. Figure 4.10 illustrates the objective evaluation results of the system performance based on exploiting various descriptors in the excerpt identification task. "Energy_ZCR" ("ZCR_Energy") denotes the use of energy (zero-crossing rate) features to select repetition group candidates and followed by zero-crossing rate (energy) feature in making segment selection amongst others segment in the group candidate. While LPC (ZCR or Energy) signifies the use of only a specific audio descriptor for both repetition group candidates and segment selections. The plot in figure 4.10 shows a distinctive performance of LPC amongst other used audio descriptors. LPC achieves the average precision and recall rates of 78% and 57% for identifying the 28 songs in our test dataset. Precision score represents the length of the identified excerpts, which enclosed the "chorus" section, over the length itself. While recall rate describes the proportion of the chorus sections that are correctly identified.

As mentioned in section 4.1.3, we have employed morphological filters to remove small fractions of line segment, which are considered to contain little significant repetitions of music. As the length of the fractions to be discarded directly depends on the filter window size, we have investigated the effect of alterations in the size of window on our system's performance. We rely on LPC descriptor to handpick both repetition group candidates and the most representative excerpt. Figure 4.11 demonstrates our system's performance corresponds to different window size in morphological filtering based on LPC. 0.0 window size in the figure denotes without the use of morphological filter. The figure shows the dependence of the precision and recall scores on window length. The optimal performance lies on the window size of 30 frames (approximately 0.3 seconds) with a precision and recall scores of 78% and 57% respectively. The overall F-measure reached 66%. Compare to the performance without applying any morphological filtering (0.0 seconds), this is an increase of more than 13% in both measures.

Approaches		**Precision**	**Recall**
(a) Repetition-based	Repetitive_ZCR Repetitive_Energy	0.5268 0.5693	0.3519 0.4045
(b) Content-based	ZCR Energy LPC	0.6270 0.5980 0.7762	0.4436 0.4533 0.5698

Table 4.2. Precision and recall rate for content-based approaches and repetition-based approaches used in identifying representative excerpt of music.

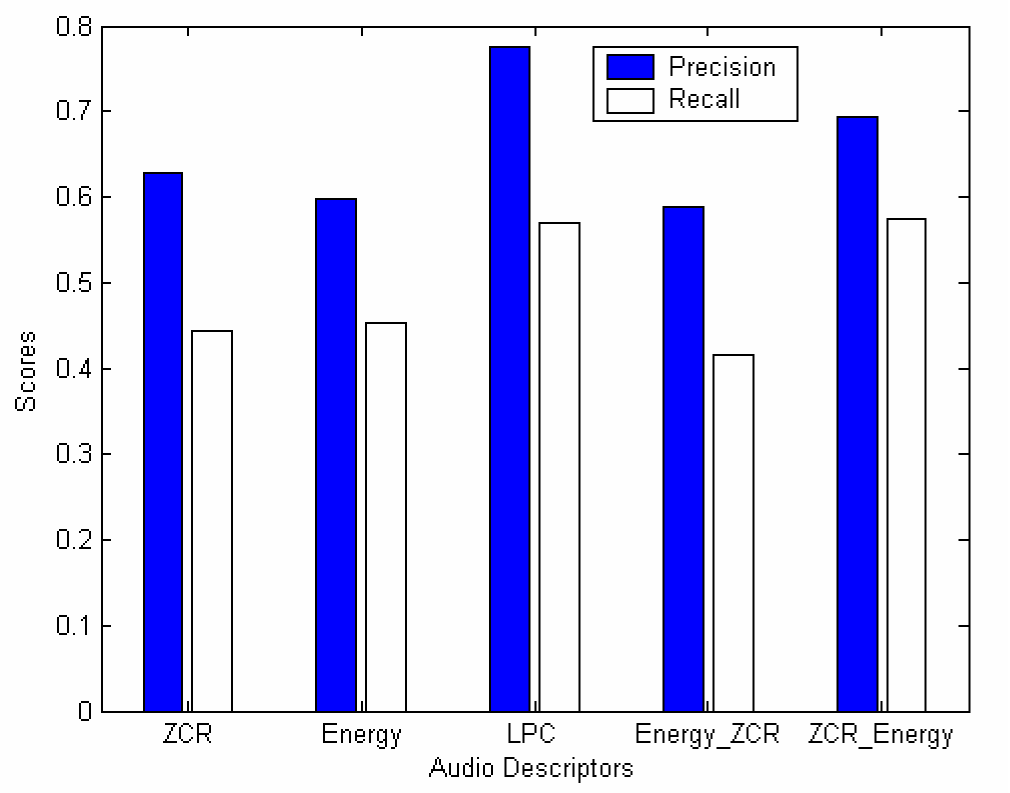


Figure 4.10 plots the system performance based on content-based approach with various audio descriptors.

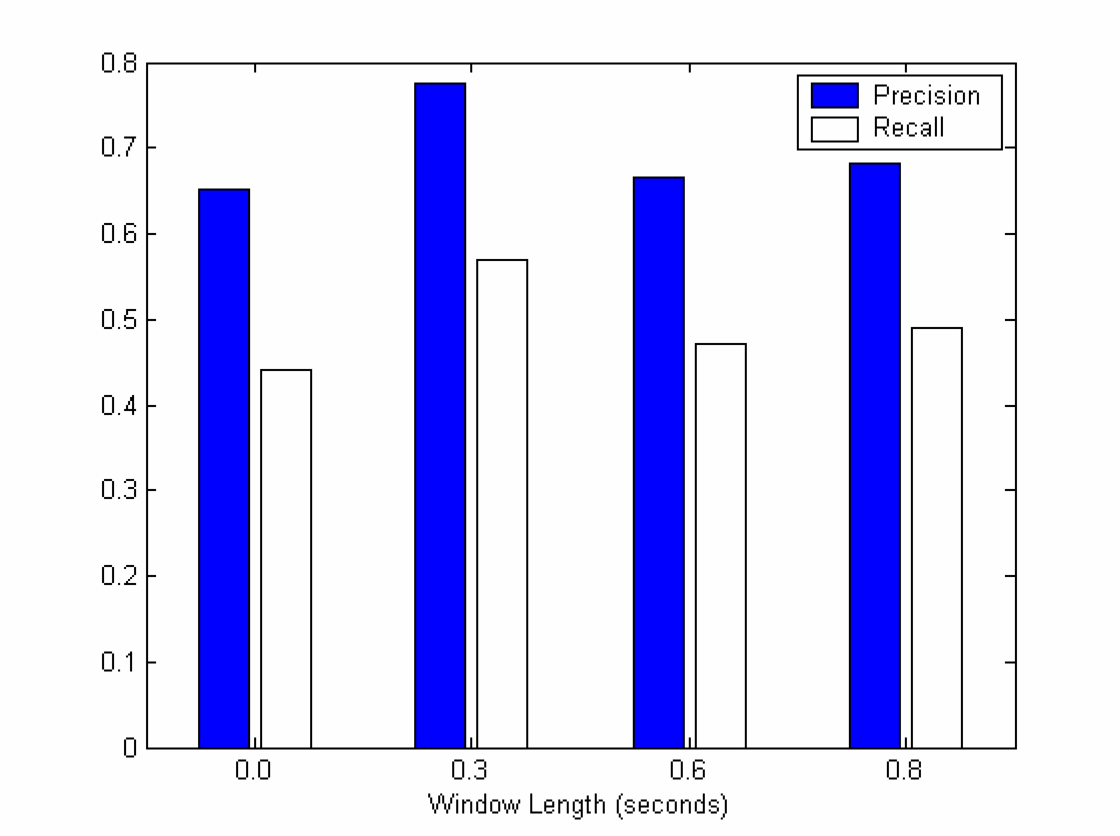


Figure 4.11. The results of representative excerpt identification based on LPC by using different window length of morphological filter.

4.3. Discussion

In this section, we summarize the key findings from the number of experiments discussed in this chapter. Perhaps the most notable results from this chapter's experiments are the distinctive dichotomy in performance among the two distinctive based selection criteria. For the segment selections that make use of content-based approach, we see generally good performance. This dichotomy has generally supported the notion that repetitiveness of music segments is important in identifying representative excerpts of music. However this is not the only assumption what we should rely on. Incorporating some other assumptions based on musical knowledge could somehow give a prominent performance.

Another primary result is importance of the choice of features. The results presented here have made a strong case for the potential of audio content descriptors in capturing specific characteristics of particular section in music. The results show quite a significant improvement of performance by using LPC compared to other descriptors. However it does not unequivocally support the notation that LPC descriptor absolutely overcomes other descriptors in identifying representative excerpts since there is a limitation with the size of our dataset. Thus, further evaluation is still needed to verify this generalized notation.

Finally, the performance results corresponding to different filter lengths used in pre-processing stage has also shown the effectiveness of morphological filtering application in our system. The results demonstrate a relatively low performance of our system when there is no applied morphological filtering operation. In contrast, when they are in operation, the performance of the system improves substantially.

4.4. Summary

In this chapter, we have presented our approach to identify representative excerpts from music audio files. We have investigated the potential of using a different approach from what has been explored in the existing literature. Experiments were conducted to evaluate the performance of our proposed approach for polyphonic audio recordings of popular music. Additionally, we have also studied the effectiveness of morphological filters for pre-processing the signal prior to the identification process by using various window sizes. We have showed the performance of using two distinctive approaches, which focus on different music aspects, in making selection criteria for identifying representative excerpt of music. Overall system performance is found to be the highest for content-based approaches instead of repetition-based approaches. Selective audio descriptors have also been studied for its capability in capturing specific characterization of music section. So far, the objective evaluation results from our test dataset show that LPC descriptors have overcome the rest of the studied audio descriptors and have yielded an eminent performance in identifying representative excerpts of music. However further studies are required to be done in order to verify this observation.

In the next and final chapter, we will present a summary of the main conclusion from this work. Additionally, we will present suggestions for improvement, open questions and potential areas for our future work that will be the core of our PhD dissertation.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

In this study, two main aspects of music content processing related to structural analysis (semantic audio segmentation and identification of representative excerpts of music) have been investigated. In this chapter, we summarize the conclusions that we have drawn from our studies and present some avenues for future work.

5.1. Conclusions

In this research work, we have fulfilled our initial goal of identifying "singular" within-song excerpts in popular music by proposing and developing frameworks and methods in two areas that are closely related to automatic audio-based music structural analysis: (1) Semantic audio segmentation; (2) Identification of representative excerpts of music

In our semantic audio segmentation study, we have proposed a two-phase approach to segment audio data according to the structural changes of music and to provide a way to separate the different music "sections" of a piece, such as "intro", "verse", "chorus", etc. We have also proposed a combination set of audio descriptors that has proved useful in detecting music structural changes. Evaluation tests have been done to assess the performance of our proposed method with the use of a test dataset which consists of 54 pop songs. So far, our proposed semantic audio segmentation approach has achieved an accuracy higher than 71% and a reliability of 79% using the ground truth set. The overall F-measure has reached 75%. In other words, with 10 detected segment boundaries, 7 of them are correctly detected compared to the ground truth data. From the quantitative evaluation results, we conclude that the exploitation of image processing techniques (i.e. morphological filtering) is significantly profitable in enhancing the detection of segment boundaries corresponding to the structural changes and facilitate semantic segmentation of music audio. By having two phases of segmentation, first focusing in rough segmentation and later in further refinement, we have yielded a semantic audio segmentation algorithm that is useful and relatively reliable for practical applications. Coupling semantic audio segmentation functionality into both hardware and software platforms of digital audio players or sound visualization and manipulation applications (i.e. WaveSurfer) will allow users to skip from one section to another section of music easily and precisely. This is certainly a big improvement over the conventional fast-forward function mode.

For the identification of the representative excerpts of music, we have considered other factors in the identification task than the one appearing in the literature. One of our primary hypotheses states that repetitiveness of music may not be the only element in detecting the 'hook' of music. Preliminary evaluation results of our work have shown some evidence to support this hypothesis. The overall performance of our system is found to surpass other content-based approaches, which put higher priority on using content descriptors in selecting representative excerpts, compared to repetition-based approaches, which put more attention on the repetitiveness of segments. Particularly, our content-based approach with the aid of LPC descriptors has achieved an accuracy of higher than 77% whereas the repetition-based approach has only reached 57% the most in its accuracy rate. This result encourages the consideration of using additional musical knowledge-based assumptions, which have not yet been explored by current references, to further improve the performance of finding representative excerpts of music. On the other hand, since our approach begins with the identification of repetitions that appear in the music, we are able to exploit the obtained information to visualize the structure of music. Thus, the identification of representative music excerpts does not only bring the benefit of giving an abstraction cue, but also a clear visual-structural representation of the music. Apparently, coupling both functionalities into audio playback devices will allow fast browsing of music data. In addition, an auxiliary add-in playback mode on in combination with the segment block structural visualization will allow users to have easy access to any particular segment of the music by just clicking on the visualization's segment block. Figure 5.1 shows an example of a sound visualization system coupled with music structure visualization and add-in segment playback functionalities.

Our general conclusion obtained by reviewing the current developments in this area states that there exist a few limitations with respect to algorithm evaluation in present literature works. The first limitation is the lack of generality of the test databases. Databases consisting of a few hundreds of songs with different diversity and complexity will not be able to reflect the real-world music specificities. Hence, by using such a database, it is quite difficult to obtain an objective evaluation of the algorithm efficiency in the general context of all existing music. Online evaluation tests, by giving access permission to internet users to run the algorithm on their audio files and acquiring feedback from the users, would be a way to have the algorithm running on an unlimited range of music and obtain an appropriate assessment on the algorithm efficiency to real-world music, even though there may have some other factors that would need to be considered.

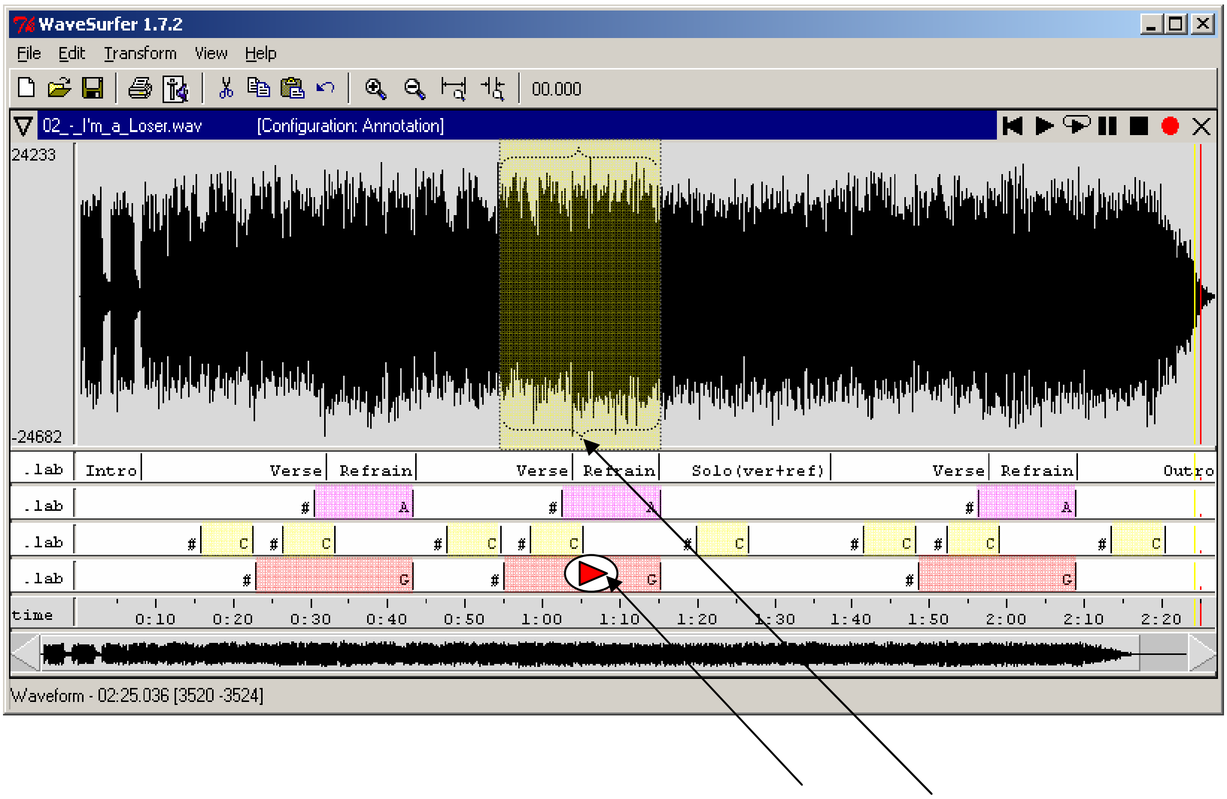
Another limitation is the method used to weight the importance of extracted music sections. The significance of the musical excerpts in audio signal highly depends on human perception. Musicians and non-musicians may not have the same viewpoint on “which sections are the representative excerpts of a piece of music”. A musician may have a strong impression of the solo instrumental sections whereas it may not be the case for a non-musician. Hence, it would be useful to have two groups of listening subjects and taking into consideration the differences between these two groups when evaluating the significance of the extracted music sections.

5.2. Future Work

Apparently, our work in semantic audio segmentation and identification of representative excerpts of music audio are relatively preliminary works and there are still many aspects that can be improved.

5.2.1. Semantic Audio Segmentation

In semantic audio segmentation, quantitative evaluations have shown a significant improvement in both precision and recall measures compared to previous method described in [Chai03c]. It should be noted however that the generality of our music database is quite limited. So far, we have not yet tested our approach on different music genres (i.e., instrumental music, techno, jazz or computer music). Thus, in our future work, we will take into consideration some other different music genres that we have not yet explored, in order to assess our proposed method on a wide generality of music applications.



Click & play

Figure 5.1 An example of a sound visualization system coupled with music structure visualization and add-in segment playback functionalities.

In addition, it is worthwhile to pay attention to the fact that the precision and recall rates of our proposed segmentation method are particularly low for those songs which include smooth transitions between sections. It seems that our descriptors are not sensitive enough to mark these changes. On the other hand, songs with coarse transition between sections usually achieve a better rate on these measures. Thus, using some other disregarded descriptors perhaps we will be able to cope with this matter.

Currently, we allow a tolerance deviation of 3 seconds from manually labeled boundaries for our segmentation algorithm. Thus, our detected segment boundaries sometimes may differ up to 3 seconds from the manually labeled boundaries. Practically, the less divergence the detected segment boundaries show from the manually labeled boundaries, the better and more precise the segmentation algorithm is. Thus, we may make use of higher-level analysis technique, such as beat detection, phrase detection and so forth, for truncating segments according to the beat or phrase information provided by the algorithms to achieve better segment truncation.

In the segment boundaries refinement phase, so far we have only exploited the mean value of the attribute to represent feature vectors in each detected segment from phase 1. Thus we plan to explore the applicability of some other descriptive statistics (i.e., variance, median, quartile and etc.) for describing the pattern and trend of the feature data and have better representations of the used features.

Finally, an interesting direction is the use of computed segment information to automatically label sections according to their structural titles, such as intro, verse, bridge, outro, etc. By doing this, we can provide a more informative description of the structure of music.

5.2.2. Identification of Representative Excerpts of Music Audio

Apparently, our work in identification of representative excerpts of music audio is still in the preliminary stage and there are many (see above) that can be improved. The overall performance of our system is found to be higher for content-based approaches instead of repetition-based approaches. Thus we plan to make more detailed analyses on the use of some other audio descriptors in detecting representative excerpts of music.

It is important to note that objective evaluation results from our test dataset have shown that LPC descriptors have outperform the rest of the studied audio descriptors and yielded an eminent performance in identifying representative excerpts of music. However, due to limitation of our small database, it is not possible to make any justification of this observation. Currently, we are in the process of collecting more data for our test database to verify this observation.

Our current algorithm for the identification of representative excerpts suffers from quite a low recall scores. Thus, we plan to embed our semantic audio segmentation approach into the representative excerpts identification algorithm to yield better beginning and ending boundaries of the identified excerpts and improve the recall rate.

The ultimate intention of a music creator is that the music is to be listened by an audience, which usually consists of a group of individuals. Each individual will make different and subjective interpretations of music presented to him or her. Our current evaluation method assumes that "chorus" sections are the significant sections to represent a piece of music. In future, if time and resources allow us, we would like to assess our system based on the judgment from human listeners in order to obtain a subjective evaluation. This can be done through generating short excerpts from music pieces and ask an acceptable amount of listeners to select the correct song titles.

Finally, other future direction includes incorporating our algorithm in practical applications. It is clear that identifying representative musical excerpts of audio files has relevance to music summarization. If a music piece contains the following structure: AABACBAA. The generation of the audio summary can be accomplished by [Peeters02]:

- Providing an audio example of the most repetitive section in term of number of appearances of the section (A)
- Providing a unique audio example of only main difference sections (A, B, C)
- Producing an audio example of containing short example of each different section sequence (A, B, A, C, B, A)
- Providing audio example of section transition (A→B, B→A, A→C, C→B)
- Etc.

Thus, our future plan is to make use of our algorithm for the detection of representative excerpts to generate summaries of music for facilitating better audio browsing and retrieval. Instead of only playing back the audio summaries, the structure of music will also be visualized on the screen display coupled with click-and-play mode to allow users to have easy access to any particular section by just clicking on the displayed section-block as shown in Figure 5.1.

5.3. Final Thoughts

The impact of the above mentioned factors on the identification of representative musical excerpts and semantic audio segmentation will not only improve the efficiency and effectiveness of the current process but also will yield a better semantic representation of musical audio signals. This will be useful for practical applications in audio indexing, audio browsing and audio database managing such as those that are being addressed in the SIMAC project. The presented overview is expected to make helpful contributions to the development of this software.

APPENDICES

A. GLOSSARY

This section is a glossary of the basic terminology used in this thesis provided for quick reference.

Beat: a rhythmic sub division of music usually felt as the regular timing within a piece of music.

Bridge: an interlude that connects two parts of a song and builds a harmonic connection between those parts.

Chorus (or refrain): the part of a song where a soloist is joined by a group of singers.

Clustering: the process of organizing objects into groups whose members are similar in some way.

Intro: introduction of a song.

Key-frames: the excerpts which best represent the content of a music sequence in an abstract manner, and are extracted from the original audio signal.

Onset: the change points in musical signals which are equivalent to the human perception of a new note starting.

Outro: the ending of a song

Summarization: the process of generating a short abstract from the original audio signal to represent the whole file.

Thumbnailing: the process of extracting a short excerpt from the original audio signal to represent the whole file.

Verse: the song sections that roughly corresponds to a poetic stanza. It is often sharply contrasted with the chorus (or refrain) melodically, rhythmically, and harmonically.

BIBLIOGRAPHY

[Adam03] Adams, W. H., Lyengar, G., Lin, C-Y, Naphade, M. R., Neti, C., Nock, H. J., and Smith, J. R. Semantic Indexing and Multimedia Content Using Visual, Audio, and Text Cues. *EURASIP Journal on Applied Signal Processing*, 2003:2, pp. 170-185, 2003.

[Arons93] Arons, B. SpeechSkimmer: Interactively Skimming Recorded Speech. *ACM Symposium on User Interface Software and Technology (UIST'93), ACM Press,* pp. 187-196, 1993.

[Aucouturier01] Aucouturier, J. -J. and Sandler, M. Segmentation of Musical Signals Using Hidden Markov Models. *AES 110th Convention,* Amsterdam, the Netherlands, 2001.

[Aucouturier02] Aucouturier, J. -J. and Sandler, M. Finding Repeating Patterns in Acoustic Musical Signals: Applications for Audio Thumbnailing. AES *22nd International Conference on Virtual, Synthetic and Entertainment Audio, Espoo*, Finland, 2002.

[Bartsch05] Bartsch, M. and Wakefield, G. Audio Thumnailing of Popular Music Using Croma-Based Representations. *IEEE Transactions on Multimedia*, vol. 7 No. 1, 2005.

[Bartsch01] Bartsch, M. and Wakefield, G. To Catch A Chorus: Using Chroma-Based Representations for Audio Thumbnailing. *IEEE Workshop on Applications of Signal Processing on Audio andAcoustics, WASPAA,* New Paltz, New York, USA, 2001.

[Birmingham01] Birmingham, W. P., et al. MUSART: Music Retrieval via Aural Queries. *Proceedings Second International Symposium on Music Information Retrieval,* pp. 73-81, 2001.

[Burgeth04] Burgeth B., Welk M., Feddern C., and Weickert J. Morphological Operations on Matrix-Valued Images. *The 8th European Conference on Computer Vision*, Prague, Czech, pp. 155-167, May 2004.

[Chai03a] Chai, W. and Vercoe, B. Music Thumbnailing via Structural Analysis. *Proceedings of ACM Multimedia Conference*, November 2003.

[Chai03b] Chai, W. and Vercoe, B. Structural Analysis of Musical Signals for Indexing and Thumbnailing. *Proceedings of ACM/IEEE Joint Conference on Digital Libraries*, May 2003.

[Chai03c] Chai, W. *Structural Analysis of Musical Signals for Indexing, Segmentation and Thumbnailing*. Paper for the Major Area of the PhD General Exam, March 2003.

[Chai03d] Chai, W. Structural Analysis Of Musical Signals via Pattern Matching. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2003.

[Cheng03] Cheng, Y. *Content-based Musical Retrieval on Acoustical Data*. Ph.D. thesis, Stanford University, August 2003.

[Cooper02] Cooper, M. and Foote, J. Automatic Music Summarization via Similarity Analysis. *International Symposium on Music Information Retrieval*, Paris, France, 2002.

[Cooper03] Cooper, M. and Foote, J. Summarizing Popular Music via Structural Similarity Analysis, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 19, 2003.

[Croonen94] Croonen, W. L. Effects of Length, Tonal Structure, and Contour in the Recognition of Tone Series. *Perception & Psychophysics*, 55, pp. 623–632, 1994.

[Dannenberg02a] Dannenberg, R. B. and Hu, N. Discovering Musical Structure in Audio Recordings. *Proc. 2nd Int. Conference in Music & Artificial Intelligence (ICMAI),* 2002.

[Dannenberg02b] Dannenberg, R. B. Listening to `Naima': An Automated Structural Analysis of Music from Recorded Audio. *Proceedings of the 2002 International Computer Music Conference,* San Francisco: International Computer Music Association, pp. 28-34, 2002.

[Dowling78] Dowling, W. J. Scale and Contour: Two Components of a Theory of Memory for Melodies. *Psychological Review,* 85, pp. 342-354, 1978.

[Edworthy85] Edworthy, J. Melodic Contour and Musical Structure. In P. Howell, I. Cross, & R.J. West (Ed.), *Musical Structure and cognition*, Academic Press Inc., London, pp. 169-188, 1985.

[Ellis94] Ellis, G. M. Electronic Filter Analysis and Synthesis, Artech House, 1994.

[Filonov05] Filonov, A. S., Gavrilko, D. Y., and Yaminsky, I. V. *Scanning Probe Microscopy Image Processing Software User's Manual "FemtoScan". version 4.8*. Moscow: Advanced Technologies Center. (2005) http://www.spm.genebee.msu.su/manual/en/node108.html

[Foote99] Foote, J. Visualizing Music and Audio Using Self-Similarity. *ACM Multimedia*, pp. 77–84, Orlando, Florida, USA, 1999.

[Foote00] Foote, J. Automatic Audio Segmentation Using a Measure of Audio Novelty. *IEEE Int. Conf. Multimedia and Expo (ICME)*, vol. I, page 452-255, New York City, NY, USA, 2000.

[Foote03] Foote, J. and Cooper, M. Media Segmentation Using Self-Similarity Decomposition. *Proceedings of SPIE*, vol. 5021:1, pp. 67-75, 2003.

[Fujishima99] Fujishima, T. Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music. *Proceedings of International Computer Music Conference*, Beijing, China, 1999, *ICMA*, San Fransixco, pp. 464-467, 1999.

[Gang02] Gang, Q., Sural, S. and Pramanik, S. A Comparative Analaysis of Two Distance Measures in Color Image Databases. *IEEE International Conference of Image Processing,* vol. 1, pp. 22-25, 2002.

[Gómez03] Gómez, E., Klapuri, A., and Meudic, B. Melody Descriptions and Extraction in the Context of Music Content Processing. *Journal of New Music Research*, vol. 32.1, 2003.

[Gómez04] Gómez, E. Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing, Special Cluster on Computation in Music*. Guest Editor: Elaine Chew. Associate Editors: Roger B. Dannenberg, Joel Sokol and Mark Steedman. 2004.

[Goto99] Goto, M. A Real-Time Music Scene Description System: Detecting Melody and Bass Lines in Audio Signals. *Proceedings of the IJCAI Workshop on Computational Auditory Scene Analysis*, pp. 31-40, 1999.

[Goto00] Goto, M. Robust Predominant-F0 Estimation Method for Real-time Detection of Melody and Bass Lines in CD Recordings. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. II-757-760, 2000.

[Goto03a] Goto, M. A Chorus-Section Detecting Method for Musical Audio Signals. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.V-437-440, April 2003.

[Goto03b] Goto, M. A SmartMusicKIOSK: Music Listening Station with Chorus-Search Function. *Proceedings of UIST*, pp.31-40, 2003.

[Huron99] Huron, D. *Humdrum User-s Guide*, http://dactyl.som.ohio-state.edu/Humdrum/guide.toc.html, last update 1999.

[Logan00] Logan, B. and Chu, S. Music summarization using key phrases. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Istanbul, Turkey, 2000.

[Lu04] Lu, L., Wang, M., and Zhang, H-J. Repeating Pattern Discovery and Structure Analysis from Acoustic Music Data. *MIR' 04*, October 15-16, New York, USA, pp. 275-282, 2004.

[MacQueen67] MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* vol. 1, pp. 281-296, 1967.

[Maddage04] Maddage, N. C., Xu, C., Kankanhalli, M. S., and Shao, X. Content-Based Music Structure Analysis with the Applications to Music Semantic Understanding. *ACM Multimedia Conference (ACM MM04)*, 2004.

[Mulhem03] Mulhem, P., Gensel, J. and Martin, H. Adaptive Video Summarization. In *The Handbook of Video Databases Design and Applications*, B. Furht and O. Marques (Eds), CRC Press, Boca Raton, pp279-298, 2003.

[Nam97] Nam, J., Cetin, A. E., and Tewfik, A. H. Speaker Identification and Video Analysis for Hierarchical Video Shot Classification. *Proc. IEEE Int. Conf. Image Processing,* vol. 2, Santa Barbara, CA, Oct, pp. 550-555, 1997.

[Ong04] Ong, B., and Herrera, P. "Computing Structural Descriptions of Music through the Identification of Representative Excerpts from Audio Files", *Proceedings of 25th International AES Conference London,* UK, June 2004.

[Otsu79] Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. SMC*, vol. SMC-9, no. 1, pp. 62–66, 1979.

[Peeters02] Peeters, G., Laburthe, A., and Rodet, X. Toward Automatic Music Audio Summary Generation From Signal Analysis. *International Conference on Music Information Retrieval, ISMIR*, Paris, France, 2002.

[Rabiner86] Rabiner, L. R. and Juang, B. H. *An Introduction to Hidden Markov Models.* IEEE ASSP Magazine, pp. 4-15, 1986.

[Rabiner89] Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE,* vol. 77 (2) , pp. 257-286, February 1989.

[Rabiner93] Rabiner L. and Juang, B. H. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[Roediger05] Roediger, H. L. Memory (psychology). *Microsoft® Encarta® Online Encyclopedia 2005* http://encarta.msn.com, 2005.

[Selfridge97] Selfridge-Field, E.(editor), *Beyond MIDI: The Handbook of*

Musical Codes. Cambridge, Massachusetts: MIT Press, 1997.

[Selfridge98] Selfridge-Field, E. Conceptual and Representational Issues in Melodic Comparison. *Melodic Comparison: Concepts, Procedures, and Applications, Computing in Musicology* 11, pp. 3-64, 1998.

[Siegler97] Siegler, M. A., Jain, U., Raj, B. and Stern, R. M. Automatic Segmentation, Classification and Clustering of Broadcast News Audio. Proceeding of the DARPA speech recognition workshop pp. 97-99, 1997.

[Smith04] Smith, M. A. and Kanade, T. *Multimodal Video Characterization and Summarization.* (The Kluwer International Series in Video Computing), Springer. 2004.

[Solomon97] Solomon, L. *Music Theory Glossary.* Web publication, last updated 2002, http://solo1.home.mindspring.com/glossary.htm., 1997.

[Tzanetakis99] Tzanetakis, G. and Cook, P. Multifeature Audio Segmentation for Browsing and Annotation. *In WASPAA*, New Paltz, New York, USA, 1999.

[Tzanetakis02] Tzanetakis, G. Pitch Histograms in Audio and Symbolic Music Information Retrieval. *International Symposium on Music Information Retrieval*, 2002.

[Steelant02] Van Steelant, D., DeBaets, B., DeMeyer, H., Leman, M., Martens, S.-P., Clarisse, L., and Lesaffre, M. Discovering Structure and Repetition in Musical Audio. *In Eurofuse*, Varanna, Italy, 2002.

[Wang00] Wang, Y., Liu, Z., and Huang J-C. Multimedia Content Analysis Using Both Audio and Visual Clues. *IEEE Signal Processing Magazine*, pp.12-36, 2000.

[Warren03] Warren, J. D., Uppenkamp, S., Patterson, R. D., and Griffiths, T. D. Separating Pitch Chroma and Pitch Height in the Human Brain. *Proceedings of the National Academy of Sciences of the United States of America,* vol. 17, pp. 10038–10042, 2003.

[Weyde03] Weyde, T. Case Study: Representation of Musical Structure for Music Software. *Proceedings of the Music Notation Workshop: XML based Music Notation solutions,* 2003.

[Widmer03] Widmer, G., Dixon, S., Goebl, W., Pampalk, E., and Tobudic, A. *In Search of the Horowitz Factor.* AI Magazine vol. 24, no. 3, pp. 111-130, 2003.

[Xu02] Xu, C., Zhu, Y., and Tian, Q. Automatic Music Summarization Based on Temporal, Spectral and Cepstral Features. *Proceedings of IEEE International Conference on Multimedia and Expo* pp.

117-120, 2002.

[Xu04] Xu, C., Shao, X., Maddage, N. C., Kankanhalli, M. S., and Tian, Q. Automatically Summarize Musical Audio Using Adaptive Clustering. *IEEE International Conf of Multimedia Explore(ICME04)*, Taibei, Taiwan, China, 2004.

[Xu05] Xu, C., Maddage, N. C., and Shao, X. Automatic Music Classification and Summarization. *IEEE Transactions of Speech and Audio Processing*, vol. 13. No. 3, May 2005.

[Young02] Young, N. *Mathematical Morphological*. http://www.bath.ac.uk/eleceng/pages/sipg/research/morphology.htm. Last updated July 2002.

[Zhang03] Zhang, T. Semi-Automatic Approach for Music Classification. *SPIE's Conference on Internet Multimedia Management Systems IV (part of ITCom'03),* vol. 5242, Orlando, Sep. 2003.